Méthodes d'extraction de connaissances à partir de données modélisables par des graphes. Application à des problèmes de synthèse organique.

Frédéric Pennerath

Nancy-Université
Université Henri Poincaré

Département de formation doctorale en informatique **École doctorale IAEM Lorraine**
UFR STMIA

Méthodes d'extraction de connaissances à partir de données modélisables par des graphes. Application à des problèmes de synthèse organique.

THÈSE

présentée et soutenue publiquement le 2 juillet 2009

pour l'obtention du

Doctorat de l'université Henri Poincaré – Nancy 1

(spécialité informatique)

par

Frédéric Pennerath

Composition du jury

Président :	Marie-Christine Haton	Professeur émérite à l'université Henri Poincaré de Nancy
Rapporteurs :	Bruno Crémilleux	Professeur à l'université de Caen Basse-Normandie
	Pascal Poncelet	Professeur à l'université Montpellier II
Examinateurs :	Amedeo Napoli	Directeur de recherches CNRS au Loria (codirecteur de thèse)
	Gilles Niel	Chargé de recherches CNRS à l'Institut Charles Gerhardt de Montpellier
	Lhouari Nourine	Professeur à l'université Blaise Pascal de Clermont-Ferrand
	Géraldine Polaillon	Enseignant-chercheur à Supélec (codirectrice de thèse)

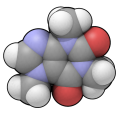
Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Loria

Mis en page avec la classe thloria.

Remerciements

Les personnes qui ont, à des degrés divers, contribué à l'aboutissement de ce travail sont nombreuses. Je les en remercie toutes. En premier lieu, je remercie Amedeo Napoli pour avoir accepté de diriger ma thèse, pour avoir proposé un sujet très intéressant, pour m'avoir formé aux pratiques du métier de chercheur et pour m'avoir fait confiance par delà des incompréhensions passagères. Je remercie particulièrement Bruno Crémilleux et Pascal Poncelet d'avoir accepté d'être les rapporteurs de ce mémoire. Je les en remercie d'autant plus qu'il leur a fallu lire en relativement peu de temps un document assez long et dont l'intrigue, je dois me rendre à l'évidence, n'en fera pas le meilleur récit à suspens de cet été 2009. Je remercie également Géraldine Polaillon pour avoir participé à la direction de ma thèse, ainsi que Lhouari Nourine, Marie-Christine Haton et Gilles Niel pour avoir accepté de faire partie du jury et s'être ainsi penchés sur mes travaux. Merci aux chercheurs montpelliérains chimistes ou informaticiens Philippe Jauffret, Claude Laurenço, Gilles Niel et Philippe Vismara avec qui j'ai apprécié de collaborer dans le cadre d'un projet PEPS financé par le CNRS. Vos remarques toujours justes et constructives m'ont convaincu de l'importance qu'un scientifique digne de ce nom doit accorder à la précision de ses propos et à la rigueur de son travail. Merci notamment aux chimistes Claude Laurenço et Gilles Niel d'avoir partagé avec enthousiasme un peu de leur science. Grâce à vous, les mauvais souvenirs des cours de chimie sont oubliés. Je remercie tout particulièrement Gilles Niel pour son hospitalité et pour avoir toujours répondu avec la plus grande amabilité à mes demandes répétées de service. Je tiens également à remercier la direction de Supélec, qui m'a témoigné sa confiance en me donnant la chance et les moyens de préparer cette thèse. Merci en particulier à Patrick Turelle et Joel Jacquet de m'avoir ménagé du temps afin que ma thèse puisse se dérouler dans de bonnes conditions et d'avoir renouvelé leur confiance à mon égard à des moments difficiles. Merci aux nombreux collègues qui m'ont témoigné leur soutien, notamment Olivier Pietquin, Hervé Frezza-Buet et Stéphane Vialle dont les plaisanteries parfois narquoises à mon égard n'avaient pour seul objectif, je n'en doute pas, de piquer mon amour propre et ainsi de me remotiver. Et puis surtout je tiens à dire infiniment merci à Aurélie pour avoir supporté mes monologues de chercheur contrarié avec la plus grande patience, pour son indéfectible soutien et ses encouragements sans lesquels je n'aurais jamais osé remettre en question mes choix professionnels. J'espère que tu ne le regrettes pas ! Merci aussi à mes parents qui ont encouragé et soutenu leurs enfants à faire des études et plus particulièrement à ma mère pour toute l'attention qu'elle a portée depuis toujours à ses enfants. Je mesure combien cette thèse est aussi le fruit de ton travail. Merci aussi à toute la famille d'Alsace ou d'Auvergne et à tous les amis de France et d'outre-Rhin pour les moments de réconfort et de convivialité fort appréciés qu'ils ont apportés. Merci par avance à toutes celles et ceux qui m'excuseront de ne pas les avoir cités ici. Merci enfin à cette petite molécule à qui tant de doctorants doivent leur salut et dont je connais maintenant la formule !



À Aurélie.

Table des matières

Chapitre 1

Introduction

Aux origines

Si le sujet de ce mémoire devait se résumer en une seule phrase, ce serait une question : *Peut-on imaginer des méthodes de fouille de données et plus précisément de fouille de graphes qui puissent à partir des bases de données de réactions chimiques extraire des connaissances utiles à la synthèse des molécules ?* Pour mieux comprendre cette question, il est intéressant d'en connaître son histoire : cette question qui me fut posée initialement par Amedeo Napoli, tire ses origines du GDR TICCO[1] qui regroupait autour d'un projet baptisé RESYN (Vismara *et al.*, 1992, 1998), aussi bien des chercheurs chimistes, notamment les membres de l'ancienne équipe SIC[2], que des chercheurs informaticiens, notamment Amedeo Napoli et Philippe Vismara. L'objet de ce projet était la conception d'un système d'assistance à la synthèse de molécules, c'est-à-dire d'un système informatique conçu pour faciliter, à défaut d'automatiser, le processus de conception d'un plan de synthèse d'une molécule, prenant la forme d'une séquence de réactions chimiques. L'originalité du projet résidait dans l'architecture de ce système appelé RESYN, fondée non pas comme d'autres systèmes sur des règles ad hoc, mais sur un modèle de représentation des connaissances adapté au problème de la synthèse organique (i.e. la branche de la chimie qui s'intéresse à la synthèse des molécules) et sur le raisonnement par classification. Cette architecture devait en outre faciliter l'extension et la mise à jour de sa base de connaissances, voire la compléter par un processus d'inférence automatique.

Au delà de sa finalité, ce projet illustre bien la forte interaction qui existe depuis longtemps entre informatique et chimie et qui permet aujourd'hui aux chimistes de disposer de nombreux outils informatiques adaptés à leurs besoins, au sein d'une branche de la chimie qui a adopté le nom beaucoup plus récent de chémoinformatique. Cette relation ancienne entre les deux disciplines s'explique par le fait que la chimie a besoin, notamment de par sa nature combinatoire, de l'informatique pour résoudre certains de ses problèmes, et que la résolution de ces derniers est une source d'innovation très fructueuse pour la recherche en informatique et plus particulièrement pour l'intelligence artificielle. À ce propos, les systèmes d'aide à la synthèse tels que RESYN présentent les inconvénients caractéristiques des systèmes experts qu'on a l'habitude de reconnaître en intelligence artificielle. Le projet RESYN a ainsi mis en évidence la difficulté qui existe à formaliser avec précision et le plus complètement possible, la connaissance relative à un domaine aussi complexe que la synthèse organique. De plus, il est difficile d'assurer la mise à jour de la base de connaissances dans la durée. En comparaison,

[1] GDR 1093 du CNRS « Traitement Informatique de la Connaissance en Chimie Organique », 1993 – 2001.
[2] Équipe *Systèmes d'Information Chimique* de l'UMR 5076 CNRS-ENSCM, Montpellier

les systèmes d'information chimique capables de gérer de très grandes bases de données de molécules et de réactions chimiques ont connu un développement beaucoup plus rapide et sont aujourd'hui utilisés quotidiennement dans l'industrie. Ce succès s'explique par le fait qu'il est beaucoup plus simple de collectionner de grandes quantités de données indépendantes que de réorganiser en permanence une base de connaissances complexe, même quand celle-ci est de petite taille. Aujourd'hui les *bases de données de réactions chimiques* – abrégées par BdR – contiennent des millions de réactions qui reflètent l'évolution de la connaissance en synthèse organique au cours de plusieurs décennies. Cependant cette connaissance est implicite et pour ainsi dire invisible, les BdR n'étant que la juxtaposition de données décrivant des réactions particulières et indépendantes. Certains membres du GDR TICCO, en particulier Claude Laurenço et Amedeo Napoli, ont donc eu l'idée de chercher à extraire une partie de cette connaissance à l'aide de méthodes de fouille de données. Cette idée qui s'est concrétisée à travers la thèse de Sandra Berasaluce (Berasaluce, 2002), ouvre une perspective intéressante sur le long terme : celle de passer des systèmes d'aide à la synthèse organique fondés sur la représentation des connaissances à ceux fondés sur l'extraction de cette connaissance à partir des BdR. Dans sa thèse, Sandra Berasaluce a abordé ce problème en représentant par des symboles, certains groupes d'atomes particuliers appelés fonctions, présents dans les réactions des BdR ; puis elle a fouillé les motifs qui apparaissent fréquemment dans ces ensembles de symboles. Si cette approche a donné lieu à des résultats intéressants, sa portée est toutefois limitée par la description symbolique des réactions qu'elle adopte : cette description est en effet une « vue » assez étroite de la réalité complexe des molécules et des réactions, car composée d'un nombre limité de symboles définis a priori. Pourtant les chimistes ont développé des représentations iconiques (i.e. à base de diagrammes) qui sont beaucoup plus adaptées que les représentations symboliques pour saisir la structure et comprendre les propriétés des molécules et des réactions chimiques. Dans la mesure où certains de ces diagrammes sont disponibles dans les BdR sous une forme simplifiée de graphes, il est apparu intéressant de compléter les travaux de fouille de données symbolique réalisés par S. Berasaluce, en fouillant cette fois-ci les graphes contenus dans les BdR. L'émergence alors récente de méthodes de fouille de graphes permettant la recherche des sous-graphes fréquents a conforté cette approche.

Des problèmes de fouille de graphes

C'est ainsi qu'est né le sujet de ce mémoire. À partir de là, la première étape du projet a consisté à mieux appréhender quels sont les problèmes posés aux chimistes pour synthétiser une molécule et pour lesquels la fouille de graphes pourrait apporter des éléments de réponse pertinents. Cette étude et les notions de chimie qui l'accompagnent sont décrites en détail au chapitre 3. En résumant, le problème de la synthèse d'une molécule consiste à trouver un *plan de synthèse*, c'est-à-dire un enchaînement de réactions chimiques partant de composés chimiques disponibles et aboutissant à la molécule convoitée. La conception de ce plan de synthèse se fait grâce à l'aide de la notion centrale de *méthode de synthèse*. De manière très simplifiée, une méthode de synthèse est une transformation générique (au sens de réutilisable) de la structure de molécules qui permet d'atteindre un objectif stratégique, c'est-à-dire la construction d'une sous-structure moléculaire particulière. En pratique une méthode de synthèse se représente par un, parfois plusieurs schémas de réactions génériques qui peuvent être réutilisés – ou instanciés dirait un informaticien – dans différentes expériences. Il en ressort que la synthèse organique comprend deux problématiques complémentaires que sont la méthodologie de synthèse et la synthèse ciblée : d'une part, la *méthodologie de synthèse* a

pour objet d'identifier, d'optimiser et de répertorier les *méthodes de synthèse*. D'autre part, la *synthèse ciblée* a elle, pour objet de synthétiser une molécule cible donnée en concevant un plan de synthèse le plus efficace possible (i.e. le moins coûteux, le plus sûr ...), s'appuyant sur les méthodes de synthèse établies préalablement par la méthodologie de synthèse. Du point de vue applicatif, ce mémoire contribue à l'une et l'autre des problématiques : concernant la méthodologie de synthèse, le problème abordé est celui de l'identification des schémas de réactions génériques caractéristiques de méthodes de synthèse, abrégés par schémas CMS. Si l'on considère que ces schémas de réactions sont des motifs « enfouis » dans les graphes des BdR, il apparaît clairement que cette question peut se traiter comme un problème de fouille de graphes. Concernant la synthèse ciblée, le problème abordé est celui de la détermination, dans une molécule cible – i.e. que l'on cherche à synthétiser – des liaisons entre atomes qui sont les plus faciles à former à l'aide d'une réaction chimique. Là encore l'intérêt d'aborder ce problème sous l'angle d'un problème de fouille de graphes est évident quand on sait que d'une part, les BdR renferment de nombreux exemples de liaisons formées et que d'autre part, la plupart des liaisons formées sont caractérisées par des environnements structuraux spécifiques représentables par des graphes.

Les problèmes soulevés par la synthèse organique sont donc le point de départ de ce mémoire. Cependant les questions traitées au cours des différents chapitres sont essentiellement de nature informatique, à l'exception peut-être du chapitre 4 qui nécessite la prise en compte de nombreux détails techniques spécifiques à la modélisation des réactions chimiques. La problématique générale au centre de ces différentes questions est l'extraction dans des données de motifs qui soient pertinents relativement à l'application traitée. Cette problématique de la sélection des motifs est récurrente en fouille de données, mais elle est ici conditionnée par deux points essentiels : premièrement et compte tenu des applications visées, les méthodes de sélection de motifs doivent pouvoir traiter le cas particulier où les données et donc les motifs sont des graphes. Ce pré-requis impose des contraintes d'ordre théorique, mais surtout d'ordre pratique, liées à la complexité de calcul que nécessite la fouille d'un ensemble de graphes. Deuxièmement, l'intérêt ou la pertinence d'un motif ne peut pas se définir de façon absolue mais relativement à une application donnée. Il est donc important de pouvoir intégrer dans le processus de sélection des motifs la connaissance a priori du problème, non pas nécessairement sous la forme de modèles formels de représentation des connaissances, mais sous une forme plus générale, comme par exemple des fonctions de score, des heuristiques, des contraintes particulières ou une modélisation appropriée des données. Cette nécessité d'intégrer des contraintes spécifiques à l'application traitée s'est manifestée tout au long du mémoire, et a conduit pour chacun des chapitres 5, 6 et 7 à formaliser un problème original de fouille de graphes, ainsi qu'au développement d'une méthode spécifique pour y répondre. Par ailleurs, ces problèmes ont pu contribuer à des problématiques de recherche actuelles dans le domaine de la fouille de données. Ainsi le modèle des motifs les plus informatifs exposé au chapitre 5 peut être rattaché aux différentes représentations condensées de motifs et à la réduction de la redondance d'information entre motifs alors que les chapitres 6 et 7 proposent des méthodes de fouille de graphes originales, qualifiées de « transductives ». En résumé, ce mémoire porte sur l'extraction sélective de motifs structuraux complexes que sont les graphes, et de l'adaptabilité de cette extraction aux besoins spécifiques d'applications ici choisies dans le domaine de la synthèse organique.

Plan et contributions du mémoire

Le plan du mémoire présente deux facettes différentes selon que l'on adopte le point de vue de l'informaticien ou du chimiste. En effet, si les deux premiers chapitres forment une première partie consacrée à l'état de l'art, les contributions présentées dans les quatre chapitres suivants se regroupent différemment selon le point de vue adopté, comme l'illustre la figure 1.1. D'abord du point de vue de la fouille de données, les chapitres 4 et 5 sont liés

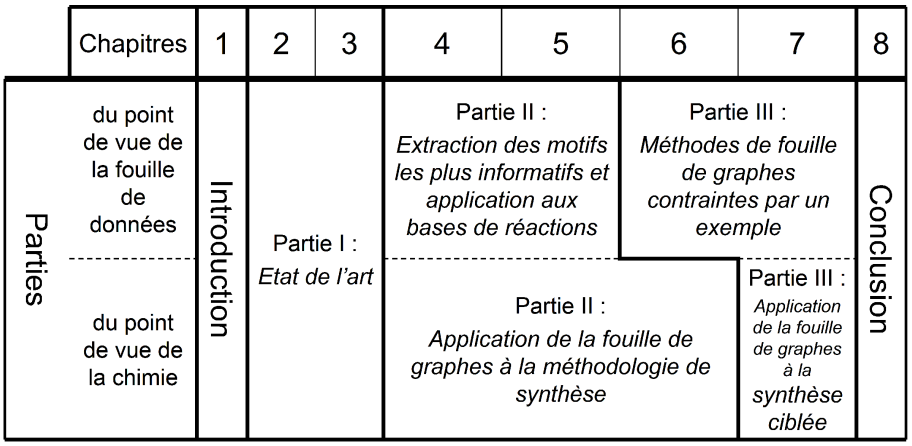


Fig. 1.1 – Plan du mémoire

puisque le premier permet d'extraire les schémas de réactions fréquents à partir desquels le second sélectionne à l'aide d'un modèle général, un ensemble réduit de schémas pertinents. De même les méthodes présentées aux chapitres 6 et 7 adoptent des principes similaires pour traiter tantôt un problème de classification non supervisée de graphes, tantôt un problème de classification supervisée de sommets ou d'arêtes d'un graphe. À l'inverse, si on adopte le point de vue du chimiste, les trois chapitres 4, 5 et 6 forment un développement cohérent rattaché à la méthodologie de synthèse et consacré à l'extraction des schémas CMS, alors que le chapitre 7 est le seul à aborder le problème de la synthèse ciblée.

Plus précisément, les deux premiers chapitres sont dévolus à l'état de l'art et plus exactement aux deux piliers de connaissances sur lesquels s'appuie le reste du mémoire : la fouille de graphes d'un côté et la chémoinformatique de l'autre. Le chapitre 2 commence par dresser un état de l'art des méthodes existantes de fouille de graphes et les situent au sein du contexte plus général de la fouille de données. Y sont notamment présentés certains formalismes permettant de fouiller des données relationnelles, dont certains furent étudiés un temps comme des alternatives potentielles aux méthodes de fouille de graphes. Il s'agit en particulier de l'analyse de concepts relationnels et des structures de motifs – vus tous deux comme des généralisations de l'analyse de concepts formels – ou encore de la programmation logique inductive.

Le chapitre 3 vise quant à lui à remplir deux objectifs distincts mais qui ont en commun la synthèse organique comme sujet. Le premier objectif est de fournir une introduction très succincte des notions fondamentales de chimie organique qui sont essentielles à la compréhension du volet applicatif de cette thèse. Il s'agit essentiellement des molécules, des réactions chimiques et des méthodes de synthèse. Le second objectif est de produire un état de l'art sur les branches de la chémoinformatique concernant et concernées par les travaux de ce mémoire. En particulier sont développés les systèmes d'aide à la résolution de problèmes de synthèse ainsi que les travaux liés à la fouille de données chimiques.

Viennent ensuite les quatre chapitres présentant les travaux réalisés dans le cadre de cette thèse. Le chapitre 4 commence par traiter du problème de la recherche des schémas de réactions fréquents dans les bases de données de réactions. Une transposition adéquate du problème permet de le résoudre à l'aide des méthodes existantes de recherche de sous-graphes fréquents présentées au chapitre 2. Sont développés en particulier les processus de prétraitement des BdR et de post-traitement des résultats qui ont été nécessaires pour filtrer, corriger et surtout transformer les données dans un modèle adapté à la fouille de graphes. Par ailleurs le prétraitement a incidemment permis d'estimer la qualité des données dans les BdR. Le chapitre conclut sur le faible intérêt que présente l'analyse des sous-graphes fréquents (et donc des schémas de réactions fréquents) trop nombreux et en moyenne peu pertinents, ce qui justifie le développement des chapitres suivants.

Le chapitre 5 introduit une nouvelle famille de motifs, qualifiés de *motifs les plus informatifs*. L'objectif de ce modèle est de produire un ensemble réduit de motifs fréquents qui soient à la fois représentatifs des données et peu redondants structurellement. Le chapitre s'intéresse ensuite aux propriétés formelles du modèle avant de proposer deux algorithmes d'extraction des motifs les plus informatifs fréquents, soit en fouillant directement les données initiales, soit en filtrant les motifs fréquents. L'efficacité des deux algorithmes est comparée dans le cas des motifs d'attributs. Le modèle est ensuite appliqué aux bases de données de réactions, en filtrant les schémas de réactions fréquents étudiés au chapitre précédent. L'analyse des schémas les plus informatifs fréquents confirme qu'ils sont bien moins nombreux et moins similaires structurellement que certaines familles de motifs condensés, comme les motifs fermés fréquents, tout en étant représentatifs des données. Par ailleurs, les schémas les plus informatifs apparaissent pertinents du point de vue de la synthèse organique, dans la mesure où la plupart de ces schémas représentent des familles remarquables de réactions ou de molécules. L'extraction des motifs les plus informatifs fréquents semble toutefois inadaptée à l'extraction des schémas CMS car les schémas visés ont une fréquence si faible qu'ils sont hors d'atteinte des algorithmes de recherche de sous-graphes fréquents.

Pour fouiller les motifs à des fréquences aussi faibles que celles des schémas CMS, le chapitre 6 propose de contraindre davantage l'espace de recherche en cherchant le ou les motifs qui maximisent une fonction de score tout en étant inclus structurellement dans un exemple spécifique. Cette approche est qualifiée ici de « transductive » par analogie avec les méthodes transductives d'apprentissage numérique. Ainsi, au lieu de fouiller l'ensemble de tous les motifs fréquents présents dans un ensemble de graphes, l'algorithme `CrackReac` contraint les motifs fouillés à être inclus dans un intervalle de graphes spécifique. Étant donnée une réaction chimique, cette méthode permet alors d'extraire le *schéma caractéristique* de cette réaction, qui s'avère être, sous certaines conditions, une bonne approximation du schéma convoité (i.e. du schéma CMS sous-jacent à la réaction tel qu'il serait défini par un expert).

Le chapitre 7 étend l'approche « transductive » introduite au chapitre 6, au problème de la classification des sommets ou des arêtes d'un graphe fondée sur leur environnement. Ce problème est développé pour servir une application rattachée non plus à la problématique de la méthodologie de synthèse mais à celle de la synthèse ciblée : cette application consiste à estimer à partir d'exemples, la « formabilité » des liaisons d'une molécule cible, c'est-à-dire la facilité avec laquelle chaque liaison peut être formée par une réaction chimique. Ce problème est résolu à l'aide d'une méthode de recherche heuristique de motifs baptisée `GemsBond` calculant à partir d'exemples de graphes, un niveau de confiance associée à l'hypothèse qu'un sommet ou une arête d'un graphe appartienne à une classe cible. La méthode produit en outre le sous-graphe présent dans l'environnement de chaque sommet ou arête, qui justifie de cette confiance. En ce sens, la méthode contribue à la problématique de l'extraction de

connaissances à partir de données. L'analyse statistique des résultats produits par `GemsBond` pour estimer la formabilité des liaisons est très encourageante et ces conclusions ont pu par ailleurs être confirmées lors de l'examen des résultats par un expert de la synthèse organique.

Pour finir, le chapitre 8 fait la synthèse des idées présentées dans ce mémoire en retraçant l'histoire de leur développement, avant de mentionner quelques prolongements possibles de ces travaux.

Chapitre 2

La fouille de données modélisables par des graphes

Sommaire

Les travaux présentés dans ce mémoire se rapportent à la *fouille de graphes*[3] (Cook et Holder, 1994; Yoshida *et al.*, 1994; Inokuchi *et al.*, 2000; Yan et Han, 2002; Karwath et Raedt, 2004; Cook et Holder, 2006), qui forme une branche non négligeable de la recherche en fouille de données[4]. La fouille de graphes regroupe l'ensemble des méthodes qui recherchent des motifs structuraux (i.e. de sous-graphes) dans des données représentées sous forme de graphes. Le lien entre fouille de graphes et fouille des BdR s'explique par le fait que de nombreux phénomènes liés aux réactions chimiques peuvent s'expliquer par la présence dans les molécules de configurations particulières d'atomes liés par des liaisons. Ces configurations forment donc des motifs représentables par des graphes que l'on peut rechercher dans les BdR, dans l'objectif d'apprendre des connaissances relatives à ces phénomènes. Cependant les

[3]Des dénominations anglaises *graph mining* ou *graph-based data mining*.

[4]Les problèmes de fouille de graphes ont représenté en 2008 entre 5 à 10 % des quelques 300 articles acceptés aux conférences SIGKDD, PKDD/ECML et ICDM 2008.

méthodes de fouille de graphes ne sont pas les seules méthodes d'extraction de connaissances à pouvoir tenir compte de motifs structuraux complexes semblables aux graphes. Ainsi, si on interprète les liaisons chimiques comme des relations binaires entre atomes, on peut vouloir utiliser les méthodes de *fouille de relations*[5] dont l'objectif est de décrire les objects spécifiés dans les données par les relations qui lient ces objets entre eux. De manière analogue, si on interprète les liaisons comme des prédicats binaires, la programmation logique inductive, dont l'objet est d'induire une théorie logique à partir de faits, est une autre possibilité intéressante. L'objet de ce chapitre est donc double : d'une part, il s'agit de décrire ce que recouvre précisément la fouille de graphes au sein du contexte plus général de la fouille de données relationnelles. D'autre part, il s'agit de préciser les raisons qui ont poussé à choisir la fouille de graphes parmi les différentes solutions candidates pour aborder le problème de la fouille des BdR.

Le plan de ce chapitre et sa logique sont indiqués sur la figure 2.1. La section 2.1 commence

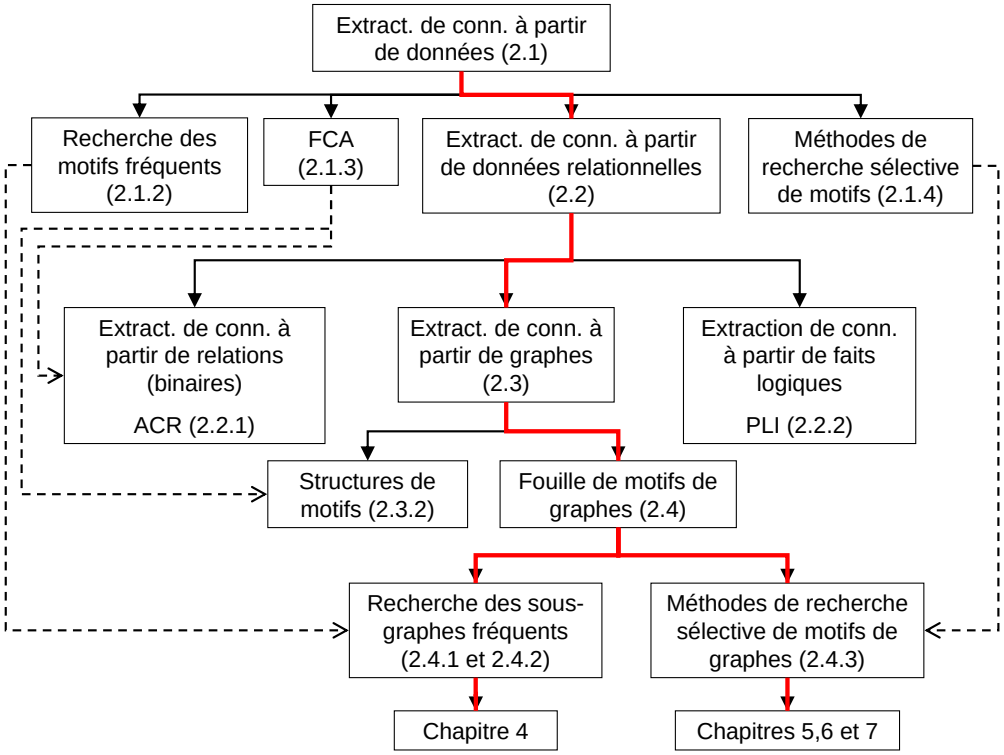


FIG. 2.1 – Plan du chapitre 2. Une flèche en pointillés indique l'extension d'un modèle. Le trait en gras rouge suit le fil conducteur qui relie l'état de l'art aux contributions des chapitres ultérieurs.

ainsi par résumer l'histoire et l'enjeu de la fouille de données en général, avant d'en préciser quelques méthodes ou formalismes qui ont en commun de considérer des données décrites

[5]Le terme de fouille de données relationnelles étant trop flou et recouvrant notamment les méthodes de fouille de graphes, le terme « fouille de relations » est introduit ici pour distinguer clairement les méthodes décrites dans la section 2.2.1, de celles décrites ultérieurement de fouille de graphes, c'est-à-dire qui recherchent explicitement des motifs prenant la forme de graphes.

par des attributs. Comme le souligne la figure 2.1 (cf lignes en pointillés), ces méthodes sont introduites pour avoir depuis été adaptées à des problèmes de fouille de graphes (i.e. la recherche des motifs fréquents) ou de fouille de relations entre objets (i.e. l'analyse de concepts formels ou ACF) ou encore qui ont un lien direct avec certains chapitres de ce mémoire (i.e. méthodes de recherche sélectives de motifs). La section 2.2 présente ensuite deux approches distinctes de la fouille de graphes mais capables de fouiller des données relationnelles (i.e. de descriptions d'objets munis de relations), que sont d'une part, l'analyse de concepts relationnels ou ACR (Huchard *et al.*, 2007) dérivée de l'ACF et d'autre part, la programmation logique inductive ou PLI (Muggleton, 1990). Cette section adopte un point de vue critique qui tient compte des exigences de la fouille des BdR aussi bien en terme d'expressivité des motifs extraits que d'efficacité des calculs. Il montre ainsi à travers des exemples que l'ACR ne permet pas d'exprimer facilement des motifs complexes (comme les motifs cycliques par ailleurs importants en chimie) et d'autre part que la PLI aborde des problèmes très difficiles et donc aussi très lourds en calcul. Ce point de vue critique prépare à la section 2.3 qui introduit les méthodes d'extraction de connaissances à partir de graphes comme une approche intermédiaire entre ACR et PLI, au sens où ces méthodes produisent des motifs sous forme de graphes qui sont compatibles avec les attentes des chimistes en étant à la fois plus « expressifs » que les motifs relationnels de l'ACR mais moins que les théories logiques de la PLI. Les problèmes théoriques et pratiques que pose la fouille de graphes sont ensuite abordés, et en particulier ceux liés à la notion d'isomorphisme et à l'absence de structure de treillis dans le cas des graphes. Ces deux derniers points permettent d'introduire une seconde extension de l'ACF, appelée structure de motifs (Ganter et Kuznetsov, 2001), qui permet théoriquement d'extraire à partir de données un treillis de concepts décrits par des motifs de graphes. Toutefois une analyse de la complexité de calcul pour extraire ce type de treillis laisse à penser que leur utilisation est difficile en pratique, sauf à faire des approximations (i.e. des projections, cf Ganter et Kuznetsov (2001)) qui sont trop restrictives dans le cas de la fouille de BdR. En conséquence la section 2.4 développe les méthodes de fouille de graphes comme principales sources d'inspiration pour concevoir les algorithmes présentés dans ce mémoire. En particulier les algorithmes de recherche des sous-graphes fréquents utilisés au chapitre 4 sont présentés comme des extensions successives des algorithmes de recherche de séquences puis d'arbres fréquents. Enfin le chapitre conclut sur différentes approches pour effectuer une sélection de motifs de graphes, en relation directe avec les chapitres 5, 6 et 7 de ce mémoire.

2.1 L'extraction de connaissances à partir de données

Après une présentation générale de la fouille de données, cette section introduit un certain nombre de notions et de méthodes pour fouiller des données tabulaires de type objets × attributs qui sont transposables ou réutilisables dans le cadre de la fouille de graphes. Sont d'abord présentées dans la section 2.1.2 la recherche des motifs fréquents et l'extraction des règles d'association fréquentes et ce pour deux raisons essentielles : tout d'abord, les algorithmes de recherche des motifs fréquents ont été la principale source d'inspiration pour mettre au point les premières méthodes de fouille de graphes ; ensuite, les premiers travaux (Berasaluce, 2002) de fouille des BdR présentés en section 3.3.2 reposent sur l'extraction des règles d'association fréquentes. La section 2.1.3 introduit ensuite l'analyse de concepts formels ou ACF, là aussi à plusieurs titres : d'une part, certains travaux de recherche ont eu pour objet de généraliser l'ACF à d'autres familles de motifs comme les graphes (cf section 2.3.2) et ont un temps, été étudiés dans le cadre de cette thèse en vue de leur utilisation éventuelle ; d'autre

part, l'ACF met en évidence certaines propriétés dont bénéficient les motifs d'attributs et dont ne disposent pas les graphes. L'introduction de l'ACF permet ainsi à la section 2.3.1 de mettre plus facilement en lumière les limites de l'analogie qui peut être faite entre motifs d'attributs et motifs de graphes. Enfin, l'ACF facilite l'introduction dans la section 2.1.4 des représentations condensées de motifs. Cette dernière section est importante à double titre : d'une part, cette section introduit des notions utiles comme les motifs fermés ou générateurs, qui se transposent au cas des graphes ; d'autre part, le chapitre 5 fait référence aux représentations condensées de motifs.

2.1.1 Introduction

Les moyens informatiques modernes ont permis de produire et d'archiver d'énormes masses de données numériques depuis maintenant au moins deux décennies. Si ces données sont généralement collectées pour rendre un service donné ou répondre à une question précise, incidemment ces données renferment aussi de nombreux éléments de connaissances relatifs aux objets qui y sont décrits. Le problème de l'accès à cette connaissance dépasse toutefois largement les capacités humaines d'analyse tant ces éléments — ou pépites — de connaissances sont disséminés dans une quantité importante de données souvent complexes. La mise à disposition de grandes quantités de données d'une part et l'impossibilité de les exploiter pleinement d'autre part, ont favorisé dès le début des années 90 l'essor d'une nouvelle discipline scientifique appelée tantôt *extraction de connaissances à partir de données*[6] (Lubinsky, 1989; Piatetsky-Shapiro et Frawley, 1991; Han *et al.*, 1992; Fayyad *et al.*, 1996a) par la communauté d'intelligence artificielle, tantôt *fouille de données* (Anwar *et al.*, 1992; Michalski *et al.*, 1992; Stonebraker *et al.*, 1993; Holsheimer *et al.*, 1995) par la communauté des bases de données. Ainsi selon Fayyad *et al.* (1996a), l'extraction de connaissances à partir de données se définit comme « l'extraction automatique de connaissances nouvelles, utiles et valides à partir de grandes quantités de données ». En terme de positionnement scientifique, la fouille de données se situe à l'intersection de l'informatique et des statistiques : les méthodes de fouille de données font appel à la fois aux statistiques et aux méthodes d'apprentissage automatique — dont l'objet est la conception d'algorithmes capables d'apprendre à résoudre un problème à partir d'exemples de solutions — pour induire des modèles qui soient représentatifs des données tout en étant robustes aux erreurs, et à la fois à la conception d'algorithmes efficaces pour pouvoir traiter de grandes quantités de données et considérer des questions plus ouvertes que celles posées antérieurement par l'analyse statistique de données. Ainsi contrairement à cette dernière, l'objectif de la fouille de données n'est pas d'évaluer la vraisemblance d'hypothèses mais d'élargir le champs de connaissances à travers tous les outils et modèles de représentation disponibles. La fouille de données recoupe ainsi des méthodes aussi variées que l'extraction de règles associatives, les méthodes de régression de fonctions ou de « clustering » d'objets similaires.

Fouiller des données ne sert à rien si les résultats de cette fouille ne sont pas interprétables. La fouille de données n'est donc qu'une étape d'un processus d'extraction de connaissances plus large qui s'étend de la préparation des données jusqu'à l'interprétation des résultats. La fouille de données établit par ce biais des connexions vers d'autres disciplines préexistantes telles que les bases de données, les outils de visualisation, les modèles de représentation de la connaissance et plus largement l'intelligence artificielle. L'intégration de la fouille de données au sein du processus d'extraction de connaissances tel que décrit dans Fayyad *et al.* (1996c)

[6]Le terme anglais *Knowledge Discovery in Databases* (KDD) apparaît en 1989 lors d'un atelier de la conférence IJCAI qui lui est consacré (Lubinsky, 1989).

est illustrée par le schéma de la figure 2.2. Ce processus décrit l'ensemble des étapes que

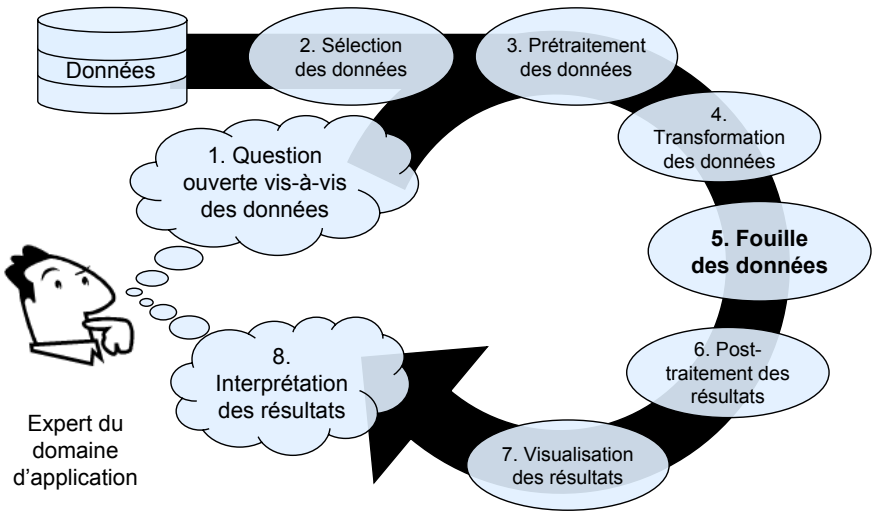


FIG. 2.2 – Le processus d'extraction de connaissances

l'utilisateur d'un système de fouille de données, qui est souvent aussi l'*expert* d'un certain domaine d'application, doit en pratique réaliser pour obtenir une réponse à son interrogation. L'étape de fouille de données proprement dite constitue l'étape centrale mais souvent la plus transparente du point de vue de l'utilisateur. Fayyad *et al.* (1996c) insistent notamment sur le caractère itératif et interactif du processus : l'expert ajuste ses questions au fil des réponses que lui renvoie le système de fouille de données. Rapporté à nos travaux, le cycle d'extraction de connaissances s'illustre le mieux à travers la prédiction des liaisons formables (cf chapitre 7) et leur analyse par Gilles Niel, expert en synthèse organique. Fayyad *et al.* soulignent également l'importance de l'étape de prétraitement pour filtrer, compléter et transformer les données, afin de permettre une utilisation optimale de la méthode de fouille de données. Cette remarque prend tout son sens au chapitre 4 tant le prétraitement des bases de données de réactions chimiques s'est révélé être une tâche délicate.

Les méthodes de fouille de données sont nombreuses et varient principalement en fonction d'une part, de la nature et de la quantité des données considérées et d'autre part, des questions auxquelles on cherche à répondre (Voir par exemple Hand *et al.*, 2001; Han, 2005). Nombre de ces méthodes ont été empruntées à l'apprentissage automatique, en particulier les méthodes de régression statistique, les méthodes de clustering comme par exemple les méthodes des plus proches voisins (Cover et Hart, 1967), ou de classification supervisée en particulier symbolique, comme les arbres de décision (Breiman *et al.*, 1984; Quinlan, 1986, 1993). La fouille de données est toutefois plus que l'assemblage hétéroclite de méthodes préexistantes : la fouille de données en tant que domaine de recherche identifié, s'est clairement affirmée quand furent proposées certaines méthodes qui ne se rattachaient plus aux grandes classes de problèmes

traités jusqu'alors par l'apprentissage automatique. Les problèmes de recherche des motifs fréquents et d'extraction des règles d'association introduits dans la section suivante, sont à ce titre emblématiques.

2.1.2 La recherche des motifs fréquents et l'extraction des règles d'association

Le problème de la recherche des motifs fréquents fut introduit par Agrawal *et al.* (Agrawal *et al.*, 1993b) et résolu grâce à son algorithme `Apriori`. Ces travaux s'annoncent dès le premier article (Agrawal *et al.*, 1993b) en rupture avec les travaux d'apprentissage symbolique réalisés jusque là : il ne s'agit pas en effet de traiter un problème de classification, que cette classification soit supervisée ou non. La méthode ne tente pas non plus de décrire globalement les données, comme peuvent le faire les méthodes de régression, mais produit des règles d'association « locales » décrivant chacune un sous-ensemble réduit des données. Enfin contrairement aux systèmes existants de classification à base de règles (Breiman *et al.*, 1984; Piatetsky-Shapiro, 1991; Han *et al.*, 1992; Quinlan, 1993; Fayyad *et al.*, 1993), la méthode propose un algorithme complet pour extraire un grand nombre de règles significatives tout en traitant de grandes quantités de données.

Le problème de la recherche des motifs d'attributs fréquents

Dans son formalisme initial, la recherche des motifs fréquents considère un ensemble $\mathcal{O}$ de n *objets* (ou transactions) décrits par un ensemble $\mathcal{A}$ de m *attributs* selon une relation binaire $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{A}$. Un exemple d'une telle relation binaire est représenté par la table de la figure 2.3. Les colonnes représentent les attributs de A à D alors que les lignes représentent les objets de o_1 à o_5. Une croix indique quels sont les attributs qui sont en relation avec un objet. La

$\mathcal{R}$	A	B	C	D
o_1	×	×	×	×
o_2	×	×	×	
o_3			×	×
o_4	×		×	
o_5			×	

FIG. 2.3 – Un exemple de relation binaire

donnée de cette relation binaire permet de déterminer l'ensemble des objets présentant un ensemble donné d'attributs. L'objectif de la recherche des motifs fréquents est de déterminer les cooccurrences d'attributs qui apparaissent le plus fréquemment dans les données.

Définition 2.1.1. Soient les définitions suivantes :

1. Un *motif* est un ensemble d'attributs.
2. Un motif $M \subseteq \mathcal{A}$ *décrit* ou *couvre* un objet de $\mathcal{O}$ si cet objet présente (i.e. est en relation avec) tous les attributs éléments de M.
3. Le *support* $\text{support}(M)$ du motif M est le nombre d'objets décrits par M : $\text{support}(M) = |o \in \mathcal{O} | \forall a \in M, o\mathcal{R}a|$[7]. Le support porte parfois aussi le nom de *fréquence absolue*.

[7]Les notations utilisées dans ce mémoire sont résumées en annexe A.

4. La *fréquence relative* d'un motif M notée $\text{freq}_r(M)$[8] est la fraction du support de M sur le nombre n d'objets et est donc comprise entre 0 et 1.

Le problème de la *recherche des motifs fréquents* consiste alors, étant donné un seuil minimal f_{min} de fréquence relative ou absolue, à déterminer la fréquence de tous les *motifs fréquents*, c'est-à-dire dont la fréquence est supérieure ou égale à f_{min}.

Ainsi le motif $\{A, C\}$, noté AC sous forme abrégée, a pour support $|\{o_1, o_2, o_4\}| = 3$ et pour fréquence relative 3/5. La fréquence présente la particularité d'être une fonction décroissante dans l'ordre des motifs ordonné par la relation $\subseteq$ d'inclusion ensembliste (i.e. $M_1 \subseteq M_2 \subseteq \mathcal{A} \Rightarrow \text{freq}(M_1) \geq \text{freq}(M_2)$). Cette propriété est visible sur le *diagramme de l'ordre des motifs* (parfois appelé diagramme de Hasse) représenté sur la figure 2.4. On y observe la décroissance de la fréquence des motifs indiquée entre parenthèses, le long de tout chemin descendant du sommet représentant le motif vide. Cette propriété de décroissance sur

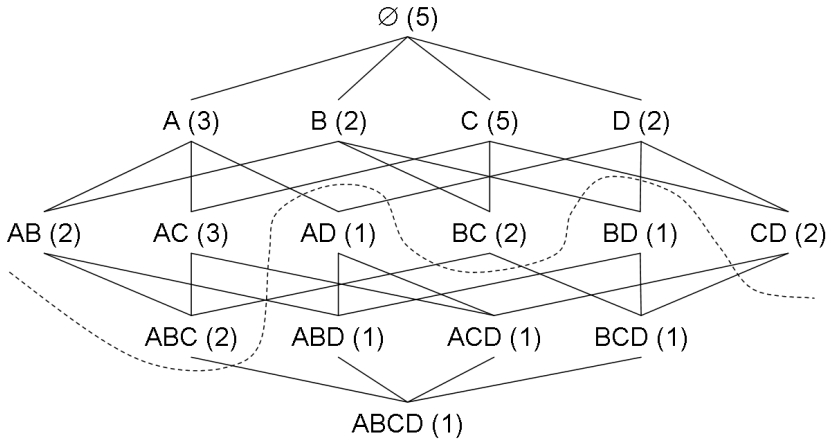


FIG. 2.4 – La décroissance des fréquences au sein de l'ordre des motifs

les fréquences implique la propriété d'*anti-monotonie* sur le caractère fréquent d'un motif : un motif ne peut être fréquent que si tous les motifs qu'il inclut sont eux-mêmes fréquents (i.e. $M_1 \subseteq M_2 \subseteq \mathcal{A}$ et $\text{freq}_r(M_2) \geq f_{min} \Rightarrow \text{freq}_r(M_1) \geq f_{min}$). L'ensemble des motifs de support supérieur ou égal à 2 sont ainsi $\{\emptyset, A, B, C, D, AB, AC, BC, CD, ABC\}$ et représente la partie supérieure du diagramme d'ordre au dessus de la ligne de partage en pointillés.

Au delà de son approche novatrice, le problème de la recherche des motifs fréquents s'attaque également à un problème informatique difficile : dans le pire cas où $f_{min} = 1$ et où les données sont telles que tout motif est décrit par au moins un objet, tous les 2^m motifs possibles deviennent fréquents et doivent par conséquent être fouillés. Chaque calcul de fréquence nécessitant de tester l'inclusion du motif dans chaque description d'objet, la complexité du problème dans le pire cas est (au moins) exponentielle en le nombre m d'attributs. Or les algorithmes de complexité exponentielle sont généralement considérés comme inutilisables en informatique théorique. La recherche des motifs fréquents apporte une réponse pragmatique à ce problème puisque le réglage du paramètre f_{min} permet d'ajuster le nombre de motifs fréquents qu'il est possible d'extraire en un temps raisonnable. En pratique l'ajustement de

[8]La fréquence relative se définit relativement à la taille des données par opposition à la fréquence absolue. Lorsque la distinction entre fréquences absolue et relative est inutile, le terme *fréquence* et la notation $\text{freq}(M)$ seront utilisés sans autre précision.

ce paramètre est un facteur essentiel puisqu'il permet de s'adapter à la difficulté variable qu'il existe à fouiller un jeu de données, selon que ce dernier soit creux (i.e. les données contiennent relativement peu de motifs de fréquence élevée) ou au contraire dense. Il n'en demeure pas moins que le problème reste difficile et devient « non faisable » dès que le seuil f_{min} devient trop faible ou que les données sont trop denses. Si la complexité du problème est défavorable vis à vis du nombre d'attributs, elle est au contraire extrêmement avantageuse du point de vue de la taille des données puisqu'elle est une fonction linéaire du nombre n d'exemples (pour un nombre de motifs fréquents constant). Cette capacité à traiter rapidement de grandes masses de données est une autre caractéristique déterminante des méthodes de fouille de données.

Avant de décrire les algorithmes de recherche des motifs fréquents, il est important de préciser que la recherche de ces motifs fréquents est rarement un but en soi. Le nombre de motifs potentiellement fréquents croit exponentiellement avec le nombre d'attributs considérés et il n'est pas rare en pratique de produire des centaines de milliers voire des millions de motifs fréquents. Par ailleurs la fréquence n'est pas d'une grande utilité pour identifier les motifs intéressants puisque sa valeur obéit principalement à la longueur des motifs : les motifs les plus courts auront tendance à être les plus fréquents. Le motif vide est ainsi le plus fréquent alors qu'il n'apporte aucune information. Les motifs fréquents sont donc généralement suivis d'un post-traitement dont l'object est la sélection d'un ensemble réduit de motifs. Cette sélection peut par exemple se faire grâce à une fonction de score pertinente vis-à-vis de l'application traitée, en triant les motifs par ordre décroissant de score. Mais dans la plupart des applications, les motifs fréquents servent d'intermédiaires de calcul pour extraire efficacement les règles d'association fréquentes introduites à la section suivante.

L'extraction des règles d'association fréquentes

Intuitivement une règle d'association $H \rightarrow C$ construite à partir de deux motifs disjoints H et C permet d'exprimer une relation de causalité ou du moins de corrélation entre les attributs de H et ceux de C. Pour cette raison les règles d'associations sont plus expressives pour les experts que ne le sont les motifs fréquents. Une règle n'a toutefois d'intérêt que si on lui joint sa confiance, c'est à dire la probabilité conditionnelle pour qu'un objet présentant le motif H présente aussi le motif C.

Définition 2.1.2. Formellement,

1. Une *règle d'association* $H \rightarrow C$ est définie par un *motif hypothèse* H (ou prémisse) et un *motif conclusion* C disjoint de l'hypothèse. Elle exprime le degré de vraisemblance selon lequel un objet présentant les attributs du motif H présente aussi ceux de C.
2. Le *support* $\text{support}(H \rightarrow C)$ de la règle $H \rightarrow C$ est le support de $H \cup C$. Le support d'une règle permet d'évaluer sa représentativité dans les données.
3. La *confiance* $\text{conf}(H \rightarrow C)$ d'une règle $H \rightarrow C$ est le rapport du support de $H \cup C$ sur celui de H. La confiance représente le degré de vraisemblance ou précision de la règle.

Ainsi sur l'exemple de la figure 2.4, la règle d'association $A \rightarrow BC$ a pour confiance $2/3$ et pour support 2. Le problème de la *recherche des règles d'association fréquentes* consiste alors, étant donnés des seuils minimaux f_{min} de fréquence et c_{min} de confiance choisis arbitrairement entre 0 et 1, à déterminer la fréquence et la confiance de toutes les *règles fréquentes*, c'est-à-dire dont la fréquence relative et la confiance sont respectivement supérieures ou égales à f_{min} et c_{min}.

L'objectif est d'extraire des données un ensemble de règles qui soient en premier lieu précises (i.e. de confiance élevée) et en second lieu représentatives (i.e. de support élevé). Une règle $H \rightarrow C$ est fréquente vis-à-vis des seuils f_{min} et c_{min} si le motif $M = H \cup C$ est fréquent relativement à f_{min} et si pour un motif M fréquent donné, le motif $H = M \setminus C$ est tel que $\text{freq}_r(H) \leq f'_{max}$ pour le seuil $f'_{max} = c_{min}^{-1} \times \text{freq}_r(M)$. À partir de cette observation, Agrawal *et al.* (1996) extraient efficacement les règles fréquentes en traitant cette extraction comme deux problèmes imbriqués de recherche de motifs définis par une contrainte anti-monotone : le premier consiste à chercher tous les motifs fréquents relativement à f_{min} puis pour chaque motif M fréquent trouvé, le second consiste à chercher tous les motifs C inclus dans M tels que $\text{freq}_r(M \setminus C) \leq f'_{max}$.

Les algorithmes de recherche des motifs d'attributs fréquents

Le principe fondamental de l'algorithme `Apriori` (Agrawal et Srikant, 1994) est d'exploiter la propriété d'anti-monotonie des motifs fréquents pour limiter au maximum le nombre de fréquences à calculer. L'idée consiste à générer des motifs de longueur $l+1$ candidats à être fréquents résultant de l'union de motifs fréquents de longueur l partageant $l-1$ attributs. Parmi ces motifs candidats seuls sont retenus les motifs dont tous les sous-motifs de longueur l se sont révélés préalablement fréquents, toujours d'après la propriété d'anti-monotonie des motifs fréquents. Les fréquences de ces motifs candidats sont ensuite calculées en une seule passe dans la base de données. Seuls les motifs candidats fréquents sont conservés et la procédure est réitérée pour les motifs de longueur $l+2$ jusqu'à épuisement des motifs fréquents. La force de l'algorithme tient dans la réduction du nombre des calculs de fréquence, passant du nombre $2^{|\mathcal{A}|}$ total de motifs à la somme du nombre de motifs fréquents et du nombre de motifs dans la frontière négative[9] (Boulicaut *et al.*, 2003). La *frontière négative* associée ici au caractère fréquent des motifs mais généralisable à tout autre prédicat anti-monotone, est l'ensemble des motifs non fréquents minimaux, c'est-à-dire des motifs non fréquents dont tous les prédécesseurs immédiats[10] sont fréquents. C'est la raison pour laquelle l'algorithme fournit en pratique le résultat en un temps raisonnable tant que le seuil minimal f_{min} de fréquence reste suffisamment élevé.

`Apriori` est un algorithme par niveau considérant successivement les générations – ou niveaux d'ordre k – des motifs candidats de longueur k. Depuis, de nombreux autres algorithmes ont été proposés qui se sont révélés plus efficaces que `Apriori`. En particulier les algorithmes de recherche en profondeur comme `Eclat` (Zaki *et al.*, 1997; Zaki, 2000) énumèrent les motifs fréquents selon un parcours en profondeur de l'ordre des motifs. Afin d'éviter des générations redondantes de motifs, ce parcours en profondeur se construit en ordonnant les attributs des motifs selon un ordre lexicographique : le motif $\{c; a; g; h\}$ se représente par la séquence triée de ses attribut (a, c, g, h) si on classe les attributs par ordre alphabétique. Cet ordre total permet de confier la génération d'un motif à un motif parent unique : le motif $acgh$ est généré par son motif parent acg si ce dernier est fréquent et par lui seul. Ce principe évident dans le cas des motifs d'attributs devient un problème complexe et essentiel de la fouille de graphes. L'avantage des algorithmes en profondeur est de pouvoir calculer très rapidement le support du motif courant M en mémorisant la liste $L(M)$ des transactions contenant M. Grâce à un codage vertical des données associant à chaque attribut a la liste $L(\{a\})$ des transactions présentant cet attribut, il est alors possible de calculer très rapidement le support

[9] Negative border en anglais

[10] Un motif M_1 est un *prédécesseur immédiat* d'un motif M_2 vis à vis d'une relation d'ordre $\subseteq$ et se note $M_1 \prec M_2$ si $M_1 \subset M_2$ et s'il n'existe pas de motifs dans l'intervalle $]M_1; M_2[$: $M_1 \subseteq M \subset M_2 \Rightarrow M = M_1$.

des motifs $M \cup \{a\}$ résultant de l'extension d'un attribut au motif courant comme étant égal à $|L(M) \cap L(\{a\})|$[11]. Enfin l'algorithme `FP-growth` (Han *et al.*, 2004) utilise une approche hybride fondée sur une structure FP-Tree. Cet arbre préfixé enrichi de pointeurs supplémentaires permet de compresser en mémoire l'ensemble des transactions de telle manière que la fréquence de tout motif s'en déduise rapidement.

2.1.3 L'analyse de concepts formels

La recherche des motifs d'attributs fréquents est liée à l'*analyse de concepts formels*[12] ou ACF dont l'objet est d'étudier le problème de l'extraction et de la représentation des connaissances sous l'angle de la théorie mathématique des treillis développée par Birkhoff (1940) dans les années 30 et en particulier par celle des treillis de Galois (Barbut et Monjardet, 1970). L'ACF (Ganter et Wille, 1999; Ganter *et al.*, 2005) a été introduite par Wille en 1982 et appliquée à « l'acquisition automatique de connaissances » (Wille, 1989) avant même que ne soit posé le problème de la recherche des motifs d'attributs fréquents. Tout comme la recherche des motifs fréquents, l'ACF considère un *contexte* $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ constitué d'une relation binaire $\mathcal{R}$ entre un ensemble d'objets $\mathcal{O}$ et un ensemble d'attributs $\mathcal{A}$. L'ACF exploite une correspondance de Galois qui s'établit entre l'ordre des motifs d'attributs $(2^{\mathcal{A}}, \subseteq)$ et l'ordre des ensembles d'objets $(2^{\mathcal{O}}, \subseteq)$, tous deux ordonnés par la relation d'inclusion, pour extraire du contexte un treillis de concepts dits formels. Précisément, étant donné un contexte $(\mathcal{O}, \mathcal{A}, \mathcal{R})$, l'ACF introduit deux fonctions p et q permettant respectivement de passer d'un motif d'attributs aux objets que ce motif décrit et inversement d'un ensemble d'objets au motif d'attributs commun à tous ces objets. Formellement :

$$p : M \subseteq \mathcal{A} \mapsto p(M) = \{o \in \mathcal{O} | a \in M \Rightarrow o\mathcal{R}a\} \text{ et } q : O \subseteq \mathcal{O} \mapsto q(O) = \{a \in \mathcal{A} | o \in O \Rightarrow o\mathcal{R}a\}$$

Le couple (p, q) définit alors une *correspondance de Galois* permettant d'exprimer la dualité qui existe entre les ordres $(2^{\mathcal{A}}, \subseteq)$ et $(2^{\mathcal{O}}, \subseteq)$ des sous-ensembles d'attributs et d'objets :

Propriété 2.1.3. *Le couple (p, q) de fonctions définit une correspondance de Galois :*

1. *p et q sont des fonctions décroissantes : $M_1 \subseteq M_2 \subseteq \mathcal{A} \Rightarrow p(M_1) \supseteq p(M_2)$ et $O_1 \subseteq O_2 \subseteq \mathcal{O} \Rightarrow q(O_1) \supseteq q(O_2)$*
2. *Les fonctions composées $f = q \circ p : 2^{\mathcal{A}} \to 2^{\mathcal{A}}$ et $g = p \circ q : 2^{\mathcal{O}} \to 2^{\mathcal{O}}$ sont extensives : $\forall M \subseteq \mathcal{A}, f(M) \supseteq M$ et $\forall O \subseteq \mathcal{O}, g(O) \supseteq O$*

La correspondance de Galois fait des deux fonctions f et g des opérateurs de fermeture :

Propriété 2.1.4. *Les opérateurs $f = q \circ p$ et $g = p \circ q$ sont des opérateurs de fermeture, c'est-à-dire des fonctions croissantes, extensives et idempotentes :*

$$\forall M_1 \subseteq \mathcal{A}, \forall M_2 \in \mathcal{A}, \quad M_1 \subseteq M_2 \Rightarrow f(M_1) \subseteq f(M_2) \quad \textit{(croissance)} \tag{2.1}$$

$$\forall M \subseteq \mathcal{A}, \quad f(M) \supseteq M \quad \textit{(extensitivité)} \tag{2.2}$$

$$\forall M \subseteq \mathcal{A}, \quad f(f(M)) = f(M) \quad \textit{(idempotence)} \tag{2.3}$$

Les fermés *associés à f sont alors les éléments stables de f (i.e. les éléments e tels que $f(e) = e$).*

[11] La notation $|E|$ désigne le cardinal de l'ensemble E.
[12] *Formal Concept Analysis* (FCA) en anglais.

Les couples $(M, p(M))$ des motifs M fermés de $q \circ p$ auxquels sont adjoints les ensembles $p(M)$ des objets qu'ils décrivent, forment l'ensemble des *concepts formels*. M et $p(M)$ sont respectivement appelés l'*intension* et l'*extension* du concept[13]. Les concepts formels peuvent s'obtenir par fermeture des motifs, c'est-à-dire comme l'ensemble des valeurs que prend $(q(p(M)), p(M))$ lorsque M décrit tous les sous-ensembles possibles d'attributs. Intuitivement les concepts formels correspondent aux rectangles pleins maximaux dans la représentation tabulaire de $\mathcal{R}$ (à une permutation des colonnes et des lignes près).

L'ordre des concepts formels peut être muni d'opérateurs d'union et d'intersection dotés des propriétés algébriques propres aux treillis commutatifs.

Définition 2.1.5. L'ensemble $\mathcal{C}$ des concepts formels est un treillis commutatif $(\mathcal{C}, \vee, \wedge)$ où :

1. L'intersection de deux concepts est définie par $(M_1, O_1) \wedge (M_2, O_2) = (q(O_1 \cap O_2), O_1 \cap O_2)$.
2. L'union de deux concepts est définie par $(M_1, O_1) \vee (M_2, O_2) = (M_1 \cap M_2, p(M_1 \cap M_2))$.

L'intérêt fondamental de l'ACF est de transformer la vue tabulaire peu expressive du contexte en un treillis de concepts adapté à des tâches d'extraction et de représentation des connaissances, comme l'extraction automatique d'ontologies, la classification automatique d'objets ou la recherche d'information (Ferré, 2002). Ainsi le contexte de la figure 2.3 se transforme en un treillis de concepts représenté par le diagramme de Hasse de la figure 2.5. À chaque concept sont associées son intension et son extension (par exemple l'intension $AC = \{A, C\}$ et l'extension $124 = \{o_1, o_2, o_4\}$). Lorsque les données sont creuses, le treillis de concepts formels

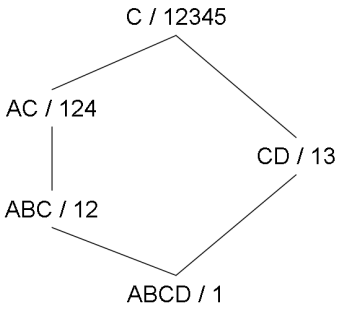


Fig. 2.5 – Treillis de concepts formels

est en général beaucoup plus petit que le treillis des motifs associés à leur fréquence. Ainsi le treillis de la figure 2.5 comporte 5 concepts là où l'ordre des motifs de la figure 2.4 ne compte pas moins de 16 motifs. Ces deux représentations sont pourtant équivalentes puisqu'il est possible de retrouver la table des données initiales à partir soit du treillis des concepts, soit des fréquences de tous les motifs. Par rapport à la donnée des motifs et de leur fréquence, les treillis de concepts formels apportent une représentation plus condensée des données, concept qui a été repris ultérieurement par les méthodes de recherche des motifs fermés et générateurs présentées dans la section suivante.

[13] La dualité intension - extension sont des termes empruntés aux langages de représentation des connaissances.

2.1.4 Les méthodes de recherche sélective de motifs

Les représentations condensées de motifs

L'extraction des règles d'association fréquentes produit en pratique un nombre très important de règles, entraînant deux conséquences néfastes : d'une part le passage en revue des règles par l'expert devient une tâche fastidieuse, d'autre part l'information utile moyenne que présente chaque règle s'en trouve affaiblie du fait de la redondance d'information entre les règles. Cette constatation a motivé le remplacement de l'ensemble des règles d'association fréquentes par une base de règles réduite mais équivalente (i.e. sans perte d'information). L'ACF s'est révélée être une théorie très utile pour répondre à ce problème. L'opérateur de fermeture $f = q \circ p$ de la section 2.1.3 définit en effet une relation d'équivalence entre motifs. Deux motifs sont dits équivalents s'ils ont même image par f. Les classes d'équivalence associées sont donc en bijection avec les concepts formels de l'ACF. Chaque classe d'équivalence contient des éléments minimaux et des éléments maximaux au sens de l'inclusion $\subseteq$. Du fait des propriétés des opérateurs de fermeture, chaque classe d'équivalence C a un et un seul élément maximal qui est le motif fermé $f(M)$ pour tout motif M de C. Plusieurs motifs minimaux encore appelés motifs libres ou générateurs, peuvent au contraire coexister au sein d'une même classe. Les classes d'équivalence des motifs de la figure 2.4 sont précisées sur la figure 2.6. Chacune des cinq classes est en bijection avec un concept formel et regroupe un certain nombre de motifs dont un motif fermé maximal (en gras sur la figure), un ou plusieurs motifs générateurs minimaux (en italique) et éventuellement d'autres motifs qui ne sont ni fermés ni générateurs. Ainsi le concept $(ABCD, 1)$ est associé à la classe d'équivalence $\{AD; BD; ABD; ACD; BCD; ABCD\}$ qui a pour motif fermé $ABCD$ et pour motifs générateurs AD et BD. Les motifs fermés fréquents munis de leurs fréquences (ou les générateurs

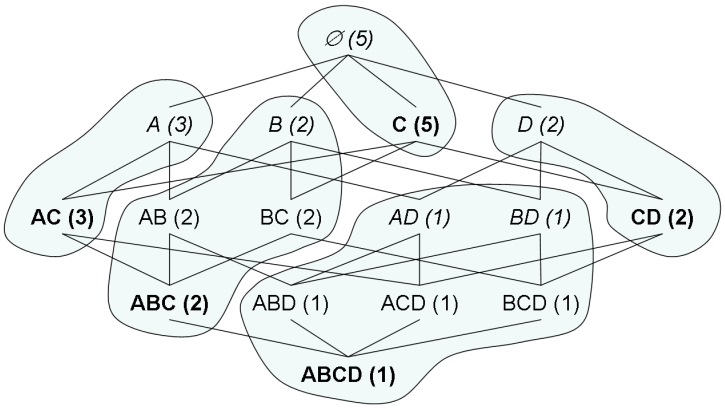


FIG. 2.6 – Classes d'équivalence de motifs. Les motifs représentés en gras (resp. en italique) sont les motifs fermés (resp. générateurs), c'est-à-dire les motifs maximaux (resp. minimaux) dans leur classe d'équivalence.

fréquents plus la frontière négative) forment une représentation dite *condensée*, c'est-à-dire réduite mais équivalente des motifs fréquents. La donnée d'une telle représentation permet en effet d'en déduire la fréquence de tout motif fréquent. Si on relâche l'exigence d'équivalence, il est possible de produire des représentations condensées approximatives comme celle des motifs δ-libres (Boulicaut *et al.*, 2003) qui généralise la notion de motifs générateurs. Plusieurs algorithmes permettent d'extraire efficacement les motifs fermés et générateurs comme `Close` (Pasquier *et al.*, 1999b), `Close-A` (Pasquier *et al.*, 1999a), `Pascal` (Bastide *et al.*, 2000b) et

`Charm` (Zaki et Hsiao, 2002), `Zart` (Szathmary *et al.*, 2007) et `Snow` (Szathmary *et al.*, 2008).

Cette notion de représentation condensée se propage ensuite aux règles d'association : à partir des motifs fermés et générateurs fréquents, peut être construit l'ensemble réduit des *règles d'association minimales non redondantes* fréquentes (Bastide *et al.*, 2000a; Pasquier *et al.*, 2005). Une telle règle $H \rightarrow C$ est telle que $H \cup C$ soit un motif fermé et H un motif générateur. Cet ensemble de règles est une représentation condensée des règles d'association fréquentes.

La recherche de motifs sous contraintes

La recherche de motifs sous contraintes (Ng *et al.*, 1998; Raedt *et al.*, 2008b) est une approche complémentaire des représentations condensées de motifs qui introduit des contraintes spécifiques dans le processus de fouille de données, afin de sélectionner un sous-ensemble restreint de motifs pertinents. De façon générale, une contrainte est un prédicat prenant comme argument un motif : seuls les motifs vérifiant ce prédicat sont sélectionnés pour figurer dans l'ensemble résultat. La vérification de la satisfaction des contraintes peut se faire lors de la phase de post-traitement qui suit la recherche des motifs fréquents. Une telle solution est cependant peu économique en temps de calcul voire irréalisable, tant le nombre de motifs fréquents peut être élevé vis-à-vis du nombre de motifs retenus in fine. L'idée de la recherche de motifs sous contraintes est donc d'intégrer les contraintes dans la phase d'extraction des motifs afin d'élaguer certaines branches de l'espace de recherche et ce faisant, économiser du temps de calcul et repousser les limites de ce qui est « extractible ». Pour ce faire les algorithmes tirent parti des propriétés spécifiques des contraintes considérées. Parmi ces propriétés, l'anti-monotonie d'un prédicat dans l'ordre des motifs est très efficace et facile à prendre en compte. D'autres familles de contraintes peuvent toutefois être intégrées dans le processus de fouille, comme les contraintes monotones ou convertibles (Pei *et al.*, 2004a). Certaines contraintes intéressantes dans l'absolu sont toutefois plus difficiles à traiter : c'est par exemple le cas des motifs émergents (Dong et Li, 1999; Bailey *et al.*, 2002) dont le taux de croissance, défini comme le rapport $freq_r(M, \mathcal{D}^+)/freq_r(M, \mathcal{D}^-)$ des fréquences du motif M dans des ensembles $\mathcal{D}^+$ et $\mathcal{D}^-$ d'exemples positifs et négatifs, est supérieur un certain seuil ρ.

La recherche sous contraintes est étroitement liée aux bases de données inductives (Imielinski et Mannila, 1996) dont le langage de requête permet de rechercher des données particulières mais aussi des motifs partagés par ces données. Pour ce faire, les bases de données inductives doivent intégrer en leur sein des algorithmes de recherche de motifs qui puissent être contraints par la requête posée. Dans la mesure où le nombre de contraintes (i.e. les prédicats) envisageables et a fortiori le nombre de leurs combinaisons logiques sont potentiellement infinis, certains travaux (Raedt *et al.*, 2002; Wang *et al.*, 2003; Soulet et Crémilleux, 2005; Soulet et Crémilleux, 2008; Raedt *et al.*, 2008b) ont proposé des modèles intégrant de façon flexible et unifié certaines familles de contraintes sans que cette souplesse d'intégration ne se fasse nécessairement au détriment des performances. Soulet et Crémilleux (2008) proposent par exemple de généraliser les représentations condensées de motifs à d'autres fonctions que la fréquence, appelées fonctions condensables. Citons également Raedt *et al.* (2008b) qui abordent le problème de la recherche des motifs fréquents comme un problème de programmation de satisfaction de contraintes (CSP), ce qui présente l'avantage d'intégrer naturellement différents types de contraintes tout en tirant parti de l'efficacité des solveurs CSP existants. En pratique les motifs sous contraintes restent parfois très pour ne pas dire trop nombreux pour être analysés par un expert. Une des façons de réduire ce nombre de mo-

tifs k à analyser est de le fixer à l'avance. Cette approche dite des top-k motifs revient alors à extraire les k meilleurs motifs qui maximisent une fonction de score (Wang *et al.*, 2005b; Xin *et al.*, 2006; Soulet et Crémilleux, 2007) en plus de satisfaire d'éventuelles contraintes. Enfin il est intéressant de noter qu'un certain nombre de travaux récents cherchent à produire un nombre réduit de motifs non redondants (Xin *et al.*, 2005; Siebes *et al.*, 2006; van Leeuwen *et al.*, 2006; Bringmann et Zimmermann, 2007). Plus récemment Crémilleux et Soulet (2008) ont soulevé l'intérêt qui existe à faire coexister des contraintes locales (i.e. portant uniquement sur le motif considéré, comme par exemple le caractère fréquent d'un motif) avec des contraintes dites globales (i.e. nécessitant la comparaison de plusieurs motifs, comme le caractère fermé ou générateur d'un motif). Cette idée est exactement celle qui a été développée parallèlement au sein du modèle des motifs les plus informatifs présenté au chapitre 5. Ce modèle propose en effet de concilier les avantages des méthodes de sélection de motifs dites intra-motifs et inter-motifs, soit respectivement de contraintes locales et globales.

En conclusion, les motifs d'attributs fréquents disposent de nombreuses méthodes pour les extraire des données (cf la recherche des motifs fréquents), les interpréter (cf l'extraction des règles d'association), les représenter et les organiser (cf l'ACF) ou encore les sélectionner selon divers critères et contraintes. Indépendamment des recherches qui visent à enrichir la fouille de données de nouvelles méthodes ou à améliorer l'efficacité des méthodes existantes, un autre axe de recherche s'est développé pour adapter les méthodes conçues au départ pour des données tabulaires à des données plus complexes, et plus particulièrement aux données décrites par des relations. La *fouille de données relationnelles*, vue comme un grand ensemble regroupant aussi bien les méthodes de fouille ou d'apprentissage à partir de relations, les méthodes de fouille de graphes ou encore la programmation logique inductive, fait l'objet de la section suivante.

2.2 L'extraction de connaissances à partir de données relationnelles

Les plus efficaces des algorithmes de recherche des motifs fréquents, qui plus est, exécutés sur des ordinateurs toujours plus rapides permettent de traiter en pratique bon nombre d'applications. La principale limitation de ces méthodes ne provient plus alors d'une insuffisance de performance mais du manque d'expressivité que présentent les tables de correspondance entre objets et attributs. En effet dans certains domaines d'application, les données ne se représentent pas naturellement sous la forme d'une conjonction d'attributs. L'expert est alors obligé de projeter ces données dans une table objets-attributs et ce faisant, de perdre tout ou partie de l'information qu'il comptait fouiller. La recherche des motifs d'attributs fréquents dans les bases de données de réactions (Berasaluce, 2002) est un exemple d'une telle projection où certains sous-graphes particuliers sont réduits à l'état d'attributs (cf section 3.3.2).

Parmi les plus importantes lacunes d'un contexte tabulaire $(\mathcal{O}, \mathcal{A}, \mathcal{R})$, est le fait de ne pouvoir prendre en compte l'existence des relations entre objets. La suite de cette section montre qu'il n'existe pas une seule manière de prendre en compte ces relations mais plusieurs, dont la complexité varie en fonction des exigences de l'application traitée. Ainsi dans certains cas, l'analyse au niveau des relations entre classes d'objets (ou concepts) plutôt qu'au niveau des relations entre objets eux-mêmes, peut suffire. Ce niveau d'analyse, appelé ici fouille de relations, est développé dans la section 2.2.1. À l'opposé, les relations peuvent être interprétées comme des observations de prédicats logiques extrêmement généraux. L'enjeu est alors d'induire de ces données une théorie logique la plus générale et la plus cohérente possible.

C'est l'objet de la programmation logique inductive développée dans la section 2.2.2. Entre ces deux extrêmes, un compromis existe, qui s'intéresse à la fouille de motifs structuraux tels que des séquences, des arbres ou des graphes. La section 2.3 est entièrement consacrée à cette dernière approche.

2.2.1 L'extraction de connaissances à partir de relations

Vers des motifs plus complexes

De nombreuses propositions ont été faites pour étendre les modèles fondés sur les relations binaires objets × attributs à des objets décrits par des attributs plus complexes. Les travaux réalisés dans le domaine de l'analyse de concepts formels font figure de référence dans la mesure où ils s'appuient sur un socle théorique solide. Ainsi le principe de *Conceptual Scaling* (Ganter et Wille, 1999, p. 36-45) a permis d'intégrer au sein de l'ACF le cas des attributs multi-valués. Un attribut multi-valué est un attribut associant à chaque objet une valeur prise dans un domaine de définition, par opposition aux attributs mono-valués qui sont en relation ou non avec chaque objet. Le Conceptual Scaling consiste à doter chaque attribut multi-valué d'un treillis adéquat définissant les relations de spécialisation entre les différentes valeurs (interprétées comme des concepts) que peut prendre l'attribut, puis par passage au treillis produit, à obtenir un treillis de concepts formels bénéficiant alors des outils standards de l'ACF. Polaillon et Diday (Polaillon et Diday, 1997; Polaillon, 1998) a ainsi étendu l'ACF au cas d'attributs de type intervalle ou histogramme. Il n'en demeure pas moins que l'ordre des motifs considéré reste l'ordre produit[14] d'ordres indépendants associés à chaque attribut multi-valué. Que se passe t-il si ces attributs ne sont plus des dimensions indépendantes mais sont liés par des relations binaires voire n-aires ? Là encore le problème peut se ramener à un ensemble d'attributs mono-valués.

La prise en compte des relations entre attributs et les graphes relationnels

Dans certaines applications, les attributs peuvent être liés par des relations. Dans le cas de relations binaires, les attributs forment alors un graphe orienté où les sommets et les arcs représentent respectivement les attributs et les relations entre couples d'attributs. Chaque arc est étiqueté par la relation qu'il représente. Dans le cas de relations d'arité quelconque, le graphe devient un hypergraphe où les hyper-arêtes sont les tuples d'attributs mis en relation. Chaque objet est alors associé à une description sous forme de graphe ou d'hypergraphe. Ces graphes ou hypergraphes présentent toutefois la particularité d'être superposables facilement entre eux et de façon unique puisque chaque sommet représente un attribut singulier. La figure 2.7 représente un ensemble d'objets pour lesquels les attributs qui les décrivent sont liés par différentes relations binaires (représentées par des arcs de couleurs). Ainsi sur la figure 2.7, l'objet o_2 est décrit par un attribut a ayant une relation de type α avec l'attribut e, un attribut g en relation de type β avec h lui même en relation avec g selon β et deux attributs isolés b et c. Si chaque objet est décrit par le même ensemble d'attributs et de relations, les objets (i.e. o_1, o_2 et o_3) sont toutefois indépendants au sens où il n'existe pas de relations liant ces objets. De tels graphes superposables sont parfois appelés *graphes relationnels* et sont utiles pour représenter des portions d'un réseau où chaque nœud est un objet singulier unique. Ces graphes relationnels jouent ainsi un rôle important dans de nombreuses applications ayant trait aux réseaux sociaux, aux réseaux biologiques ou encore aux

[14] L'ordre produit de deux ordres $(E_1, \leq_1)$ et $(E_2, \leq_2)$, est l'ordre $(E_1 \times E_2, \leq_{12})$ où $E_1 \times E_2$ est le produit cartésien de E_1 et E_2 et où $(a_1, a_2) \leq_{12} (b_1, b_2)$ si et seulement si $a_1 \leq_1 b_1$ et $a_2 \leq_2 b_2$.

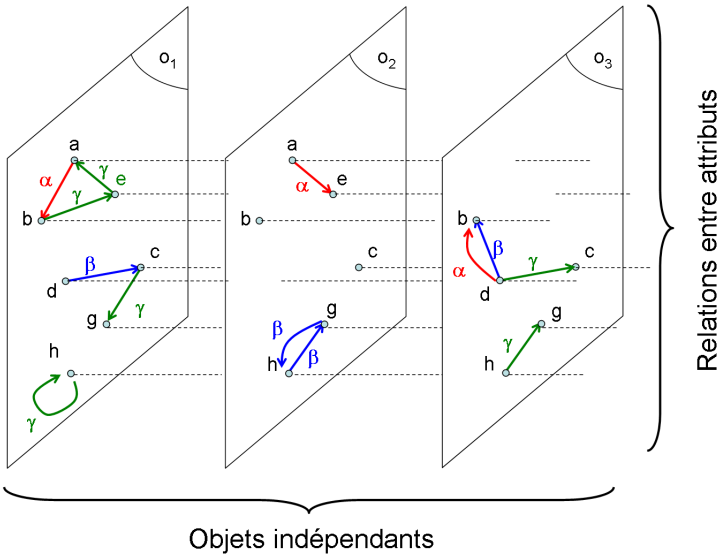


FIG. 2.7 – Un contexte de relations binaires entre attributs

réseaux informatiques, comme Internet ou le World Wide Web. Dans la mesure où les sommets sont singuliers, les arcs et les hyper-arêtes reliant ces sommets le sont aussi. Il suffit donc d'introduire autant d'attributs supplémentaires que nécessaires pour représenter ces arcs ou hyper-arêtes. On se trouve ainsi ramené à un contexte objets × attributs classique. C'est ainsi que Boley *et al.* (2007) aborde la recherche des trajectoires (définies ici comme des ensembles de segments connexes) empruntées fréquemment par des individus entre un ensemble de lieux géographiques. Il résout le problème en représentant chaque segment par un attribut puis en recherchant les motifs d'attributs fréquents soumis à la contrainte particulière de connexité. De même Yan *et al.* (2005b) propose de chercher les sous-graphes d'un graphe relationnel qui sont à la fois fréquents et de connectivité supérieure à un certain seuil. Dans les deux cas les arêtes des graphes sont modélisées par des attributs.

La prise en compte des relations entre objets et l'analyse de concepts relationnels

Les objets d'un contexte peuvent également être liés par des relations indépendamment du fait que les attributs soient eux-mêmes mis ou non en relation. La figure 2.8 illustre un tel contexte relationnel dans le cas particulier de relations binaires. Ainsi l'objet o_6 est décrit par l'attribut c et par les relations qu'il a avec o_4 selon α (i.e. o_4 α o_6) et avec o_1 selon β (i.e. o_6 β o_1). La donnée d'un ensemble de molécules ou de réactions chimiques correspond davantage à cette représentation qu'à celle des graphes relationnels introduits à la section 2.2.1 : les atomes sont alors vus comme des objets dont les attributs correspondent à leurs caractéristiques (élément chimique, charge...) et chaque type de liaisons correspond à une relation binaire symétrique liant des paires d'atomes.

Cependant le traitement des relations entre objets est comme on va le voir très différent et plus compliqué que le traitement des relations entre attributs. Contrairement à ce que peut laisser penser la dualité objets-attributs dans l'ACF, les objets et les attributs ne jouent pas le même rôle dans le problème de l'extraction de connaissances. Cette dissymétrie vient du fait que les objets ne sont que des vecteurs anonymes de leur description, c'est-à-dire leurs attributs. Les objets ne sont donc pas singuliers comme peuvent l'être les attributs et

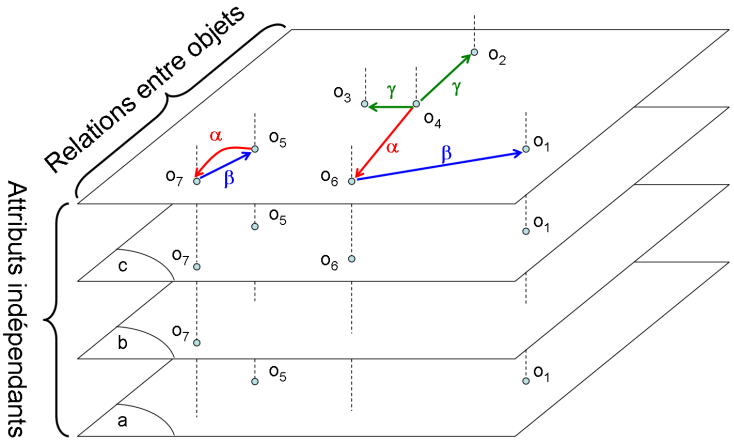


FIG. 2.8 – Un contexte de relations entre objets

à description égale, deux objets sont interchangeables. La question posée n'est alors pas de savoir si tels ou tels objets sont reliés par telles ou telles relations, mais plutôt de savoir si par le truchement de ces objets, telles ou telles combinaisons d'attributs sont reliées par telles ou telles relations. Ainsi sur l'exemple de la figure 2.8, peu importe de savoir que les objets o_6 et o_7 sont reliés respectivement aux objets o_1 et o_5 selon la relation β. Cette observation permet toutefois d'en déduire un motif qui peut s'avérer révélateur : dans les deux cas un objet (o_6 ou o_7) présentant l'attribut c est relié selon la relation β à un objet (o_1 ou o_5) présentant les attributs a et c. Ce motif est maximal (i.e. les descriptions de o_1 et o_5 ne partagent pas d'autres éléments) et il n'existe pas d'autres objets que o_1 et o_5 présentant ce motif. Ce motif associé à l'extension $\{o_1, o_5\}$ peut donc être vu comme un concept formel.

Huchard *et al.* (2007) ou et Ferré *et al.* (2005) proposent d'étendre l'ACF afin d'y intégrer les relations binaires entre objets. En particulier Huchard *et al.* (2007) proposent l'*Analyse de Concepts Relationnels* (ou ACR), formalisme théorique dans lequel l'intension d'un concept formel n'est plus seulement caractérisé par les attributs communs aux objets de l'extension mais aussi par les relations qu'ils entretiennent avec les objets d'autres concepts. Leur méthode est itérative : les concepts formels du contexte sont d'abord déterminés sans tenir compte des relations puis le contexte est enrichi par l'ajout d'autant d'attributs qu'il y a de couples (r, C) relation - concept. Un objet est mis en correspondance avec l'attribut (r, C) s'il est en relation selon r avec un objet faisant partie de l'extension de C. Le processus est ensuite réitéré jusqu'à obtenir le niveau de granularité voulu. L'analyse des relations renvoie à certains langages de représentation des connaissances. Ainsi l'ACR est étroitement liée aux logiques de descriptions (Baader *et al.*, 2002; Napoli, 1997) dans lesquelles l'opérateur $\exists r.C$ de restriction existentielle typée est satisfait par un objet si cet objet est lié selon la relation r à au moins un objet appartenant au concept C. Les concepts formels extraits par Huchard *et al.* (2007) peuvent donc se reformuler en concepts des logiques de descriptions (Voir Hacene *et al.*, 2007) : dans l'exemple précédent, l'intension du concept peut ainsi s'exprimer par l'expression $\mathcal{C}_c \sqcap \exists\beta.(\mathcal{C}_a \sqcap \mathcal{C}_c)$ où le concept $\mathcal{C}_x$ désigne l'ensemble des objets présentant l'attribut x. La perspective envisageable est la déduction d'une base de concepts descriptifs des données à partir d'un contexte relationnel, c'est-à-dire dans le vocabulaire de la logique de description, d'extraire ou de compléter une T-box à partir de la donnée d'une A-box.

Le processus décrit par Huchard *et al.* (2007) ne semble toutefois pas le plus adapté pour

fouiller les motifs structuraux contenus dans les BdR, et ce en raison de plusieurs obstacles sérieux. Le premier problème est de pouvoir gérer l'explosion combinatoire du nombre d'attributs, ce qui oblige rapidement à choisir quels sont les concepts que l'on tient à garder à chaque itération. Par ailleurs à chaque itération, le contexte courant décrit chaque objet par ses attributs ainsi que par ses relations avec les concepts extraits du contexte de l'itération précédente. De ce fait la méthode ne caractérise pas les relations d'objets à objets mais seulement d'objets à classes d'objets, ce qui limite la nature des concepts que l'ACR peut exprimer. Prenons l'exemple représenté sur la figure 2.9(a) de trois objets (cercles) sans attributs et donc indiscernables, qui sont en relation les uns avec les autres selon une relation binaire notée α unique et symétrique (flèches). Ces trois objets forment dans le treillis initial

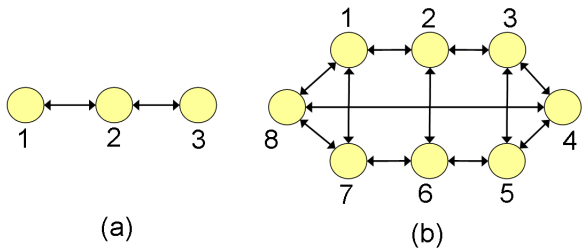


FIG. 2.9 – La limitation des relations objets à concepts

(i.e. sans tenir compte des relations) un concept unique $\top = (\emptyset/123)$ de sorte que chaque objet est décrit dans le contexte enrichi par deux attributs $\top$ et $(\alpha, \top)$ qui se confondent. Les itérations de l'ACR ne permettent donc pas de distinguer les deux objets o_1 et o_3 de l'objet o_2 qui diffère pourtant des précédents par le nombre de ses relations. On peut certes vouloir enrichir le contexte pour tenir compte de la multiplicité des relations (i.e. en ajoutant des attributs du type (α, C, n) pour n variant de 1 à la multiplicité de la relation) mais cela ne résout pas des cas plus complexes où les objets ne sont discernables que par les cycles dont ils font partie. Ainsi sur la figure 2.9(b) les objets sont indiscernables du point de vue des attributs ou de la multiplicité (puisqu'ils sont tous en relation avec trois autres objets indiscernables) mais se distinguent pourtant puisque les objets o_2 et o_6 n'appartiennent à aucun cycle de longueur 3 contrairement aux autres objets. En résumé, l'ACR est utile pour enrichir les concepts de l'ACF en tenant compte de l'existence de relations entre ces objets. Mais l'ACR ne semble pas le modèle le plus adapté pour extraire sous forme de concepts les motifs structuraux présents dans les BdR : d'une part la construction itérative et assez lourde de treillis de concepts est incompatible avec le traitement de quelques milliers de molécules, c'est à dire de plusieurs centaines de milliers d'atomes et de liaisons. D'autre part les intensions des concepts de l'ACR ne semblent pas pouvoir facilement exprimer des motifs tels que des graphes complexes (en particulier les graphes cycliques).

L'ACR est présentée ici car elle aborde l'extraction de connaissances à partir de relations à travers un modèle théorique rigoureux dans le prolongement direct de l'ACF. L'ACR n'est toutefois pas le seul paradigme qui s'intéresse à la fouille de données relationnelles au sens large. La fouille de données relationnelles est encore appelée *fouille de liens* – Link Mining – par Getoor et Diehl (2005) et recoupe l'ensemble des méthodes qui de près ou de loin s'intéressent à l'information topologique que renferment les relations entre objets. Parmi ces méthodes on compte l'analyse statistique de liens, en particulier la fouille du web – Web Mining – et les algorithmes de classement de sites comme `PageRank` ou `HITS`, l'apprentissage

relationnel statistique (Getoor et Taskar, 2007) – Statistical Relational Learning – qui proposent entre autre d'adapter les réseaux bayésiens et les réseaux de Markov au cas de données relationnelles, les méthodes d'apprentissage numérique à base de noyaux de graphes (Gärtner, 2003; Horváth *et al.*, 2004; Kashima et Tsuboi, 2004), la programmation logique inductive et la fouille multi-relationnelle (cf section 2.2.2) et enfin la fouille de graphes (cf section 2.3). Les applications de la fouille de liens se répartissent selon Getoor et Diehl (2005) en quelques grandes familles de problèmes que sont le classement ou la classification de sommets ou de liens dans un graphe, le clustering d'objets en communauté, la fusion d'identité d'objets, la découverte de motifs dans les graphes, la classification de graphes et les modèles génératifs de graphes aléatoires. Selon cette catégorisation, les travaux de ce mémoire se rapportent à l'extraction de motifs dans les graphes et à la classification de sommets ou d'arêtes au chapitre 7. D'autres formalismes permettent de traiter ces problèmes. Ainsi et dans la mesure où le pouvoir expressif que peut en pratique fournir les concepts de l'ACR est plutôt en deçà des exigences de la fouille des BdR, la section suivante s'intéresse à la programmation logique inductive, qui au moins en théorie, peut induire à partir de données des motifs structuraux aussi sinon plus complexes que des graphes.

2.2.2 La programmation logique inductive

La programmation logique inductive (ou PLI) – *Inductive Logic Programming* ou ILP en anglais – est née aux débuts des années 1990 (Muggleton, 1990, 1991) dans le prolongement de l'inférence automatique de programmes à partir d'exemples d'entrée-sortie (Shapiro, 1983). L'idée initiale de la PLI était d'induire automatiquement un programme logique (Kowalski, 1974) tel que ceux exprimables dans le langage `Prolog` (Colmerauer et Roussel, 1993), en généralisant un ensemble d'exemples spécifiés par des clauses de prédicats du premier ordre[15]. La PLI est depuis devenue une des principales branches de l'apprentissage symbolique. Contrairement aux méthodes d'apprentissage qui induisent des ensembles de règles de classification à partir d'exemples décrits par des attributs (interprétables comme des conjonctions d'atomes exprimées dans la logique des propositions), la PLI se place au niveau beaucoup plus expressif de la logique des prédicats du premier ordre. Cette plus grande expressivité se fait toutefois au détriment de la « tractabilité » des problèmes abordés, du fait d'un espace d'hypothèses extrêmement vaste. Heureusement cet espace de recherche peut être contraint par la donnée d'un ensemble de clauses que le processus de recherche doit respecter. Cette intégration transparente d'un modèle de représentation des connaissances a priori dans le processus d'apprentissage constitue un des atouts majeurs de la PLI.

Formellement, étant donnés un prédicat cible C, un ensemble $E = E^+ \cup E^-$ d'exemples positifs et négatifs qui caractérisent C (cf précisions plus loin) et une théorie causale B spécifiant la connaissance a priori – ou *Background Knowledge* – du problème, la PLI cherche à induire une théorie causale H la plus générale possible (i.e. la plus concise) qui soit consistante (i.e. non contradictoire) avec B et telle que la théorie $H \cup B$ « couvre » chaque exemple e de E^+ et aucun de E^-. Cette relation de couverture varie selon les différentes méthodes de résolution proposées. Ainsi Raedt (1997) distingue trois approches selon que la relation de couverture se réfère à l'inférence inductive – ou inverse entailment (Muggleton, 1995) – (i.e.

[15]On rappelle brièvement le vocabulaire de la logique des prédicats : un *atome* est un prédicat n-aire entre *termes* définis récursivement à partir de termes composés et de termes terminaux qui sont soit des constantes, soit des variables. Une *clause* est une formule disjonctive $l_1 \vee l_2 \cdots \vee l_n$ de littéraux l_i équivalente à $l_1 \vdash \neg l_2 \wedge \cdots \wedge \neg l_n$, un littéral étant un atome a ou sa négation $\neg a$. Une *théorie clausale* est une conjonction de clauses.

tout exemple e, qui est une clause, est couvert par $H \wedge B$ s'il en est une conséquence logique : $H \wedge B \models e$), sur la satisfaisabilité (i.e. un exemple qui est ici une clause, est couvert si la théorie $H \wedge B \wedge e$ est satisfaisable, c'est-à-dire admet au moins un modèle), ou enfin sur les interprétations (Raedt et Dzeroski, 1994) (i.e. un exemple e, qui est ici une interprétation de Herbrand, est couvert si e est un modèle pour – i.e. « ne contredit pas »– $H \wedge B$). Les figures 2.10 et 2.11 illustrent les approches fondées respectivement sur l'inférence inductive et les interprétations.

$$
\begin{array}{rll}
 & \text{B :} & \text{personnage(X)} \vdash \text{philosophe(X)} \\
 & E^+ : & \text{mortel(aristote)} \vdash \text{philosophe(aristote)} \\
 & & \text{mortel(lucy)} \vdash \text{personnage(lucy)} \\
 & E^- : & \neg\ \text{mortel(zeus)} \vdash \text{dieu(zeus), personnage(zeus)} \\
\rightarrow & \text{H :} & \text{mortel(X)} \vdash \text{personnage(X)}, \neg\ \text{dieu(X)}
\end{array}
$$

FIG. 2.10 – Exemple du formalisme de la PLI fondé sur l'inférence inductive

$$
\begin{array}{rll}
 & \text{B :} & \text{personnage(X)} \vdash \text{philosophe(X)} \\
 & \text{E :} & I_1 = \{\ \text{mortel(aristote), philosophe(aristote)}\ \} \\
 & & I_2 = \{\ \text{mortel(lucy), personnage(lucy)}\ \} \\
 & & I_3 = \{\ \text{dieu(zeus), personnage(zeus)}\ \} \\
\rightarrow & \text{H :} & \text{mortel(X)} \vdash \text{personnage(X)}, \neg\ \text{dieu(X)}
\end{array}
$$

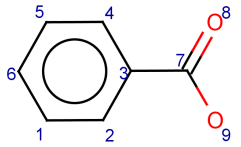
FIG. 2.11 – Exemple du formalisme de la PLI fondé sur des interprétations

Selon Blockeel *et al.* (1999), la seconde approche fondée sur les interprétations est plus naturelle et plus efficace que l'approche par résolution inverse. Par ailleurs les interprétations peuvent être vues comme des exemples indépendants formés d'une conjonction d'atomes, comme peuvent l'être les transactions décrites par des attributs dans le cas de la recherche de motifs d'attributs fréquents. Le modèle par interprétations permet ainsi d'associer aux clauses les notions de fréquence et de confiance, semblables en cela aux règles d'association. En associant à une hypothèse un degré de vraisemblance, ce modèle pallie un inconvénient majeur des premières méthodes de PLI qui ne supportaient pas la moindre contradiction dans les exemples. La différence essentielle vis-à-vis de la recherche de motifs d'attributs fréquents est que les interprétations se répartissent non pas sur une, mais plusieurs tables selon le modèle des bases de données relationnelles déjà évoqué à la section 2.2.1 : chaque prédicat n-aire est représenté par une table à n colonnes dont chaque ligne représente une instance du prédicat figurant dans une des différentes interprétations. L'analogie avec la recherche de motifs d'attributs fréquents permet de tirer parti de l'efficacité des algorithmes de recherche des motifs fréquents au cas multi-relationnel.

Ainsi l'algorithme `Warmr` (Dehaspe et Raedt, 1997) généralise l'algorithme `Apriori` à la logique des prédicats du premier ordre en remplaçant les attributs par les atomes fréquents dans les interprétations. Par analogie, les atomes jouent le rôle d'attributs, les requêtes (conjonctions d'atomes impliquant un atome spécial appelé clef pour identifier un objet) sont les motifs et le nombre de clefs différentes satisfaites par la requête joue le rôle de support. L'extension d'une requête par un nouvel atome est toutefois plus complexe que l'ajout d'un attribut puisqu'il doit prendre en compte les différents jeux de paramètres possibles pour ce nouvel atome (par une variable libre ou valuée, ou encore par une constante). `Warmr` vérifie

ensuite que les nouvelles requêtes ainsi générées ne sont pas généralisées par des requêtes non fréquentes plus courtes ou équivalentes à des requêtes fréquentes plus courtes. Le test de spécialisation se fonde sur la relation de θ-subsomption (Raedt *et al.*, 1997) qui s'avère un problème NP-complet. Ce test étant réalisé très fréquemment, `Warmr` est beaucoup plus lent que `Apriori`.

Les algorithmes `Progol` (Muggleton *et al.*, 1998)) fondé sur l'inférence inductive et `Warmr` (Dehaspe *et al.*, 1998) fondé sur les interprétations, ont été appliqués au problème de la classification de graphes moléculaires. La figure 2.12 explicite la façon dont les graphes moléculaires sont modélisés, ici sous la forme d'une interprétation telle que utilisée par `Warmr`. Quelque soit la méthode considérée, les expérimentations semblent limitées par le nombre d'exemples (quelques centaines) que peuvent traiter les algorithmes et surtout, par la taille des requêtes retournées qui représentent des motifs chimiques constitués tout au plus de quelques liaisons (i.e. moins de 5).



atom(1, c). ... atom(7, c).
atom(8, o). atom(9, o).
bond(1, 2, arom). ... bond(6, 1, arom).
bond(3, 7, 1).
bond(7, 8, 2).
bond(7, 9, 1).

FIG. 2.12 – Spécification d'un graphe moléculaire en PLI. Le prédicat $atom(n, e)$ met en relation l'atome n° n à l'élément chimique e et le prédicat $bond(n_1, n_2, t)$ déclare une liaison de type t entre les atomes de numéros n_1 et n_2.

En raison de ces limitations, les développements en PLI ont eu pour objectif de rendre les algorithmes moins consommateurs en temps de calcul quite à faire des concessions en terme d'expressivité. Ainsi Nijssen et Kok (2001) proposent l'algorithme `Farmer` dérivé de `Warmr` qui supprime le test coûteux de θ-subsomption grâce à une représentation canonique des requêtes. Ce faisant, `Farmer` met en évidence un certain nombre de principes (représentation canonique des motifs, parcours en profondeur) que l'on retrouve en fouille de graphe. Ce rapprochement de la PLI avec des langages de motifs de complexité intermédiaire ne cesse de se confirmer. Ainsi certains membres issus de la communauté de la PLI, comme Siegfried Nijssen, ont apporté des contributions majeures dans le domaine de la recherche de motifs d'arbres (Nijssen et Kok, 2003) ou de graphes fréquents (Nijssen et Kok, 2004). À l'inverse les résultats obtenus depuis en fouille de graphes, très concluants du point de vue des performances et du passage à l'échelle, ont incité les développements récents (Raedt, 2005; Raedt *et al.*, 2008a; Raedt, 2008) de la PLI à intégrer plus efficacement dans leur modèle la notion de relation binaire.

2.3 L'extraction de connaissances à partir de graphes

La PLI cherche à résoudre un problème très complexe nécessitant une recherche dans un espace d'hypothèses extrêmement vaste. Le recours à l'inférence logique de la PLI n'est toutefois pas nécessaire pour prendre en compte les phénomènes de cycles ou de multiplicité des relations qui échappent par ailleurs aux méthodes de fouille de relations du niveau de l'ACR (cf. sect. 2.2). Ces phénomènes peuvent en effet être étudiés en extrayant directement les mo-

tifs de graphes présents dans les données. Ces approches intermédiaires dont l'expressivité (i.e. la richesse des motifs exprimables) se situe entre celle de la fouille de relations et celle de la PLI, sont regroupées sous le terme générique *extraction de connaissances à partir de graphes*. Ces approches comptent parmi elles les méthodes dites de fouille de graphes, dont l'objectif est la conception d'algorithmes efficaces de fouille de données représentées par des graphes, et qui sont déjà exploitées en pratique par des applications à l'aide des moyens de calcul actuellement disponibles. Dans la mesure où les méthodes présentées dans ce mémoire se rattachent à cette catégorie, la section suivante leur est entièrement consacrée. Mais l'extraction de connaissances à partir de graphes ne se limite pas à la fouille de graphes et au traitement des bases de données. Elle inclut également d'autres méthodes qui se rattachent plus naturellement aux travaux antérieurs en IA et plus particulièrement aux travaux en représentation des connaissances. Ces méthodes abordent le problème de l'extraction de connaissances à partir de graphes sous un angle plus formel que ne le font les méthodes de fouille de graphes. Cet angle formel est repris dans la présente section pour aborder succinctement les fondations théoriques de la fouille de graphes. Ainsi la section 2.3.1 introduit de manière très générale les graphes comme support d'information et rappelle certaines propriétés fondamentales qui font de l'extraction de connaissances à partir de graphes un problème intrinsèquement plus difficile – au sens de la complexité – que la fouille de motifs d'attributs. La section 2.3.2 présente ensuite une extension théorique de l'ACF (cf. sect. 2.1.3), appelée structures de motifs, qui illustre bien les difficultés qui existent à manipuler les graphes à grande échelle.

2.3.1 Les graphes comme support d'information

Un moyen naturel de représentation des relations

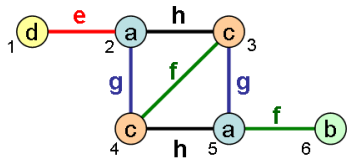
Les origines de la notion de graphe en tant qu'objet mathématique remonte au problème des ponts de Königsberg énoncé et résolu par Euler en 1735. La théorie des graphes (Berge, 1962; Bollobás, 1998; Tutte, 2001) a depuis connu un développement rapide dont les résultats ont pu servir à résoudre efficacement de nombreux problèmes notamment informatiques. Les graphes interviennent ainsi dans des champs d'application tels que la chimie avec les graphes moléculaires, les problèmes de logistique et d'optimisation en général, les réseaux biologiques, les réseaux sociaux ou de communications. Dans le domaine informatique, les graphes servent souvent de support pour structurer l'information, les sommets et les arêtes de ces graphes ayant alors la particularité d'être associés à des descripteurs. Ce besoin d'enrichir un graphe par des informations supplémentaires correspond à la notion de *graphe étiqueté*, centrale à ce mémoire :

Définition 2.3.1. Un *graphe étiqueté simple, non orienté* $G = (V_G, E_G, \mathcal{L}, l_G^v, l_G^e)$ est défini par :

- Un ensemble $V(G) = V_G$ de *sommets*.
- Un ensemble $E(G) = E_G \subseteq \mathcal{P}_2(V_G)$ d'*arêtes* constituées de paires de sommets.
- Un ensemble $\mathcal{L}$ de *types* de sommets et d'arêtes, appelés aussi *étiquettes*.
- Un *étiquetage des sommets* $l_G^v : V_G \to \mathcal{L}$.
- Un *étiquetage des arêtes* $l_G^e : E_G \to \mathcal{L}$.

Un exemple d'un tel graphe étiqueté est donné sur la figure 2.13. Chaque sommet y est représenté par un disque dont la couleur est caractéristique de son étiquette, également précisée au centre du disque par une lettre de l'alphabet. De plus, chaque sommet est identifié de manière unique par un indice jouxtant son disque. Enfin chaque arête est représentée par

un trait dont la couleur est elle aussi caractéristique de son étiquette explicitée par le symbole qui le surmonte. Dans la suite du mémoire, le terme graphe fera référence sauf exception à un graphe étiqueté, simple (i.e. sans boucles ni arêtes multiples) et non orienté dont les notations associées seront celles de la définition précédente. La suite de ce mémoire fait parfois aussi mention de quelques termes parmi les plus usités de la théorie des graphes comme les graphes orientés, non orientés, les relations d'adjacence, d'incidence et de connexité, le degré d'un sommet, les cycles et les chemins, les arbres et les forêts, les composantes connexes, les sous-graphes et super-graphes. Leurs définitions formelles sont disponibles dans tout ouvrage sur le sujet, comme par exemple Bollobás (1998).



(a) Représentation

$$V(G) = \{1; \ldots; 6\}$$
$$E(G) = \{\{1;2\}; \{2;3\}; \ldots; \{5;6\}\}$$
$$\mathcal{L} = \{a; \ldots; h\}$$
$$l^v_G(1) = d, \ldots, l^v_G(6) = b$$
$$l^e_G(\{1;2\}) = e, \ldots, l^e_G(\{5;6\}) = f$$

(b) Spécification

Fig. 2.13 – Graphe étiqueté simple non orienté

Parmi les nombreux modèles qui utilisent les graphes étiquetés pour représenter des données voire des connaissances, on peut en particulier citer les exemples suivants :

- Les langages de modélisation objets comme UML définissent des classes d'objets et des objets instances de ces classes. Les objets renferment des attributs de type élémentaire propres à leur classe et des références pointant sur d'autres objets. Les diagrammes d'objets ou de classes UML forment donc des graphes orientés étiquetés.
- Les bases de données relationnelles définissent des relations n-aires entre objets par la définition de n-tuples au sein de tables relationnelles. Les lignes de chaque table représentent les relations existantes ou tuples et les colonnes représentent des variables de type élémentaire (entier, réel, chaîne de caractère ...). Certaines de ces variables servent de clés pour identifier de manière unique un tuple et mettre en rapport différentes relations. Une fois mise sous sa forme normale (Codd, 1974), une base de données relationnelles peut se traduire en un graphe orienté de tuples : les clefs identifient les sommets, les valeurs des variables qui ne sont pas des clefs jouent le rôle d'étiquettes et l'occurrence d'une clef dans un tuple induit un arc de ce tuple vers le tuple désigné par la clef.
- Les langages de représentation des connaissances comme les logiques des descriptions (Baader *et al.*, 2002; Napoli, 1997), les réseaux sémantiques (Simmons, 1966) puis les graphes conceptuels (Sowa, 1976, 1999) et les langages de requêtes du Web sémantique comme SPARQL/RDF (Prud'hommeaux et Seaborne, 2008) ont tous pour interprétations des graphes d'objets liés par des relations binaires.

Les graphes constituent donc un formalisme très répandu pour la représentation des données informatiques. Certains travaux proposent même d'utiliser les graphes non seulement comme moyen de représentation de l'information mais comme moyen de calcul et de transformation des données. Ainsi les *grammaires de graphes* (Ehrig *et al.*, 2006) fondées sur la théorie des catégories permettent de spécifier rigoureusement des opérations de transformation de graphes par d'autres graphes et ainsi d'écrire des algorithmes sous forme de graphes.

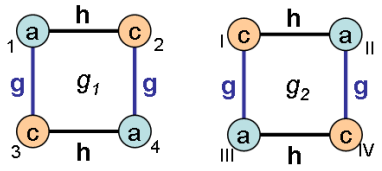
Comparaison avec les motifs d'attributs

Si les graphes constituent un moyen puissant pour représenter données et informations, les graphes sont aussi beaucoup plus difficiles à manipuler que les descriptions à base d'attributs pour trois raisons essentielles qui sont développées ci-après dans un ordre de complexité croissante :

La relation d'isomorphisme. La première difficulté posée par les graphes est de pouvoir comparer les structure respectives de deux graphes définies par les relations d'incidence ainsi que les types qui étiquettent sommets et arêtes : c'est ce que l'on fait implicitement lorsque par exemple on confond deux molécules d'eau en superposant leurs graphes moléculaires $H - O - H$ alors que dans l'absolu ces deux graphes sont différents puisque composés de sommets (i.e. d'atomes) différents. La comparaison de structures topologiques renvoie à la notion mathématique d'isomorphisme : deux graphes g_1 et g_2 sont *isomorphes* (i.e. de même structure topologique) s'il existe un *isomorphisme* entre g_1 et g_2 :

Définition 2.3.2. Un isomorphisme de g_1 vers g_2 est une bijection μ de $V(g_1)$ vers $V(g_2)$ qui préserve simultanément :
- La relation d'incidence : $\{v_1; v_2\} \in E(g_1) \Leftrightarrow \{\mu(v_1); \mu(v_2)\} \in E(g_2)$.
- L'étiquetage des sommets : $v \in V(g_1) \Rightarrow l^v_{g_2}(\mu(v)) = l^v_{g_1}(v)$.
- L'étiquetage des arêtes : $\{v_1; v_2\} \in E(g_1) \Rightarrow l^e_{g_2}(\{\mu(v_1); \mu(v_2)\}) = l^e_{g_1}(\{v_1; v_2\})$.

Un *automorphisme* d'un graphe est un isomorphisme de ce graphe vers lui-même et traduit, lorsqu'il est différent de l'identité, une symétrie interne dans la structure topologique du graphe. Les automorphismes forment un sous-groupe du groupe symétrique des permutations de sommets, qui joue un rôle important dans le calcul du représentant canonique d'un graphe (McKay, 1981). La figure 2.14 représente un exemple simple d'isomorphisme et d'automorphisme. La relation d'isomorphisme entre graphes notée $\simeq_G$ définit une relation d'équivalence



(a) Graphes isomorphes

$V(g_1)$	$V(g_2)$
1	II
2	I
3	IV
4	III

(b) Isomorphisme

$V(g_1)$	$V(g_1)$
1	4
2	2
4	1
3	3

(c) Automorphisme

FIG. 2.14 – Graphes isomorphes g_1 et g_2, isomorphisme de g_1 vers g_2 et automorphisme de g_1

sur les graphes. Deux graphes isomorphes ne se distinguent donc que par le choix somme toute arbitraire de leur ensemble de sommets. En supposant que les n sommets d'un graphe soient codés par un indice allant de 1 à n, deux graphes isomorphes se déduisent l'un de l'autre en permutant les indices de numérotation de leurs sommets. L'algorithme de recherche exhaustive permettant de détecter si deux graphes sont isomorphes nécessite de tester toutes les permutations de sommets et a donc une complexité en $O(n!)$. La propagation des contraintes liées à la relation d'incidence, aux étiquettes de sommets et d'arêtes permet en pratique de réduire sensiblement le nombre de permutations à tester. Si de nombreux algorithmes (Corneil et Gotlieb, 1970; Schmidt et Druffel, 1976; Babai *et al.*, 1982) ont été proposés pour

détecter l'isomorphisme entre deux graphes et si les meilleurs d'entre eux (Babai *et al.*, 1982) ont une complexité subexponentielle, la détection d'isomorphisme demeure un problème NP dont on ignore la complexité exacte et rien n'empêche pour l'instant que cette complexité se révèle un jour polynomiale. Cet espoir est conforté par le fait que de nombreuses familles de graphes bénéficient déjà d'algorithmes de complexité polynomiale, comme la famille des arbres (Kelly, 1957), des graphes planaires (Hopcroft et Wong, 1974) ou des graphes de degré borné (Luks, 1982) dont font notamment partie les graphes moléculaires.

En comparaison, le problème d'isomorphisme entre motifs d'attributs est trivial au point qu'il ne semble même pas se poser. Il a toutefois le mérite d'introduire très simplement la notion de représentant canonique, notion centrale aux algorithmes de fouille de graphes. En effet, à y regarder de plus près, un motif de longueur l dont les attributs sont choisis dans un ensemble de n éléments ne se représente généralement pas par un tableau booléen à n éléments mais plutôt par une séquence de ses l attributs qui a l'avantage d'être une représentation plus compacte lorsque l est petit devant n. Un motif $\{a, b, c\}$ admet alors autant de codages possibles que de permutations de ses attributs (a, b, c), (a, c, b), ..., (c, b, a). Ces codages apparaissent comme autant de variations syntaxiques d'un et un seul motif. Cette multitude de syntaxes équivalentes est un phénomène néfaste compliquant les tests d'égalité et d'inclusion entre motifs. De plus le stockage d'un ensemble de motifs dans un dictionnaire est rendu impossible en l'état car il ne permet pas de retrouver un motif à partir d'une de ses variations syntaxiques. Le cas des motifs d'attributs ne pose en pratique aucun problème : il suffit de choisir pour tout motif le codage constitué de la séquence des attributs du motif triés par ordre selon un ordre arbitraire, comme par exemple l'ordre alphabétique : $\{a, b, c\}$ se représente ainsi par (a, b, c). Les tests d'égalité et d'inclusion entre motifs ainsi représentés deviennent alors de complexité linéaire au lieu d'être quadratique (pour l'inclusion) ou du moins log-linéaire (si on effectue un tri préalable des attributs).

L'exemple précédent illustre la nécessité de disposer d'un représentant canonique parmi l'ensemble de motifs isomorphes. Plus exactement, étant donné un ensemble de codages $\mathcal{C}$ munis d'une relation d'équivalence syntaxique $\simeq$, une *représentation canonique* définie sur $\mathcal{C}$ est une application $\lambda : \mathcal{C} \to \mathcal{C}$ vérifiant deux conditions :

- Deux éléments isomorphes ont même représentant (i.e. $\forall c_1 \forall c_2, c_1 \simeq c_2 \Rightarrow \lambda(c_1) = \lambda(c_2)$).
- Un élément et son représentant sont isomorphes (i.e. $\forall c, \lambda(c) \simeq c$).

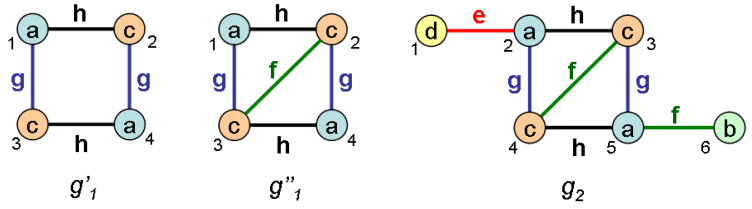
Le calcul d'une représentation canonique permet de résoudre la détection d'isomorphisme dans la mesure où deux motifs sont isomorphes si et seulement si leurs représentants canoniques respectifs sont identiques. Dans le cas des graphes, le problème revient à trouver une numérotation canonique des sommets. `Nauty` (McKay, 1981) est connu comme l'algorithme le plus efficace pour calculer une numérotation canonique des sommets d'un graphe étiqueté. Il repose sur une propagation de contraintes au sein du groupe des automorphismes du graphe.

Une fois un représentant canonique sélectionné, ce dernier peut se coder en une séquence de symboles appelée *codage canonique*. Dans le cas des graphes, il existe deux principales façon de coder un graphe, soit par la matrice d'adjacence si le graphe est simple, soit plus couramment par les listes des arêtes ou des arcs incidents à chaque sommet. Les codages canoniques sont particulièrement utiles pour confectionner des dictionnaires de graphes à un isomorphisme près dans la mesure où les codages canoniques des graphes sont ni plus ni moins des séquences de symboles que l'on peut stocker efficacement dans un arbre préfixé (i.e. trie).

La relation d'ordre entre graphes. De nombreuses méthodes d'apprentissage symbolique se fondent sur une relation de subsomption entre concepts, reliant les descriptions de concepts les plus générales au plus spécifiques[16]. Dans le cas des graphes, un graphe g_1 est plus général (i.e. moins spécifique) qu'un graphe g_2 si la structure topologique de g_1 est incluse dans la structure topologique de g_2 indépendamment là encore du choix des sommets de g_1 et de g_2. Un distinguo est possible selon que g_1 soit exigé isomorphe à un *sous-graphe* g_3 *induit* ou seulement *partiel* de g_2 :

- Le graphe g_3 est un *sous-graphe partiel* de g_2 si ses sommets et ses arêtes sont aussi des sommets et des arêtes de g_2 (i.e. $V(g_3) \subseteq V(g_2)$ et $E(g_3) \subseteq E(g_2)$) et si ses fonctions d'étiquetages $l^v_{g_3}$ et $l^e_{g_3}$ coïncident avec les restrictions de $l^v_{g_2}$ et $l^e_{g_2}$ sur les sommets et arêtes de g_3.
- Le sous-graphe partiel g_3 devient un *sous-graphe induit* de g_2 lorsque deux sommets de g_3 sont adjacents dans g_3 si et seulement s'ils le sont aussi dans g_2 (i.e. $\forall\{v_1; v_2\} \subseteq V(g_3), \{v_1; v_2\} \in E(g_2) \Leftrightarrow \{v_1; v_2\} \in E(g_3)$).

La figure 2.15 représente un exemple de sous-graphe partiel (i.e. g'_1) et induit (i.e. g''_1 qui doit inclure l'arête $\{2, 3\}$ en plus de celles de g'_1 pour être induit dans g_2). Quelle que soit la relation d'inclusion retenue, l'isomorphisme de g_1 dans g_3 induit un *morphisme injectif* $\mu : V(g_1) \rightarrow V(g_2)$ qui préserve la relation d'incidence et l'étiquetage des sommets et des arêtes de g_1 vers g_2. Pour être exact, la relation de sous-graphe isomorphe (partiel ou induit)



(a)

$V(g_1)$	$V(g_2)$
1	2
2	3
3	4
4	5

(b)

FIG. 2.15 – Sous-graphes partiel g'_1 et induit g''_1 de g_2 (a). Morphisme injectif de g'_1 vers g_2 (b).

notée $\leq_G$ n'est pas exactement une relation d'ordre mais définit seulement un pré-ordre (relation transitive et réflexive mais pas anti-symétrique) sur l'ensemble des graphes. C'est la relation quotient (par passage à l'ensemble quotient des classes d'équivalence de graphes isomorphes) qui devient une véritable relation d'ordre et qui sert de relation de subsomption dans le cadre de la fouille de graphes. En pratique la relation d'inclusion la plus utilisée pour définir la fréquence d'un graphe est celle plus générale de sous-graphe partiel isomorphe, à l'exception de Inokuchi *et al.* (2000) qui s'intéressent au problème de recherche des sous-graphes induits fréquents.

Si la relation d'isomorphisme entre graphes était déjà beaucoup plus coûteuse que le test d'égalité entre motifs d'attributs (complexité subexponentielle contre linéaire), la différence de complexité s'accroît davantage encore pour le test d'inclusion. Alors que le test d'inclusion était de complexité linéaire dans le cas de motifs d'attributs, la détection de sous-graphe isomorphe est un problème NP-complet (Garey et Johnson, 1979) dont les meilleurs

[16]Dans la suite, le symbole $\sqsubseteq$ fera référence à une relation de subsomption. Selon cette relation, un concept S plus spécifique qu'un concept G se notera $S \sqsubseteq G$.

algorithmes sont de complexité exponentielle dans le pire cas. La encore les algorithmes de référence (Ullmann, 1976; Haralick et Elliott, 1980) utilisent la propagation de contraintes liées aux relations d'incidence et d'étiquetage pour réduire le nombre de morphismes injectifs à tester. La détection de sous-graphe isomorphe est un problème central de la fouille de graphes puisqu'elle permet de définir la fréquence d'un motif de graphes comme le nombre de graphes des données qui contiennent au moins un sous-graphe isomorphe au motif.

L'absence de structure de treillis. Enfin la troisième différence avec les motifs d'attributs tient à l'absence de structure de semi-treillis sur l'ensemble des graphes rendant impossible toute définition propre d'intersection (ou d'union) entre graphes, comme c'est le cas avec les ensembles d'attributs. La notion de *sous-graphe commun maximal* (SGCM) est celle qui se rapporte le plus à la notion d'intersection. Un sous-graphe commun (induit ou partiel) à deux graphes g_1 et g_2 est un graphe isomorphe à la fois à un sous-graphe de g_1 et à un sous-graphe de g_2. Un SGCM de g_1 et g_2 est un sous-graphe g_3 commun à g_1 et g_2 qui est maximal, autrement dit tel qu'il n'existe pas d'autre sous-graphe g_4 commun à g_1 et g_2 tel que g_3 soit un sous-graphe isomorphe à g_4. À cela peuvent s'ajouter d'autres contraintes comme le fait que les SGCM doivent être connexes.

Si la notion de SGCM correspond intuitivement à la notion d'intersection entre graphes, les SGCMs n'induisent pas une structure de semi-treillis car deux graphes admettent en général plus d'un SGCM. La figure 2.16 en est un exemple. On peut vérifier que les deux graphes

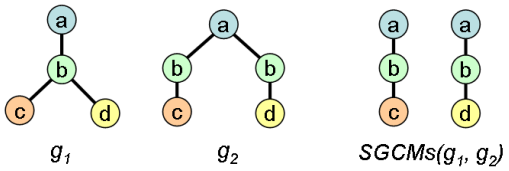


FIG. 2.16 – Les deux sous-graphes communs maximaux connexes des graphes g_1 et g_2.

proposés sont bien des SGCMs de g_1 et de g_2 puisqu'ils sont bien chacun i) isomorphes à des sous-graphes partiels de g_1 et de g_2, ii) tout ajout d'une arête à un sous-graphe de g_1 isomorphe à un des deux SGCMs produit un graphe qui n'est plus isomorphe à un sous-graphe de g_2, prouvant par là que les SGCMs sont bien maximaux. Outre le problème de non-unicité, le calcul des SGCMs est un problème NP-difficile (Garey et Johnson, 1979) qui peut se révéler particulièrement coûteux. Le calcul des SGCMs se résout une fois encore par la propagation de contraintes dans le graphe produit de g_1 et g_2 (Mcgregor, 1982; Régin, 1995). Le calcul des SGCMs est sous-jacent à nombres de méthodes d'apprentissage à partir de graphes que ce soit à travers des méthodes de généralisation à partir d'exemples (Régin *et al.*, 1995; Régin, 1995), de méthodes d'apprentissage numérique fondées sur des distances (Bunke et Shearer, 1998; Bunke *et al.*, 2002) ou sur des noyaux dits d'appariements (Fröhlich *et al.*, 2005).

2.3.2 Les structures de motifs

Liquiere et Sallantin (1998), Kuznetsov (1999) puis Ganter et Kuznetsov (2001) ont cherché à étendre le modèle formel des treillis de Galois et de l'ACF (cf 2.1.3) au cas où les objets ne se décrivent plus sous la forme d'un contexte tabulaire, c'est à dire par un ensemble d'attributs, plus éventuellement par des relations avec d'autres objets comme dans le cas

de l'ACR, mais sous la forme de descriptions plus complexes comme des graphes étiquetés. Ce formalisme est parfaitement adapté à l'application envisagée dans laquelle des réactions chimiques jouant le rôle d'objets, sont décrites par des graphes moléculaires. En particulier, Ganter et Kuznetsov (2001) aboutissent à un modèle rigoureux et général dans lequel une application $\delta : G \rightarrow D$ associe à tout objet o d'un ensemble G une description ou motif $\delta(o)$ exprimé dans un ordre de subsomption $(D, \sqsubseteq_D)$ quelconque. Deux cas peuvent alors se produire.

Soit cet espace D est un semi-treillis disposant d'un opérateur $\sqcup$ de généralisation commune la plus spécifique. Ganter et Kuznetsov (2001) montrent alors qu'on retrouve les résultats de l'ACF à partir de la donnée de $(G, (D, \sqcup), \delta)$ appelée *structure de motifs*[17].

Soit l'ordre $(D, \sqsubseteq_D)$ ne dispose pas de cette structure de semi-treillis, comme c'est le cas des graphes étiquetés connexes définis à un isomorphisme près et ordonnés par la relation $\leq_G =_D \sqsupseteq$ de sous-graphe isomorphe (cf section 2.3.1). Ganter et Kuznetsov (2001) proposent alors une méthode permettant d'induire une structure de semi-treillis à partir $(D, \sqsubseteq_D)$. L'idée consiste à manipuler les idéaux de l'ordre $(D, \sqsubseteq_D)$ des motifs plutôt que les motifs eux-mêmes : un *idéal* est un ensemble I de motifs stable par généralisation (i.e. $d_1 \in I$ et $d_1 \sqsubseteq d_2 \Rightarrow d_2 \in I$). Les idéaux ont cette propriété de rester stables sous l'action de l'intersection et de l'union ensemblistes : si I_1 et I_2 sont deux idéaux, $I_1 \cup I_2$ et $I_1 \cap I_2$ sont eux aussi des idéaux. Le semi-treillis $(\underline{D}, \sqcup = \cap)$ des idéaux de $(D, \sqsubseteq_D)$ qui en résulte, induit une structure de motifs $(G, (\underline{D}, \sqcup), \underline{\delta})$ dans laquelle un objet o est associé non plus à son motif $\delta(o)$ mais à l'idéal $\underline{\delta}(o)$ de ce motif (i.e. l'ensemble des motifs au moins aussi généraux que $\delta(o)$). Cet artifice permet en théorie d'intégrer des motifs complexes tels les graphes étiquetés au sein de l'ACF et de leur appliquer les méthodes d'apprentissage ou d'extraction de connaissances qui y sont disponibles, les idéaux jouant le rôle des intensions de concepts formels.

En pratique chaque idéal peut se représenter par l'ensemble $I = \{d_1; \ldots d_n\}$ des motifs $\sqsubseteq_D$-minimaux (i.e. les plus spécifiques) de cet idéal. Ces éléments définissent une *anti-chaîne*, c'est à dire un ensemble d'éléments qui sont non comparables deux à deux. Dans le cas des graphes, le calcul de l'intersection de deux idéaux $I_1 = \{g_1^1; \ldots g_n^1\}$ et $I_2 = \{g_1^2; \ldots g_m^2\}$ consiste alors à calculer l'union $I_1 \sqcup I_2 = \bigcup SGCMs(g_i^1, g_j^2)$ des SGCMs connexes communs à un graphe de I_1 et un graphe de I_2. L'ensemble obtenu n'étant pas nécessairement une anti-chaîne, il est ensuite nécessaire d'éliminer tous les graphes de $I_1 \sqcup I_2$ qui ne sont pas minimaux, c'est à dire qui sont un sous-graphe isomorphe d'un autre graphe élément de $I_1 \sqcup I_2$. Une solution alternative permettant d'éviter ces calculs consiste à représenter un idéal par l'ensemble de ces éléments, chaque élément étant en bijection avec un attribut (i.e. un indice dans un dictionnaire). Le calcul de l'intersection de deux idéaux revient alors à calculer l'intersection de deux ensembles d'attributs. Cette alternative nécessite cependant de calculer les idéaux associés aux objets du contexte en énumérant pour chaque objet o l'ensemble des sous-graphes inclus dans le graphe $\delta(o)$ décrivant o puis à associer dans un dictionnaire leur représentant canonique à un indice unique. Or le nombre de ces sous-graphes croit extrêmement rapidement avec la taille du graphe $\delta(o)$, rendant cette solution alternative inexploitable en pratique.

Le modèle des idéaux est intéressant d'un point de vue théorique mais nécessite des traitements particulièrement coûteux (nombreux calculs de SGCMs et de détection de sous-graphes isomorphes) qui rendent en pratique le modèle difficilement utilisable en l'état. Ganter et Kuznetsov (2001) le reconnaissent et proposent de projeter les idéaux dans des espaces plus simples afin de les manipuler plus efficacement mais aussi plus approximativement. Pour

[17] *Pattern structure* en anglais.

être compatible avec la structure de motifs, une telle projection p suppose d'être stable par intersection : $p(I_1 \cap I_2) = p(I_1) \cap p(I_2)$. C'est le cas de la projection p_k consistant à représenter un idéal de graphes par le sous-ensemble de ses graphes dont le nombre de sommets est inférieur ou égal à un entier k. L'avantage des projections est de limiter le nombre de motifs contenus dans les idéaux ainsi approximés et ainsi de pouvoir recourir à la représentation explicite des idéaux par un ensemble d'attributs si tant est que la projection en question soit suffisamment réductrice (i.e. k suffisamment petit). Ganter et Kuznetsov (Ganter et Kuznetsov, 2003; Ganter *et al.*, 2004) ont ainsi cherché à prédire la toxicité de molécules à partir d'exemples en calculant les projections p_k des idéaux associés à chacun des exemples pour k variant de 1 à 8. Les auteurs utilisent ensuite le principe de l'espace des versions (Mitchell, 1977, 1997) pour extraire les espaces de version compatibles avec les exemples positifs et négatifs.

Les structures de motifs permettent en théorie d'extraire et d'organiser sous forme de treillis de concepts, les sous-graphes présents dans des données représentées sous forme de graphes. En ce sens, les structures de motifs sont un formalisme valable pour l'extraction de connaissances à partir de BdR. Cependant, il ne semble pas évident que les structures de motifs et les autres travaux similaires soient capables d'apporter des réponses concrètes aux applications de fouille de graphes. Pour être utilisables en pratique, il faut en effet recourir à des projections drastiques de l'information topologique, ce qui fait perdre une grande partie de l'intérêt initial de l'approche qui était justement de pouvoir accéder à toute l'information topologique contenue dans les graphes. Par ailleurs les procédures de calcul semblent exagérément dispendieuses par rapport aux objectifs puisque les sous-graphes projetés sont systématiquement calculés alors que la plupart d'entre eux ne révèlent aucun intérêt a posteriori. En comparaison, l'efficacité annoncée des méthodes de fouille de graphes, et plus particulièrement des méthodes de recherche de sous-graphes fréquents (Yan et Han, 2002; Nijssen et Kok, 2004), est apparue comme un atout très séduisant pour fouiller les BdR.

2.4 Les méthodes de fouille (de motifs) de graphes

Les méthodes `Subdue` (Cook et Holder, 1994) et `BGI` (Yoshida *et al.*, 1994) sont souvent citées comme les premières méthodes de fouille de graphes à avoir été proposées. Certes ces méthodes étaient plutôt issues de la communauté d'IA et étaient destinées à résoudre des problèmes d'apprentissage symbolique plutôt que d'extraction de connaissances à proprement parler. Par ailleurs ces méthodes n'étaient pas adaptées au traitement de grandes quantités de données (même avec les ordinateurs d'aujourd'hui, cf le test à la section 6.1.2). Mais l'originalité de ces méthodes était de pouvoir fouiller des motifs (i.e. des sous-graphes) contenus dans un ensemble de graphes, sans recourir à des modèles plus complexes comme la logique des prédicats. Ces travaux tranchaient en particulier avec les approches « bottom-up » de généralisations successives de motifs de graphes notamment utilisées par la communauté de reconnaissance des formes, pour traiter des problèmes de classification supervisée de formes modélisables par des graphes. Ensuite la notion de fouille de motifs de graphes, plus communément appelée fouille de graphes (Cook et Holder, 2000), est apparue très progressivement avec celle de la fouille de données. Historiquement, la fouille de graphes n'a véritablement émergée que lorsque les premiers algorithmes sont apparus à partir du début des années 2000 pour rechercher des motifs fréquents dans des ensembles de séquences, puis d'arbres et enfin les graphes, ces derniers englobant les premiers. Aujourd'hui la fouille (de motifs) de graphes peut être considérée comme une des principales branches de la fouille de relations ou de liens

(d'après Getoor et Diehl (2005), cf section 2.2.1). Elle regroupe différents problèmes de classification et d'extraction de connaissances dont la résolution passe par la recherche de motifs dans des données représentées par des graphes et en particulier par le problème central de la recherche de sous-graphes fréquents.

2.4.1 Le problème de la recherche des sous-graphes fréquents

Définition 2.4.1. Le problème de la *recherche de sous-graphes fréquents* se définit à partir :

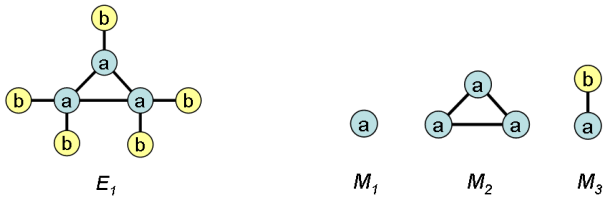
1. D'un ensemble $(\mathcal{G}, \leq_G)$ des motifs de graphes étiquetés sur un ensemble $\mathcal{L}$ et ordonnés par la relation $\leq_G$ de sous-graphe isomorphe (partiel, induit ou autre).
2. De données spécifiées par un ensemble d'objets et une application $\delta : \mathcal{O} \to \mathcal{G}$ associant à chaque objet un graphe qui le décrit. Pour simplifier les notations, on utilisera en lieu du couple $(\mathcal{O}, \delta)$ le multi-ensemble $\mathcal{D} = \{\delta(o) | o \in \mathcal{O}\}$ de graphes pour faire référence aux données.
3. D'une fonction freq : $\mathcal{G} \to \mathbb{R}^+$ qui associe à tout graphe connexe $g \in \mathcal{G}$ sa fréquence freq(g) de telle sorte que cette fonction soit décroissante dans $(\mathcal{G}, \leq_G)$ et dépende des occurrences de g dans $\mathcal{D}$ (cette fonction sera explicitée dans la section suivante).
4. D'un seuil de fréquence minimale $f_{min} \in \mathbb{R}^+$.

Le problème consiste alors à déterminer l'ensemble des motifs **connexes** de $\mathcal{G}$ fréquents (i.e. dont la fréquence freq(g) est supérieure ou égale à f_{min}) ainsi que leur fréquence associée.

La contrainte exigeant que les motifs soient connexes n'est pas indispensable mais communément admise afin de limiter la combinatoire déjà très élevée du problème. Par ailleurs l'extraction des motifs fréquents servant à exprimer des corrélations entre occurrences, par exemple à travers des règles d'association, ces corrélations au sein des graphes s'expriment généralement de proche en proche d'un sommet à ses voisins de sorte qu'il n'est pas aberrant de se limiter au cas des motifs connexes. Quant à la décroissance de la fréquence, elle est indispensable pour garantir l'anti-monotonicité du caractère fréquent des motifs et ainsi la complétude des algorithmes de recherche des motifs fréquents.

La fonction freq de fréquence n'est pas explicitée dans l'énoncé précédent car elle admet de multiples définitions, dues à la richesse combinatoire que présentent les graphes. Vanetik *et al.* (2006) caractérisent les propriétés que doit satisfaire une telle fonction pour être acceptable, c'est à dire décroissante (au sens large) dans l'ordre des motifs et en donnent plusieurs exemples. Si on omet cette exigence de décroissance, la définition la plus immédiate de la fréquence d'un graphe g serait soit le nombre occ(g) de morphismes injectifs de g vers des graphes de $\mathcal{D}$ (tels que définis dans la section 2.3.1), soit le nombre nsgr(g) de sous-graphes de graphes de $\mathcal{D}$ qui sont isomorphes à g. Ces sous-graphes sont généralement choisis comme partiels (cf section 2.3.1) même s'ils peuvent dans certains cas être contraints à être induits (Inokuchi *et al.*, 2000). Ces deux fonctions occ et nsgr sont égales sauf lorsque le motif présente une symétrie interne. Dans ce cas en effet le groupe automorphique Aut(g) de g (c'est à dire le sous-groupe des isomorphismes de g vers g) n'est pas réduit à l'identité. Il existe alors autant de morphismes injectifs qui envoie g vers un sous-graphe de $\mathcal{D}$ isomorphe à g qu'il y a d'automorphismes dans Aut(g). Pour cette raison occ$(g) =$ nsgr$(g) \times |\,\text{Aut}(g)|$, de sorte qu'en l'absence de symétrie interne, Aut$(g) = \{\text{Id}\}$ et donc occ$(g) =$ nsgr(g). Les fonctions occ et nsgr ne satisfont toutefois pas l'exigence de décroissance dans l'ordre des motifs. À titre d'exemple, la figure 2.17 considère trois motifs M_1, M_2 et M_3 dont on cherche à déterminer la fréquence (au sens des sous-graphes partiels isomorphes) dans un jeu de données constitué d'un seul exemple

E_1. Alors que M_1 est isomorphe à un sous-graphe de M_2 et de M_3, $\text{occ}(M_1) < \text{occ}(M_2)$ et $\text{occ}(M_1) < \text{occ}(M_3)$. Même en tenant compte des symétries internes des motifs, le problème n'est pas résolu puisque $\text{nsgr}(M_1) < \text{nsgr}(M_3)$. Dans le cas où les données sont constituées



(a) Exemple (b) Motifs

Motifs	M_1	M_2	M_3
$\text{occ}(M)$	3	6	5
$\text{nsgr}(M)$	3	1	5
$\text{freq}(M)$	1	1	1
$\sigma_\wedge(M)$	3	3	3

(c) Fréquences

FIG. 2.17 – Les différentes définitions de la fréquence d'un graphe (au sens des sous-graphes partiels)

de nombreux graphes indépendants, comme c'est le cas des bases de données de molécules ou de réactions chimiques, il est possible de se ramener à une définition de la fréquence qui soit anti-monotone et similaire dans son principe à la fréquence des motifs d'attributs : la fréquence $\text{freq}(M)$ d'un motif M est le nombre ou la proportion de graphes de $\mathcal{D}$ contenant au moins un sous-graphe isomorphe au motif M. Cette définition est toutefois inutilisable par les applications qui considèrent la donnée d'un seul graphe très grand tel qu'un réseau social ou biologique, la fréquence ne pouvant alors valoir que 0 ou 1. Fiedler et Borgelt (2007) et Bringmann et Nijssen (2008) proposent des mesures de fréquence décroissantes applicables à un seul graphe. La fréquence $\sigma_\wedge(M, g)$ d'un motif M dans un graphe g défini par Bringmann et Nijssen (2008) a le mérite d'être particulièrement simple et élégante. Cette fréquence définie par $\sigma_\wedge(M, g) = \min_{v \in V(M)} |\phi_i(v) : \phi_i \text{ est un morphisme injectif de } M \text{ dans } g|$ compte pour chaque sommet v du motif M, le nombre de sommets de g qui sont l'image de v par un morphisme injectif de M dans g et extrait la valeur minimale atteinte. Ainsi pour l'exemple de la figure 2.17), $\sigma_\wedge(M_2, E_1) = \min(\{3, 3, 3\}) = 3$ et $\sigma_\wedge(M_3, E_1) = \min(\{3, 5\}) = 3$. On a bien $\sigma_\wedge(M_1, E_1) \geq \sigma_\wedge(M_2, E_1)$ et $\sigma_\wedge(M_1, E_1) \geq \sigma_\wedge(M_3, E_1)$.

2.4.2 Les algorithmes de recherche de sous-graphes fréquents

Le principe de la recherche des motifs d'attributs fréquents (cf section 2.1.2) a pu être généralisé à des familles de motifs relationnels de plus en plus complexes. Différentes méthodes ont ainsi été proposées pour traiter tour à tour les séquences, puis les arbres et enfin les graphes. L'avantage de ces méthodes par rapport à la PLI est de considérer des familles de motifs relativement plus réduites que l'ensemble des théories de prédicats et pour lesquelles il existe des procédures efficaces d'énumération exhaustive des motifs. Les trois sous-sections suivantes donnent respectivement un aperçu de la recherche de séquences, d'arbres puis de graphes fréquents dans l'ordre de leur apparition historique mais aussi dans l'ordre croissant de leur complexité[18]. Cette richesse combinatoire est d'ailleurs la source de nombreuses interprétations du problème de la recherche des motifs fréquents, dont les variations dépendent principalement des contraintes structurelles que doit satisfaire la famille de motifs considérée

[18]Une approche allant du général au spécifique suivrait plutôt l'ordre inverse, les séquences pouvant être vues comme une famille particulière d'arbres qui eux-mêmes forment une famille particulière de graphes.

d'une part et des différentes définitions que peut revêtir la fréquence d'un motif d'autre part. Si certains travaux proposent d'unifier tous ces problèmes au sein d'un modèle et d'une plate-forme logicielle unique (Zaki *et al.*, 2005), une telle démarche semble difficilement réalisable tant la variété des problèmes posés entraînent une variété dans les solutions trouvées. Les sections suivantes illustrent cette multitude d'interprétations en rattachant à chacune d'entre elles les quelques algorithmes les plus représentatifs.

La recherche de motifs de séquences fréquents

Intuitivement, une séquence est une suite d'attributs ou d'ensembles d'attributs et un motif séquentiel inclus dans une séquence est une sous-séquence partielle qui respecte l'ordre d'apparition des attributs dans la séquence initiale. Si la prédiction de motifs séquentiels est un sujet traité de longue date par la communauté d'intelligence artificielle, c'est encore une fois Agrawal et Srikant (1995) qui en premier proposent de chercher de façon exhaustive les motifs fréquents dans des séquences de transactions. Comme pour le problème de recherche des attributs fréquents, Agrawal et Srikant considèrent le cas de transactions commerciales décrites chacune par un ensemble d'attributs (i.e. les articles achetés). La différence tient au fait que l'ordre dans lequel sont effectuées les transactions d'un même client est cette fois ci pris en compte. Les motifs séquentiels recherchés sont alors des séquences de motifs d'attributs $(M_1, \cdots M_n)$. Un motif séquentiel $(A_1, \ldots A_n)$ est inclus dans un motif $(B_1, \ldots B_m)$ s'il existe une sous-séquence $(B_{i_1}, \ldots B_{i_n})$ de $(B_1, \cdots B_m)$ pour laquelle $A_j \subseteq B_{i_j}$ pour tout j variant de 1 à n. La fréquence d'un motif séquentiel est alors le nombre de clients dont la séquence de transactions contient le motif séquentiel. Le problème consiste à trouver l'ensemble des motifs séquentiels fréquents maximaux (au sens de l'inclusion). La figure 2.18 illustre ce problème. Le

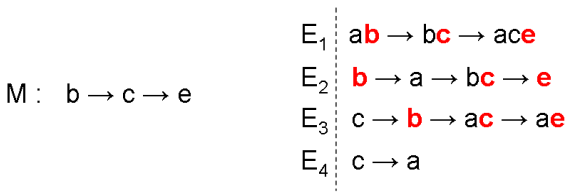


FIG. 2.18 – Exemple d'un motif M séquentiel fréquent maximal pour $f_{min} = 3$ (selon Agrawal et Srikant (1995))

motif M y a un support de 3 vis-à-vis des quatre exemples. Si le seuil de fréquence minimale f_{min} est 3, M est donc un fréquent maximal puisque tout motif incluant M voit sa fréquence chuter à 2 ou moins. La fréquence restant une fonction décroissante dans l'ordre des motifs, Agrawal et Srikant adaptent leur algorithme `Apriori` au cas des motifs séquentiels pour résoudre le problème.

Mannila *et al.* (1995); Mannila et Toivonen (1996) proposent une variante du problème où les objets sont des événements (i.e. un attribut unique) apparaissant sur une chronique unique. Les objets d'étude sont alors constitués de toutes les sous-séquences d'événements que l'on obtient en intersectant la chronique avec des fenêtres temporelles d'une durée fixée. La fréquence d'un motif séquentiel est le nombre de ces sous-séquences contenant le motif. Là encore le trait fréquent est anti-monotone dans l'ordre des motifs séquentiels et un algorithme de recherche similaire à `Apriori` est proposé. Srikant et Agrawal (1996) reprennent dans leur algorithme `GSP` cette idée de fenêtre temporelle tout en introduisant une taxonomie sur les attributs et en assouplissant la contrainte séquentielle : deux transactions suffisamment

rapprochées dans le temps peuvent fusionner. Han *et al.* (2000a) adoptent l'approche diviser-pour-régner de leur algorithme `FP-Growth` de recherche de motifs d'attributs fréquents (Han *et al.*, 2000b, 2004) pour proposer `FreeSpan`, un algorithme de recherche de motifs séquentiels fréquents plus efficace que `GSP`. Plus tard l'algorithme `PrefixSpan` (Pei *et al.*, 2004b) reprend les principes de `FreeSpan` en énumérant les motifs séquentiels selon un ordre lexicographique plus efficace. Dans les deux cas la division du problème consiste à parcourir en profondeur une partition arborescente de l'espace des motifs séquentiels. Cette division permet alors de ne fouiller à chaque étape de la récursion que la restriction D des données faites des fragments de séquences contenant le motif courant M. Cette restriction permet de calculer en une passe la fréquence de tous motifs fils de M dans D et de ne développer récursivement que les motifs fils avérés fréquents. De la même manière, Zaki procède à une adaptation du principe de stockage vertical des données déjà utilisé dans son algorithme `Eclat` (Zaki *et al.*, 1997) de recherche de motifs d'attributs fréquents pour proposer l'algorithme `Spade` (Zaki, 2001) de recherche de motifs séquentiels.

En terme d'applications, la recherche de motifs séquentiels fréquents a notamment servi à des tâches de classification de données séquentielles (Lesh *et al.*, 2000). La recherche de motifs séquentiels fréquents est aussi étroitement liée au problème de la recherche de chemins fréquents – i.e. une séquence de sommets adjacents distincts – dans un ensemble de graphes, dont la classification de molécules est une application. La seule différence d'un chemin par rapport à une séquence est que le sens de parcours du chemin d'un bout à l'autre n'importe pas. Ainsi l'algorithme `MolFea` (Kramer *et al.*, 2001) permet de chercher les chemins d'atomes éléments d'un espace de versions (Mitchell, 1997) satisfaisant un certain nombre de contraintes monotones et anti-monotones. En particulier `MolFea` intègre la contrainte anti-monotone $\text{freq}^+_{\mathcal{D}}(M) \geq f_{min}$ de fréquence d'un motif M et celle monotone $\text{freq}^-_{\mathcal{D}}(M) \leq f_{max}$ de non-fréquence vis-à-vis d'ensembles $\mathcal{D}^+$ et $\mathcal{D}^-$ d'exemples positifs et négatifs de graphes moléculaires. L'algorithme adopte une recherche par niveau similaire à `Apriori` (Agrawal *et al.*, 1993a) pour établir cet espace des versions. Des travaux ont également permis de généraliser la notion de motifs fermés (Yan *et al.*, 2003) aux motifs séquentiels et de motifs générateurs (Raïssi *et al.*, 2008) aux conjonctions de motifs séquentiels.

La recherche de motifs d'arbres fréquents

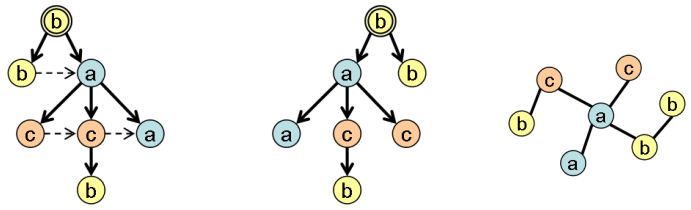
Le principe des motifs fréquents a également été appliqué au cas où les données sont décrites par des arbres étiquetés, c'est à dire des graphes acycliques connexes. Cet intérêt pour les arbres n'est pas fortuit : les arbres sont des structures fondamentales dans la recherche d'information, permettant de représenter les documents semi-structurés et les requêtes associées, c'est à dire les documents fondés sur les langages à parenthèses (ou à balises) du type XML, HTML .... De ce fait des travaux (Wang et Liu, 1998; Zhou *et al.*, 1999; Wang et Liu, 2000) ont tenté très tôt de décrire et classifier ce type de documents à l'aide de motifs d'arbres fréquents. En outre la recherche des motifs d'arbres fréquents a servi de tremplin à la fouille de graphes dans la mesure où les arbres présentent une complexité combinatoire intermédiaire entre celle des séquences et celle des graphes cycliques. Cette richesse combinatoire se traduit par de multiples formulations possibles du problème de la recherche de motifs d'arbres fréquents en particulier selon que les arbres considérés sont ordonnés, enracinés ou libres.

- Un *arbre libre* est un graphe non orienté acyclique.
- Un *arbre enraciné* est un arbre dans lequel un nœud a été identifié comme le nœud racine. Cette racine définit une orientation des arêtes selon une relation de descendance

unique et induit les notions de nœuds parents, enfants, descendants, racine et feuilles ...

– Un *arbre ordonné* est un arbre enraciné pour lequel l'ordre d'énumération des nœuds enfants d'un nœud parent importe.

A un arbre libre à n sommets correspond donc n arbres enracinés qui eux-mêmes correspondent à autant d'arbres ordonnés qu'il y a de permutations de relations parent-enfant. La figure 2.19 illustre ces trois familles d'arbres. La définition de la relation d'inclusion d'un



(a) Arbre ordonné (b) Arbre enraciné (c) Arbre libre

FIG. 2.19 – Différents familles d'arbres

motif d'arbre T_m dans un arbre des données T_d peut également varier et donner ainsi lieu à différentes définitions de la fréquence. Les trois définitions suivantes de la relation d'inclusion correspondent aux trois variations les plus courantes. Ainsi T_m est inclus dans T_d s'il existe :

– Soit un *sous-arbre induit* de T_d isomorphe à T_m, c'est à dire un arbre formé d'un sous-ensemble d'arcs ou d'arêtes de T_d. Cette définition s'applique à tout type d'arbre, libre, enraciné ou ordonné.

– Soit un *sous-arbre complet* de T_d isomorphe à T_m, c'est à dire un sous-arbre induit de T_d tels que tous les descendants dans T_d de la racine de T sont aussi dans T. Cette définition s'appuyant sur la notion de descendance, ne peut s'appliquer qu'aux arbres enracinés ou ordonnés.

– Soit un *arbre enchâssé* dans T_d isomorphe à T_m, un arbre T enchâssé de T_m étant défini par un sous-ensemble de sommets de T_m tels que un arc relie un nœud n_1 à un un nœud n_2 si et seulement si n_2 est un descendant de n_1. Cette définition plus générale que la première et donc que la seconde, s'applique uniquement aux arbres enracinés ou ordonnés dont les arcs ne sont pas étiquetées.

La figure 2.20 illustre les différentes relations d'inclusion dans le cas d'un arbre enraciné.

(a) Sous-arbre complet (b) Sous-arbre induit (c) Sous-arbre enchâssé (d)

FIG. 2.20 – Différentes relations d'inclusion dans un arbre enraciné (d)

Chaque sous-problème a donné lieu à un ou plusieurs algorithmes dont les complexités n'ont cessé d'aller en décroissant (Chi *et al.*, 2005a). Le tableau 2.21 recense les principaux algorithmes de recherche de motifs d'arbres fréquents en fonction de la famille de motifs recherchée, du type d'arbres et de relation d'inclusion considérés. Les algorithmes associés au

Famille de motifs	Type d'arbres	Type d'inclusion	Algorithmes
fréquents	ordonnés	induits	`FREQT` (Asai *et al.*, 2002), `PathJoin`♣ (Xiao *et al.*, 2003)
		enchâssés	`TreeMiner` (Zaki, 2002, 2005b)
	enracinés	induits	`Unot` (Asai *et al.*, 2003), `uFreqt` (Nijssen et Kok, 2003), `HybridTreeMiner` (Chi *et al.*, 2004a)
		enchâssés	`TreeFinder` (Termier *et al.*, 2002) (incomplet), `SLEUTH` (Zaki, 2005a)
	libres	induits	`FreeTreeMiner` (Chi *et al.*, 2003; Rückert et Kramer, 2004), `HybridTreeMiner` (Chi *et al.*, 2004a), `Gaston` (Nijssen et Kok, 2004)
fermés fréquents	ordonnés	induits	`DryadeParent`♣ (Termier *et al.*, 2008)
		enchâssés	`Dryade`♣ (Termier *et al.*, 2004)
	enracinés	induits	`CMTreeMiner` (Chi *et al.*, 2004b, 2005b)

FIG. 2.21 – Les principaux algorithmes de recherche d'arbres fréquents

symbole ♣ supposent que les arbres sont seulement enracinés sans être nécessairement ordonnés mais supposent par ailleurs que deux nœuds frères ne peuvent avoir la même étiquette afin d'éviter les problèmes de représentations canoniques des arbres enracinés. Si on se donne un ordre arbitraire sur les étiquettes, de tels arbres forment de façon univoque une sous-famille des arbres ordonnés. Du point de vue de la complexité combinatoire, ce type d'algorithme est donc rattaché aux arbres ordonnés plutôt qu'aux arbres enracinés. Si les algorithmes du tableau 2.21 se distinguent par les problèmes qu'ils abordent, la plupart de ces algorithmes se fondent sur des principes de conception ou d'optimisation similaires. Le phénomène essentiel que font apparaître les arbres enracinés et libres est le problème d'isomorphisme entre motifs qui ne se posait pas ni pour les motifs d'attributs ni pour les motifs de séquence, ni même pour les arbres ordonnés.

Dans le cas des arbres ordonnés, il n'y a en effet pas de problème d'isomorphisme. Les langages à parenthèses permettent une représentation canonique en bijection avec les arbres ordonnés. Un tel codage consiste à énumérer les étiquettes des nœuds d'un arbre ordonné selon un parcours en profondeur de gauche à droite, en insérant des symboles ↑ pour indiquer un retour arrière d'un niveau vers le haut. Le code de l'arbre ordonné de la figure 2.19(a) est ainsi $c(T_1) = bb{\uparrow}ac{\uparrow}cb{\uparrow}{\uparrow}a{\uparrow}{\uparrow}{\uparrow}$. Ce code est bijectif puisqu'il est possible de reconstruire un arbre ordonné à partir d'un code bien formé, c'est à dire d'un code comprennant autant de symboles ↑ que d'étiquettes. Du fait de cette bijection, les codes bien formés forment un *codage canonique* des arbres ordonnés. L'avantage d'un tel codage bijectif est de pouvoir facilement énumérer tous les arbres ordonnés une et une seule fois sans omettre de motifs ni générer de doublons, grâce à l'ajout d'un nœud le long de « branche la plus à droite » [19] (Asai *et al.*, 2002). Cette énumération non redondante définit un *arbre d'énumération* dans l'ordre des motifs tel que chaque motif soit généré par un seul motif, qualifié de motif parent. La

[19] Rightmost extension en anglais.

principale difficulté que doivent surmonter les algorithmes de recherche des motifs fréquents en présence d'isomorphisme, qu'ils soient des arbres ou des graphes, est de pouvoir induire un tel arbre d'énumération pour la famille de motifs considérés.

Le cas des arbres enracinés est donc plus compliqué. En effet l'ordre des nœuds frères d'un arbre enraciné n'importe plus de sorte que les sous-arbres associés dont ces nœuds sont les racines peuvent être permutés indifféremment. Exprimé de façon plus formelle, les arbres enracinés sont en bijection avec les classes d'équivalence des arbres ordonnés isomorphes, se déduisant les uns des autres par permutations de nœuds frères. Le problème général posé par les relations d'isomorphisme est qu'on ne dispose pas d'un codage canonique en bijection avec l'espace quotient $\mathcal{M}/\sim$ des motifs (i.e. l'ensemble des classes des motifs isomorphes, ici les arbres enracinés) mais seulement avec l'ensemble $\mathcal{M}$ des motifs (ici, l'ensemble des arbres ordonnés). Une des techniques pour obtenir malgré tout un codage canonique bijectif sur $\mathcal{M}/\sim$, consiste à partir d'un codage canonique des motifs $\mathcal{M}$, à considérer la relation $\leq_C$ d'ordre lexicographique sur l'espace des codes produits par ce codage et enfin à définir le code canonique d'une classe d'équivalence $\overline{M} \in \mathcal{M}/\sim$ comme le code minimal, au sens de $\leq_C$, parmi tous les codes des motifs isomorphes de $\overline{M}$. Ainsi le code canonique de l'arbre enraciné sous-jacent à l'arbre ordonné de la figure 2.19(a) devient $c(T_2) = baa{\uparrow}cb{\uparrow}{\uparrow}c{\uparrow}{\uparrow}b{\uparrow}{\uparrow}$. Ce code correspond à l'arbre de la figure 2.19(b) en supposant que celui-ci soit ordonné. Toute permutation de nœuds frères de cet arbre conduit à un code supérieur, à l'image de $c(T_1) >_C c(T_2)$. Ce codage canonique correspond au codage canonique des arbres ordonnés défini plus haut si l'ordre alphabétique dont découle l'ordre lexicographique est tel que $a < b < c < \uparrow$. Dans la mesure où le nombre d'arbres ordonnés isomorphes peut être très grand (exponentiel avec la taille de l'arbre dans le pire cas), il est impossible de calculer tous les codes canoniques de tous les arbres ordonnés pour en extraire le code minimum. Heureusement une approche « branch and bound » permet d'éliminer un grand nombre de codages candidats et de calculer rapidement le code canonique minimal. Il est même possible de se passer de ce calcul grâce à une procédure d'énumération sans doublon des arbres enracinés en temps amorti constant (Asai *et al.*, 2003; Nijssen et Kok, 2003; Chi *et al.*, 2004a). Le problème se complique encore Dans le cas des arbres libres, le problème se complique mais se traite à l'identique : le code canonique d'un arbre libre est le code minimal des codes canoniques des arbres enracinés sous-jacents. Ainsi le code canonique de l'arbre libre de la figure 2.19(c) devient $c(T_3) = aabb{\uparrow}{\uparrow}cb{\uparrow}{\uparrow}c{\uparrow}{\uparrow}{\uparrow}$ inférieur à $c(T_2)$.

La recherche de motifs de graphes fréquents

Des trois familles de motifs considérés (les séquences, les arbres et les graphes connexes), les graphes connexes posent le problème le plus complexe. Ainsi, en autorisant la présence de cycles, le meilleur algorithme (McKay, 1981) de calcul d'une représentation canonique bénéficie d'une complexité au pire exponentielle alors que des algorithmes polynomiaux sont connus dans le cas des arbres. Paradoxalement alors que le problème de la recherche d'arbres fréquents admet de nombreuses variantes dans sa définition, les algorithmes de recherche de sous-graphes fréquents traitent pour la plupart du même problème. En effet la notion d'arbre ordonné ou enraciné ainsi que la notion d'arbre enchâssé supposent une relation de parenté qui se transpose difficilement au cas des graphes (cycliques). Le problème auquel se rapporte le plus la recherche des sous-graphes fréquents est donc la recherche des arbres libres induits dans lequel on autoriserait la présence de cycles dans les motifs. Le tableau 2.22 présente les principaux algorithmes de recherche de sous-graphes fréquents dans l'ordre de leur apparition chronologique.

Algorithme
`AGM` (Inokuchi *et al.*, 2000)
`FSG` (Kuramochi et Karypis, 2001, 2004a)
`Mofa` (Borgelt et Berthold, 2002)
`gSpan` (Yan et Han, 2002)
`FFSM` (Huan *et al.*, 2003)
`Gaston` (Nijssen et Kok, 2004)

FIG. 2.22 – Les principaux algorithmes de recherche de sous-graphes fréquents

Le premier algorithme `AGM` (Inokuchi *et al.*, 2000) a consisté à adapter l'algorithme `Apriori` au cas où les motifs sont des graphes connexes. Une des originalités d'`AGM` vis-à-vis des algorithmes qui suivront est de définir sa fréquence comme le nombre d'exemples contenant au moins un sous-graphe **induit** (et pas seulement partiel) isomorphe au motif[20]. Le niveau k est représenté par l'ensemble des matrices d'adjacence d'ordre k dans une forme dite normale, sachant que plusieurs formes normales peuvent représenter un même graphe à un isomorphisme près. `AGM` définit une opération de jointure entre deux formes normales d'ordre k pour en créer une troisième d'ordre $k+1$ puis vérifie que toutes les formes normales d'ordre k contenues dans la forme normale d'ordre $k+1$ correspondent à des motifs fréquents. Enfin il calcule la fréquence d'un graphe en sommant les fréquences de toutes les formes normales représentant ce graphe grâce au calcul d'une matrice canonique associée à chaque forme normale. Si `AGM` a le mérite d'avoir été le premier algorithme de recherche de graphes fréquents, il présente certains inconvénients : recours à des formes normales plus nombreuses que les motifs, opérateur de jointure peu précis entraînant de nombreuses générations redondantes de motifs isomorphes, grande consommation de mémoire pour stocker toutes les formes normales dont le nombre croit exponentiellement avec le niveau k ...

L'algorithme `FSG` (Kuramochi et Karypis, 2001, 2004a) pallie un certain nombre des défauts de jeunesse que présentait `AGM`. Tout comme ce dernier, `FSG` est un algorithme par niveau qui utilise un opérateur de jointure et un filtrage des candidats selon le principe de « downward closure » (i.e. tout motif ne peut être fréquent que si tous ses sous-motifs sont fréquents). Contrairement à `AGM`, `FSG` définit la fréquence à partir de la relation de sous-graphe partiel isomorphe. De plus la génération des candidats est cette fois-ci définie à un isomorphisme près (i.e. deux candidats distincts ne peuvent être isomorphes) grâce à un codage canonique. L'algorithme utilise une opération de jointure complexe qui permet de générer les graphes connexes à k sommets à partir de l'union de deux graphes de $k-1$ sommets qui contiennent un sous-graphe commun isomorphe de $k-1$ sommets. L'opérateur est pensé de sorte qu'il diminue le nombre de générations redondantes sans toutefois les éliminer et sans compromettre la complétude de l'algorithme (voir démonstration en annexe de Kuramochi et Karypis (2004a)). Enfin l'algorithme associe à chaque graphe g de taille k l'ensemble $E(g)$ des exemples qui contiennent un sous-graphe isomorphe à g. Étant donné le graphe $g_1 \cup g_2$ résultant de l'union de deux graphes g_1 et g_2, l'intersection des ensembles $E(g_1) \cap E(g_2)$ est alors calculé. Comme $E(g_1 \cup g_2) \subseteq E(g_1) \cap E(g_2)$, il est possible d'éliminer le graphe candidat $g_1 \cup g_2$ si le nombre d'exemples de $E(g_1) \cap E(g_2)$ est suffisamment faible pour empêcher $g_1 \cup g_2$ d'être fréquent. Dans le cas contraire, la présence d'un sous-graphe isomorphe à $g_1 \cup g_2$ peut

[20] De ce fait, l'ensemble des motifs de niveau k (au sens de `Apriori`) sont les graphes connexes à k sommets et non à k arêtes comme cela serait le cas si la relation de sous-graphe partiel isomorphe était choisie en lieu de la relation de sous-graphe induit isomorphe.

uniquement être recherchée dans les exemples de $E(g_1) \cap E(g_2)$.

L'algorithme `Mofa` (Borgelt et Berthold, 2002) a été conçu spécialement pour la recherche des fragments fréquents de graphes moléculaires. Il s'agit d'un algorithme de parcours en profondeur qui développe un motif courant en lui ajoutant de nouvelles arêtes. Il introduit le concept de listes d'occurrences qui peut être vu comme le prolongement du codage vertical utilisé par `Eclat` (Zaki, 2000) dans le cas des motifs d'attributs. Une liste d'occurrences consiste à mémoriser l'ensemble des morphismes injectifs d'un motif dans les données. Chaque occurrence du motif M dans un graphe g est ainsi décrite par l'ensemble des sommets de g qui correspondent au sous-graphe isomorphe à M. À partir de la liste d'occurrences, il est facile de déduire la fréquence du motif comme le nombre de graphes des données pour lesquels il existe au moins un morphisme injectif du motif. L'avantage d'une telle structure est de pouvoir se mettre à jour très facilement lorsqu'une extension d'un sommet ou d'une arête est appliquée au motif courant M. La figure 2.23 illustre le fonctionnement d'une liste d'occurrences des motifs successifs g_0 à g_5 obtenus par 5 extensions successives dans trois graphes e_1, e_2 et e_3. Lors de chaque extension du motif, la liste L_i des occurrences de g_i est parcourue pour construire la liste L_{i+1}. L'extension qui transforme g_i en g_{i+1} est alors appliquée à chaque occurrence de L_i. Chaque extension réussie donne lieu à une nouvelle occurrence de g_{i+1}, stockée dans la liste L_{i+1}. Lorsqu'une extension ajoute un nouveau sommet, chaque élément de la liste L_{i+1} stocke l'indice du sommet nouvellement apparié dans les données. La figure 2.23 est une représentation plus compacte de la solution proposée initialement par Borgelt et Berthold résultant des améliorations apportées successivement par Nijssen et Kok (Nijssen et Kok, 2004) puis par nous-mêmes dans l'outil logiciel baptisé `Forage`. Le fait que les listes d'occurrences peuvent prendre une place non négligeable en mémoire (plusieurs dizaines de mégaoctets pour des données de quelques milliers de graphes) empêche de maintenir à jour les listes d'occurrences d'un nombre important de motifs. Pour cette raison les listes d'occurrences ne sont compatibles qu'avec les algorithmes de parcours en profondeur qui ne développent qu'un seul motif, contrairement aux algorithmes par niveau qui mémorisent un niveau entier de motifs fréquents. Malgré le recours aux listes d'occurrences, l'algorithme `Mofa` est toutefois peu efficace car il ne détecte pas la génération redondante de motifs isomorphes. Il est en effet difficile pour un algorithme en profondeur de savoir si le motif courant a déjà été généré à un isomorphisme près en suivant une autre branche de développement. Il est possible de stocker tous les représentants canoniques des graphes déjà fouillés dans un dictionnaire et ainsi de détecter une double génération de motifs en consultant ce dictionnaire. Mais cette solution est limitée car la taille du dictionnaire croit avec le nombre de motifs fréquents trouvés à un point que l'espace mémoire vient rapidement à manquer.

Yan et Han (2002) trouve une solution élégante à ce problème en proposant l'algorithme `gSpan` qui est un autre algorithme de parcours en profondeur cette fois ci fondé sur un codage canonique (cf section 2.3.1) appelé code DFS (pour Depth First Search). Succinctement un graphe g peut se définir comme la composition $g = (e_n \circ \cdots \circ e_1)(\emptyset)$ d'extensions e_i qui s'appliquent au graphe vide $\emptyset$. Une extension consiste à connecter un nouveau sommet à un sommet existant du motif ou à connecter deux sommets existants par une nouvelle arête. Dans la mesure où ces extensions définissent l'ordre d'apparition des sommets et donc leur numérotation, la donnée d'une suite particulière d'extensions conduisant à la construction de g revient à définir d'un codage canonique $c(g) = (e_1, \dots e_n)$ de g. L'idée maîtresse de `gSpan` est d'établir ce codage canonique DFS afin qu'il revête la propriété particulière suivante : si un motif g a pour code canonique une séquence $c(g) = (e_1, \dots, e_n)$ alors le graphe $g' = (e_{n-1} \circ \cdots \circ e_1)(\emptyset)$ a pour code canonique $c(g') = (e_1, \dots e_{n-1})$. La génération de g est alors confiée à g'. Lorsque une extension e_n est appliquée à ce motif g', il suffit pour savoir si le

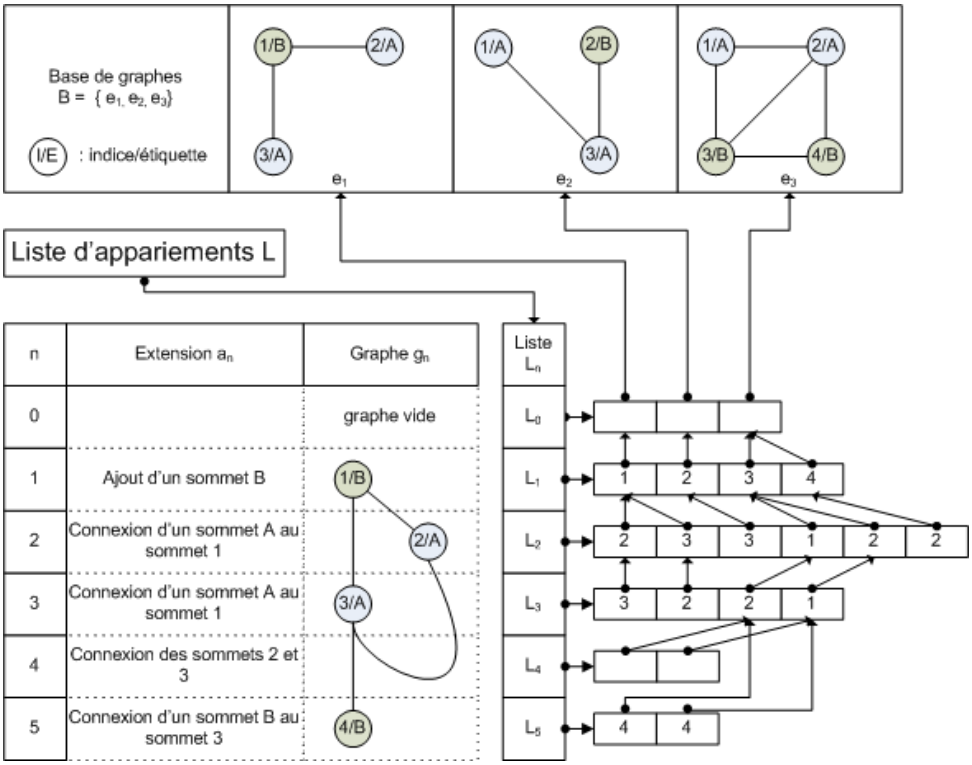


FIG. 2.23 – Listes d'occurrences

motif résultant $g = (e_n \circ \cdots \circ e_1)(\emptyset)$ a déjà été généré, de calculer son code canonique $c(g)$ et de vérifier qu'il est bien égal à $(e_1, \ldots, e_n)$. Dans le cas contraire, on sait que le motif g sera généré en suivant une autre branche d'exploration et un retour arrière peut être effectué sans compromettre la complétude de `gSpan`. La conséquence est que n'ayant pas à stocker en mémoire un ensemble de motifs candidats – comme c'est le cas des algorithmes `AGM` et `FSG` de recherche par niveau – ni à se rappeler dans un dictionnaire les motifs déjà fouillés – comme cela devrait être le cas de `Mofa` – `gSpan` consomme une quantité de mémoire qui reste faible et bornée tout au long de la fouille des données. Tout comme les algorithmes de recherche en profondeur `FFSM` et `Gaston` qui suivront, la force de `gSpan` est, au delà de sa rapidité, le fait de ne plus être limité par le nombre de motifs fréquents qu'il peut extraire. Le principal facteur limitant devient alors le temps de calcul et l'espace disque pour stocker des millions voire des milliards de motifs ! Par ailleurs `gSpan` n'utilise pas les listes d'occurrences mais calcule la fréquence des motifs à la volée en s'aidant de la liste déjà utilisée par `FSG` des exemples qui contiennent le motif courant. `ADI-Mine` (Wang *et al.*, 2004) est une version améliorée de `gSpan` qui réalise une partition de la base d'exemples afin de traiter des bases pouvant atteindre un million de graphes.

Tout comme `gSpan`, `FFSM` (Huan *et al.*, 2003) opte pour un parcours en profondeur de l'ordre des motifs et utilise un codage canonique efficace des graphes cette fois ci fondé sur les matrices d'adjacence. La principale originalité par rapport à `gSpan` est de générer certains motifs non par extension mais par jointure en proposant des opérateurs adaptés au codage canonique proposé. L'algorithme utilise par ailleurs des listes d'occurrences pour calculer la fréquence des motifs.

Enfin les auteurs de `Gaston` (Nijssen et Kok, 2004) partent du constat que seule une faible

proportion des motifs de graphes fréquents sont cycliques. Leur idée consiste alors à énumérer selon un parcours en profondeur les chemins fréquents puis les arbres libres fréquents qui se construisent à partir de chaque chemin fréquent et enfin les graphes connexes fréquents qui se construisent à partir de chaque arbre fréquent. Plus précisément, pour chaque chemin fréquent c trouvé, l'algorithme énumère les arbres fréquents dont le plus grand chemin (au sens d'un ordre lexicographique défini sur le codage canonique d'un chemin) est c. Pour chaque arbre t fréquent ainsi obtenu, `Gaston` énumère l'ensemble des graphes fréquents dont l'arbre recouvrant le plus grand (au sens d'un ordre lexicographique défini sur le codage canonique d'un arbre libre) est t. Dans la mesure où l'énumération des chemins et des arbres libres peut se faire en temps polynomial, `Gaston` limite le traitement exponentiel lié au problème de l'isomorphisme aux quelques graphes cycliques candidats à être fréquents. `Gaston` peut ou ne pas utiliser les listes d'occurrences pour calculer la fréquence du motif.

Les tests réalisés par les auteurs des algorithmes `gSpan` (Yan et Han, 2002), `FFSM` (Huan *et al.*, 2003) et `Gaston` (Nijssen et Kok, 2004) prouvent que ces algorithmes sont les plus efficaces. L'étude comparative réalisée indépendamment par Wörlein *et al.* (2005) et ciblée sur la fouille de graphes moléculaires confirme ce trio de tête avec une préférence pour `Gaston` (Nijssen et Kok, 2004).

2.4.3 La recherche de motifs de graphes optimaux ou contraints

L'énumération des graphes fréquents est rarement un but en soi. Les graphes fréquents sont généralement consommés par d'autres applications, en particulier pour faire de la classification supervisée de graphes à partir d'exemples positifs et négatifs (Deshpande *et al.*, 2003; Inokuchi et Kashima, 2003; Karwath et Raedt, 2004; Cheng *et al.*, 2007). L'idée est d'extraire un jeu restreint de motifs fréquents dans les exemples positifs et non fréquents dans les exemples négatifs et vice versa. L'identification des motifs les plus discriminants se fonde généralement sur des indicateurs statistiques calculés pour chaque motif fréquent. Une autre application est l'indexation intelligente de graphes dans une base de données (Yan *et al.*, 2004, 2005a; Chen *et al.*, 2007; Zeng *et al.*, 2008). L'idée est d'utiliser les motifs les plus représentatifs des données comme des clefs pour indexer le sous-ensemble des données qui contiennent les dits motifs.

En dehors de ces applications, la recherche de graphes fréquents reste un outil intéressant pour l'extraction de connaissances à partir de données. L'analyse visuelle par un expert des motifs fréquents pose toutefois un problème évident tant ces motifs fréquents sont nombreux. Afin d'en réduire le nombre et d'en augmenter la pertinence, différentes méthodes ont été proposées pour extraire uniquement tantôt des motifs de graphes fréquents satisfaisant certaines contraintes (section 2.4.3), tantôt des motifs qui maximisent une fonction de score (section 2.4.3).

La recherche de motifs sous contraintes

L'introduction de contraintes dans le processus de fouille de données permet de réduire le nombre de motifs produits et d'en augmenter potentiellement l'intérêt (cf section 2.1.4). Le tableau 2.24 énumère les principaux algorithmes de recherche de graphes sous contraintes. Tout comme pour les motifs d'attributs, l'idée générale de ces algorithmes est de prendre en compte certaines contraintes directement dans le processus de fouille pour ne pas avoir à éliminer a posteriori les nombreux motifs fréquents ne satisfaisant pas ces contraintes. Ainsi `SPIN` (Huan *et al.*, 2004) permet de chercher les motifs de graphes fréquents maximaux (au

Algorithme	Famille de motifs
`SPIN` (Huan *et al.*, 2004)	Graphes fréquents maximaux
`CloseSpan` (Yan et Han, 2002)	Graphes fermés fréquents
`CabGin` (Wang *et al.*, 2005a)	Contraintes monotones, anti-monotones et « succinctes » sur les graphes avec une
`CloseCut` and `Splat` (Yan *et al.*, 2005b)	Graphes relationnels fermés fréquents connectivité selon les arêtes supérieure à un seuil
`gPrune` (Zhu *et al.*, 2007)	Contraintes anti-monotones fortes et faibles sur les motifs et contraintes sur les données dépendantes ou indépendantes des motifs

FIG. 2.24 – Algorithmes de recherche de sous-graphes sous contraintes

sens de l'inclusion de sous-graphe isomorphe). L'idée de `SPIN` consiste à réaliser une partition de l'ensemble des graphes fréquents maximaux selon l'arbre recouvrant canonique auxquels ils sont rattachés. `SPIN` extrait alors l'ensemble des arbres fréquents et pour chacun d'entre eux produit l'ensemble des graphes fréquents maximaux qui lui sont rattachés.

`CloseSpan` (Yan et Han, 2002) permet d'extraire les graphes fermés fréquents. Un graphe g est fermé si tout graphe qui contient un sous-graphe isomorphe à g a une fréquence strictement inférieure à celle de g. Alors qu'il existe un seul motif fermé par classe d'équivalence dans le cas des motifs d'attributs, plusieurs graphes peuvent être fermés et décrire le même ensemble d'exemples (cf la non existence de treillis dans le cas des graphes à la section 2.3.1). De ce fait, la technique d'élagage valable dans le cas des motifs d'attributs qui consiste à ne plus faire croître un motif dès que celui-ci ne peut plus être fermé (suite à une extension qui conduit à un graphe plus grand et de même fréquence) ne s'applique pas systématiquement au cas des graphes. Yan et Han (2002) caractérisent les cas particuliers où l'élagage reste assurément valable et adaptent leur algorithme `gSpan` pour ne produire que les motifs fermés fréquents.

`CloseCut` and `Splat` (Yan *et al.*, 2005b) permettent d'intégrer une contrainte supplémentaire : les motifs recherchés doivent en plus d'être fréquents avoir une connectivité d'arêtes supérieure à un seuil fixé. La connectivité selon les arêtes d'un graphe est le nombre minimal d'arêtes à supprimer dans ce graphe pour que le graphe résultant ne soit plus connexe. La recherche de ces motifs de forte connectivité (i.e. proches d'être des graphes complets où toute paire de sommets sont reliés par une arête) peut présenter un intérêt dans le cadre de certaines applications, comme par exemple pour identifier des communautés dans des réseaux sociaux.

`CabGin` (Wang *et al.*, 2005a) propose une généralisation de `gSpan` pour intégrer un nombre arbitraire de contraintes monotones, anti-monotones et succinctes. Ce dernier type de contraintes permet de limiter les exemples à considérer dans le calcul de fréquence, notamment lors d'une phase de prétraitement. La fréquence est traitée comme une contrainte anti-monotone parmi d'autres, comme par exemple la contrainte qui veut que le nombre d'arêtes ou de sommets reste inférieur à une borne maximale ou que telle étiquette de sommet ou d'arête n'apparaisse pas dans le motif.

Enfin `gPrune` (Zhu *et al.*, 2007) s'attaque à des contraintes plus difficiles que les propriétés monotones ou anti-monotones comme les contraintes anti-monotones faibles (une telle contrainte ne peut être satisfaite par un motif que s'il existe un motif immédiatement inférieur dans l'ordre des motifs qui satisfait lui aussi cette contrainte) et les contraintes sur les données dépendantes ou indépendantes des motifs, qui permettent de limiter les données qui peuvent contenir les motifs.

La recherche de motifs de graphes optimaux

Les motifs sous contraintes sont généralement extraits par des algorithmes complets qui extraient la totalité des motifs satisfaisant les contraintes. En fonction du réglage de ces contraintes, les motifs obtenus peuvent toutefois s'avérer très nombreux, dans le cas de contraintes trop lâches, ou au contraire inexistants, dans le cas de contraintes trop sévères. Par ailleurs les contraintes n'intègrent pas la notion de préférence entre motifs : tous les motifs satisfaisant les contraintes sont considérés sur le même plan. En pratique il est souvent souhaitable de limiter le nombre de motifs à considérer et de pouvoir les trier par ordre décroissant d'un intérêt a priori. Une manière d'aborder ce problème est de se donner une fonction associant à chaque motif un score et de ne garder qu'un petit ensemble des k motifs aux scores les plus élevés. Cette approche dite des « top-k » motifs a d'abord été appliquée aux motifs d'attributs puis aux graphes. Cette approche est toutefois limitée dans la mesure où elle réduit un motif à un seul nombre quand bien même un motif aussi complexe qu'un graphe s'apprécie selon différentes dimensions parfois antinomiques. Il est bien sur possible de conjuguer ces différentes dimensions en pondérant dans la fonction de score différents facteurs éventuellement contradictoires mais il en résulte toujours une sélection de motifs fondée sur un score unique.

Une approche plus récente (Papadopoulos *et al.*, 2008) palliant ce problème consiste à extraire les motifs formant une « skyline », c'est à dire les motifs qui sont des maxima au sens de Pareto de n fonctions f_i de scores. Un score $(f_1(M), \ldots, f_n(M))$ d'un motif M est optimal au sens de Pareto s'il n'existe pas d'autre motif qui ait des scores au moins aussi grands que ceux de M pour toutes les fonctions f_i et au moins un score strictement supérieur pour une des fonctions f_i. L'avantage des optima de Pareto est de ne pas sacrifier une dimension (i.e. une fonction f_i) plutôt qu'une autre mais cet avantage se fait au prix d'un nombre de motifs optimaux plus grand que dans le cas d'une fonction de score scalaire. Au delà de leurs différences, les deux catégories d'algorithmes ont en commun le fait de chercher les motifs qui optimisent une ou plusieurs fonctions de score. Dans la mesure où la plupart des problèmes d'optimisation considérés sont des problèmes intrinsèquement difficiles, la plupart des algorithmes recourent à des heuristiques sans que cela soit toutefois systématique : Papadopoulos *et al.* (2008) et Yan *et al.* (2008) proposent tous deux des algorithmes exacts pour les problèmes qu'ils considèrent.

La table 2.25 présente un certain nombre de ces algorithmes de recherche d'optimum dans le cas des graphes. Les algorithmes développés dans cette thèse sont clairement rattachés à

Algorithme	Famille de motifs
`Subdue` (Cook et Holder, 1994; Jonyer *et al.*, 2001)	Graphes
`GBI` (Yoshida *et al.*, 1994)	
puis `CLIP` (Yoshida et Motoda, 1995)	
`GREW` (Kuramochi et Karypis, 2004b)	
`ORIGAMI` (Hasan *et al.*, 2007)	Graphes α-orthogonaux
`LEAP` (Yan *et al.*, 2008)	
`SkyGraph` (Papadopoulos *et al.*, 2008)	Sélection de type skyline

FIG. 2.25 – Algorithmes de recherche de sous-graphes optimaux

cette catégorie d'algorithmes. Cette orientation de nos recherches est principalement liée à une observation pragmatique : les algorithmes de recherche de graphes fréquents tout aussi efficaces soient-ils sont en soi d'un intérêt limité du point de vue des applications (cf chapitre

4) même si ces mêmes algorithmes peuvent être d'une grande utilité en tant qu'intermédiaire de calcul. Les efforts de recherche ont donc porté davantage sur la conception de méthodes de sélection de motifs fondées sur des heuristiques adaptées aux différents problèmes applicatifs considérés. Il est intéressant de remarquer que cette orientation semble partagée. À en juger en effet les publications ayant trait à la fouille de graphes parues en 2007 et 2008 (Hasan *et al.*, 2007; Yan *et al.*, 2008; Papadopoulos *et al.*, 2008), les algorithmes de recherche de motifs optimaux connaissent un véritable regain d'intérêt. Il s'agit en quelque sorte d'un retour aux sources puisque les premiers algorithmes de fouille de graphes comme `Subdue` (Cook et Holder, 1994, 2006) ou `GBI` (Yoshida *et al.*, 1994) étaient déjà des algorithmes de recherche heuristique. Il ne s'agit toutefois pas d'une régression dans la mesure où les algorithmes de recherche de motifs optimaux récents ont bénéficié des avancées réalisées par les algorithmes de recherche de graphes fréquents entre les années 2000 à 2005.

`Subdue` (cf Cook et Holder (1994) et chapitre 7 de Cook et Holder (2006)) est connu pour être un sinon le premier algorithme de fouille de graphes. Cet algorithme fait croître un graphe connexe g et calcule le nombre d'occurrences $\text{occ}(g)$ de ce graphe dans un ensemble $\mathcal{D}$ d'exemples. Il en déduit un score approximativement égal au produit $\text{occ}(g) \times |g|$ du nombre d'occurrences et de la taille de g. Ce score correspond à la taille d'espace mémoire économisée lorsque l'on remplace dans les données $\mathcal{D}$ chaque occurrence de g par un sommet particulier v_g et d'informations supplémentaires qui rendent la transformation réversible. `Subdue` cherche à l'aide d'un algorithme de recherche par faisceau[21] les top-k motifs qui maximisent leur score. Chaque occurrence du motif de plus grand score est ensuite contractée en un sommet et le processus est réitéré jusqu'à ce que le meilleur score obtenu devienne trop faible pour justifier d'un remplacement. Le principe sous-jacent à `Subdue` est la *Longueur de Description Minimale*[22] (Rissanen, 1978). qui considère qu'un modèle est d'autant plus descriptif de la réalité (ici les données) qu'il est capable de la compresser de manière réversible. `Subdue` a depuis connu de nombreux développements pour remplir diverses tâches de classification dont le clustering conceptuel (Jonyer *et al.*, 2001).

`GREW` (Kuramochi et Karypis, 2004b) est un algorithme qui permet d'extraire rapidement d'un graphe relationnel très grand (250000 sommets) des motifs fréquents de grande taille qui seraient inaccessibles par les méthodes classiques de recherche de sous-graphes fréquents. La fréquence d'un motif se définit comme le nombre maximal d'occurrences non recouvrantes (i.e. dont les sommets et arêtes sont disjoints) et son calcul est approximatif. Le principe de `GREW` consiste à fusionner des occurrences de motifs fréquents voisins pour en faire de nouveaux motifs candidats à être fréquents. Les fréquences des motifs candidats sont calculés et le processus est réitéré pour le nouvel ensemble de motifs fréquents. À chaque itération le nombre d'arêtes des motifs est en moyenne doublé par deux au lieu d'être augmenté d'une arête. `GREW` est ainsi capable de produire rapidement des motifs fréquents très grands qui seraient sinon inaccessibles avec des algorithmes de recherche de sous-graphes fréquents complets.

`ORIGAMI` (Hasan *et al.*, 2007) produit un ensemble restreint de motifs représentatifs et non redondants d'une base de graphes. `ORIGAMI` se fonde sur un indice de similarité entre graphes (Bunke et Shearer, 1998) qui calcule les sous-graphes communs maximaux à deux graphes pour qualifier la non-redondance entre motifs. Cet indice compris entre 0 et 1 permet de formaliser la notion de redondance et de représentativité des motifs : ainsi les éléments d'un ensemble $\mathcal{M}$ de motifs sont dits α-orthogonaux si l'indice de similarité entre toute paire de motifs de $\mathcal{M}$ est inférieur au seuil $\alpha \in [0; 1]$. Ce même ensemble est β-représentatif d'un

[21] *Beam search* en anglais.
[22] Minimal Description Length (MDL) en anglais

ensemble $\mathcal{G}$ de graphes si pour tout graphe $g \in \mathcal{G}$, il existe un motif M de $\mathcal{M}$ tel que l'indice de similarité entre g et M soit supérieur au seuil $\beta \in [0; 1]$. `ORIGAMI` détermine parmi les motifs fréquents maximaux, un sous-ensemble $\mathcal{M}$ de motifs α-orthogonaux dont le résidu – c'est à dire le nombre d'exemples dans les données pour lesquels $\mathcal{M}$ n'est pas β-représentatif – est minimal.

`LEAP` (Yan *et al.*, 2008) propose d'extraire les motifs « les plus significatifs » d'un ensemble de graphes de façon exacte. Ces motifs M sont ceux qui obtiennent un score maximal. Les fonctions de score considérées sont du type $f(\text{freq}^+(M), \text{freq}^-(M))$ calculées à partir des fréquences $\text{freq}^+(M)$ et $\text{freq}^-(M)$ de M dans un ensemble d'exemples respectivement positifs et négatifs et doivent fournir les scores les plus élevés pour des motifs discriminants (i.e. très fréquents dans un ensemble d'exemples et très peu dans l'autre). Yan *et al.* (2008) proposent de résoudre le problème non pas par une approche « branch and bound » classique qu'il juge inefficace mais par une recherche « par bonds » [23] tirant parti d'un encadrement du score d'un motif par le score d'un sous-graphe et d'un super-graphe.

Enfin `SkyGraph` (Papadopoulos *et al.*, 2008) propose d'extraire la skyline des motifs maximaux au sens de Pareto (cf plus haut) selon deux dimensions que sont la taille du motif et la connectivité selon les arêtes (cf algorithme `CloseCut` plus haut). L'algorithme de complexité polynomiale prend en entrée un grand graphe relationnel unique et le décompose successivement à l'aide d'un algorithme de coupure minimale. Il obtient de cette façon tous les motifs appartenant à la skyline sans exception.

En conclusion, de toutes les approches étudiées pour fouiller les motifs structuraux contenus dans les BdR, celle qui semble la plus adaptée est la fouille de graphes qui offre la possibilité d'extraire les sous-graphes inclus dans les BdR tout en ayant fait la preuve de son efficacité, à travers des algorithmes comme `gSpan` (Yan et Han, 2002) ou `Gaston` (Nijssen et Kok, 2004). La recherche systématique de tous les sous-graphes fréquents ne répond toutefois pas aux besoins spécifiques de chaque application, pour lesquels des algorithmes de recherche plus sélective sont nécessaires. Les principales contributions présentées dans ce mémoire s'inscrivent dans cette logique qui consiste à proposer des méthodes de fouille de graphes répondant à des problèmes originaux. En l'occurrence ces problèmes trouvent leur inspiration dans des questions relatives à la synthèse organique, dont la problématique générale est présentée au chapitre suivant.

[23] *Leap search* en anglais.

Chapitre 3

Synthèse organique et chémoinformatique

Sommaire

Ce chapitre répond principalement à deux objectifs. Le premier, qui fait l'objet de la section 3.1, est d'introduire les notions de chimie organique et plus précisément de synthèse organique qui sont essentielles à la compréhension des chapitres suivants. Le second objectif est de résumer, au cours des deux sections 3.2 et 3.3, les travaux en chémoinformatique qui sont en relation avec les problèmes traités dans le mémoire. La section 3.2 propose d'abord une introduction à la chémoinformatique et ses applications à la synthèse organique. Elle développe plus particulièrement les systèmes d'assistance à la rétrosynthèse, auxquels fait référence le chapitre 7, ainsi que les bases de données de réactions. La section 3.3 étudie plus particulièrement les contributions de la fouille de données à la chémoinformatique et plus précisément encore, l'extraction de connaissances à partir des BdR.

3.1 Introduction à la synthèse organique

Cette section est destinée aux lecteurs non-chimistes à qui les notions spécifiques de la synthèse organique ne sont pas familières. Son contenu constitue une présentation très succincte – et donc très simplifiée – des notions élémentaires de chimie organique récurrentes dans ce

mémoire, que sont les molécules décrites à la section 3.1.1, puis les réactions chimiques et la synthèse organique introduits dans la section 3.1.2. La section 3.1.3 décrit ensuite les deux problèmes complémentaires posés par la synthèse organique : la connaissance des méthodes de synthèse d'une part et l'application de ces méthodes à de nouveaux problèmes de synthèse d'autre part. Cette distinction est importante dans la mesure où les travaux réalisés et présentés dans ce mémoire contribuent au traitement de l'un ou l'autre problème. Enfin la section 3.1.4 introduit la rétrosynthèse en tant que méthode générale de résolution de problèmes de synthèse organique.

3.1.1 Les molécules et les graphes moléculaires

La chimie est la science qui étudie la structure (niveau microscopique), les propriétés physiques et les propriétés chimiques (niveau macroscopique) des substances. Une grande part des substances étudiées en chimie, en particulier des substances organiques, sont faites de molécules que l'on se représente de manière abstraite, comme des assemblages d'atomes dont la cohésion est assurée par la présence de liaisons inter-atomiques. En simplifiant, tout *atome* est constitué d'un agrégat de neutrons et de protons appelé *noyau* autour duquel gravitent différentes *couches d'électrons*. Le nombre de protons présents dans le noyau détermine à lui seul de nombreuses propriétés chimiques. Cette découverte a amené les chimistes à regrouper les atomes ayant le même nombre de protons en familles appelées *éléments chimiques*. Le *carbone*, l'*hydrogène*, l'*oxygène* sont des exemples d'éléments chimiques omniprésents dans les substances organiques. La plupart des atomes isolés sont toutefois instables et se lient sous l'effet de forces spécifiques, comme représentés schématiquement sur la figure 3.1. L'effet

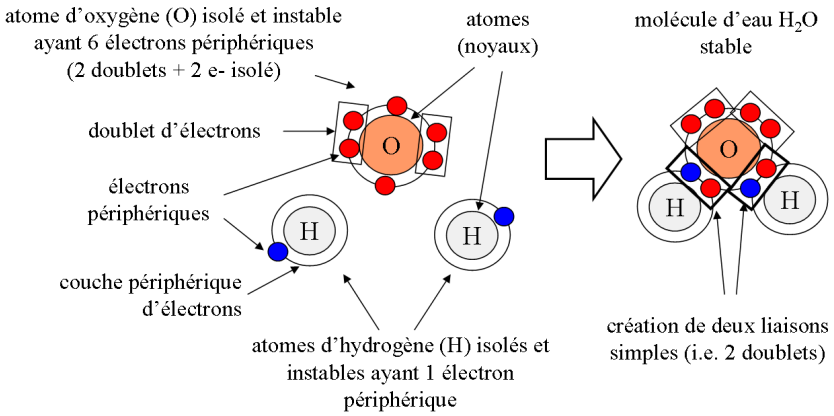


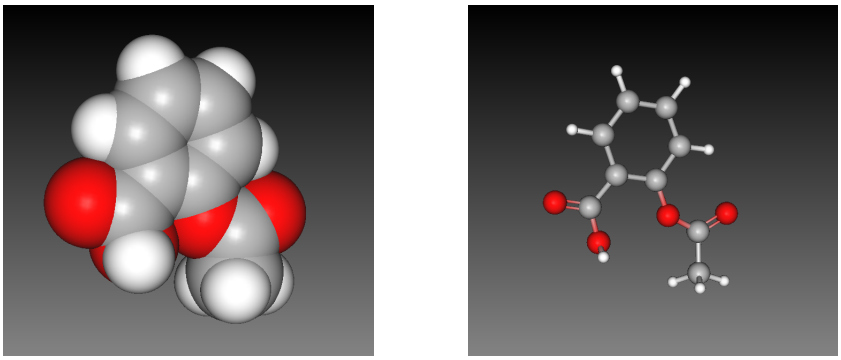
FIG. 3.1 – Une représentation abstraite des atomes constitués d'un noyau et d'électrons périphériques ainsi que des liaisons constituées de doublets d'électrons.

de ces forces est la formation d'une *liaison* dite de covalence dans laquelle les deux atomes liés mettent en commun un nombre égal d'électrons périphériques, formant ainsi un certain nombre de paires, ou *doublets d'électrons*. Un assemblage stable d'atomes ainsi liés forme une *molécule*. La mécanique quantique prédit différentes distributions spatiales de ces doublets d'électrons. Cette distribution détermine le *type de la liaison* que les chimistes qualifient de *simple* (i.e. un doublet commun aux deux atomes), *double* (i.e. deux doublets), *triple* (i.e. trois

doublets) ou *aromatique* (i.e. doublets délocalisés sur plus de deux atomes). Une molécule est donc une cohésion stable d'atomes solidaires reliés de proche en proche par des liaisons.

Il est important de souligner que le schéma de la figure 3.1 et tous ceux qui vont suivre ne sont que des représentations abstraites, qu'il faut bien dissocier de toute représentation de la réalité matérielle à l'échelle microscopique, si tant est qu'une telle représentation ait un sens et soit accessible. Le rôle joué par ces différents modes et langages de représentation est essentiel pour comprendre les propriétés des molécules. En particulier, Grosholz et Hoffmann (2000) distinguent deux familles de langages en chimie : d'une part, les *langages symboliques* permettent comme dans toute autre science, de nommer les différents objets et concepts abstraits spécifiques pour pouvoir ensuite les manipuler ; d'autre part, les *langages « iconiques »* jouent un rôle peut-être encore plus important pour représenter et raisonner sur les molécules à l'aide de diagrammes. Selon Grosholz et Hoffmann (2000), les représentations iconiques ont l'avantage d'exprimer énormément d'information implicite en plus des quelques « coups de crayons » explicites, quand les langages symboliques n'expriment essentiellement que ce qui est explicite. À cela s'ajoute une troisième famille de langages qui correspond aux représentations utilisées par la chémoinformatique pour manipuler les objets de la chimie à l'aide d'algorithmes. Ces représentations résultent souvent de formalisations simplifiées des représentations symboliques ou iconiques utilisées par les chimistes. Ainsi, la chémoinformatique adopte comme on va le voir, certaines représentations iconiques des molécules et des réactions grâce au concept de *graphe moléculaire*.

Les chimistes ont développé plusieurs modes de représentation iconique qui mettent en valeur différentes facettes des molécules selon différents niveaux de précision. Une molécule peut ainsi se modéliser par un édifice géométrique de sphères représentant les espaces qu'occupent en moyenne les atomes. Ce mode de représentation se matérialise sous la forme de maquettes manipulables par les chimistes ou bien par des images « 3D » de réalité virtuelle telles que celle de la figure 3.2(a). On peut y observer une convention informatique selon laquelle la couleur des différents atomes représente leur type ou élément chimique : les atomes gris sont généralement ceux de carbone, les blancs ceux d'hydrogène et les rouges ceux d'oxygène. Si



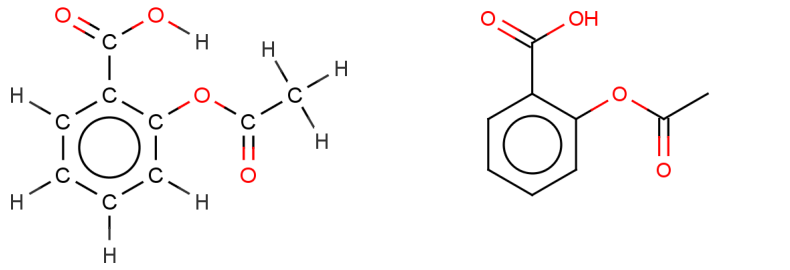
(a) Représentation spatiale (sphères de Van der Waals)

(b) Représentation éclatée

FIG. 3.2 – Représentation géométrique de l'aspirine.

maintenant on matérialise artificiellement les liaisons par des « tiges » dans une *représentation dite éclatée* telle que celle de la figure 3.2(b), on fait apparaître une autre représentation appelée *formule développée* et représentée sur la la figure 3.3(a). Chaque atome y est représenté par le symbole de son élément chimique (C pour carbone, H pour hydrogène, O pour oxygène,

N pour azote ...) et chaque liaison est représentée par des traits : un trait simple, double ou triple représente une liaison simple, double ou triple alors qu'un cercle représente un cycle de liaisons aromatiques. En pratique les chimistes utilisent une représentation conventionnelle allégée mais équivalente de la formule développée, appelée formule structurale, dans laquelle la plupart des atomes d'hydrogène et le symbole du carbone disparaissent. Du fait du principe de valence expliqué ultérieurement au chapitre 4, les formules développée et structurale d'une même molécule sont équivalentes : la formule développée de la figure 3.3(a) peut être reconstruite à partir de la formule structurale de la figure 3.3(b). Cette représentation structurale



(a) Formule développée (b) Formule structurale réduite

FIG. 3.3 – Diagramme de l'aspirine.

plane est certes plus pauvre que la représentation géométrique de la figure 3.2(a) puisque sont perdues toutes les informations relatives aux distances inter-atomiques et aux angles que forment les liaisons entre elles. Toutefois la perte de l'information géométrique dans les formules structurales doit être relativisée : d'abord, la vue géométrique est elle-même une approximation représentant la molécule dans son état d'énergie minimale (i.e. puits de l'énergie potentielle liée aux forces de répulsion et de cohésion entre atomes). En pratique la géométrie d'une molécule n'est pas rigide, se déforme sous l'effet des chocs entre molécules et admet un certain nombre de degrés de liberté, notamment de rotation libre autour des liaisons simples. Ensuite, l'essentiel de la géométrie de la molécule (dans son état minimal d'énergie) peut être reconstruite à partir de sa formule structurale. Dans le cas contraire, la formule structurale peut être enrichie de l'information géométrique (les chimistes disent stéréochimique) manquante, nécessaire à la reconstruction de la représentation géométrique. Enfin et surtout, beaucoup de lois chimiques peuvent s'expliquer à partir de considérations sur les formules structurales sans avoir à recourir à d'autres informations de nature géométrique. En pratique un grand nombre de formules structurales présentes dans la littérature ne font apparaître que des atomes et des liaisons. Il est alors possible de transformer ces formules structurales en des graphes, appelés *graphes moléculaires*, dans lesquels les sommets et les arêtes représentent respectivement les atomes et les liaisons étiquetés par leur type (i.e. l'élément chimique pour les atomes). Les graphes moléculaires ne sont pas véritablement un mode de représentation utilisé par les chimistes mais ont été introduits pour permettre aux algorithmes de chémoinformatique de manipuler plus facilement les structures de molécules.

Si le modèle des schémas structuraux et donc des graphes moléculaires est souvent suffisant pour expliquer les phénomènes chimiques, il est aussi nécessaire : la structure topologique des molécules que véhiculent les graphes moléculaires, explique à elle seule et en grande partie des propriétés physico-chimiques comme la solubilité, les effets inductifs ou mésomères, la résonance et plus généralement la réactivité des molécules. De ce fait, les schémas structuraux

(et donc les graphes moléculaires) suffisent dans bien des cas aux chimistes pour exprimer leurs connaissances de façon à la fois simple et précise et de raisonner avec celles-ci. C'est aussi pour ces raisons que dans la suite du mémoire, les molécules sont modélisées exclusivement par leurs graphes moléculaires du type de celui de la figure 3.3(a).

3.1.2 La synthèse organique et les réactions chimiques

Les molécules sont omniprésentes dans la nature et pour cette raison sont très étudiées dans les industries pharmaceutiques ou agroalimentaires mais aussi dans d'autres secteurs clef de l'économie comme celui de l'énergie ou des matériaux synthétiques. À ce titre, la recherche de nouvelles molécules aux propriétés innovantes constitue un moteur important du développement de l'économie moderne. Quel que soit le domaine d'application considéré, les molécules posent aux chercheurs et aux industriels le problème récurrent de leur synthèse : étant donnée une nouvelle molécule cible, comment établir le processus de transformation chimique qui permettra de disposer de cette molécule en quantité voulue ? Dans bien des cas en effet l'intérêt suscité par une molécule précède sa synthèse en laboratoire, soit parce qu'il s'agit d'une molécule naturelle présentant des propriétés intéressantes, soit parce que cette molécule n'existe pas dans la nature mais que les chimistes lui prédisent des propriétés intéressantes. Même dans le cas d'une molécule qu'on sait préparer en laboratoire, le problème de sa synthèse industrielle présente des exigences de rendement et de coût tellement difficiles à atteindre qu'il devient nécessaire de repenser entièrement le processus de synthèse.

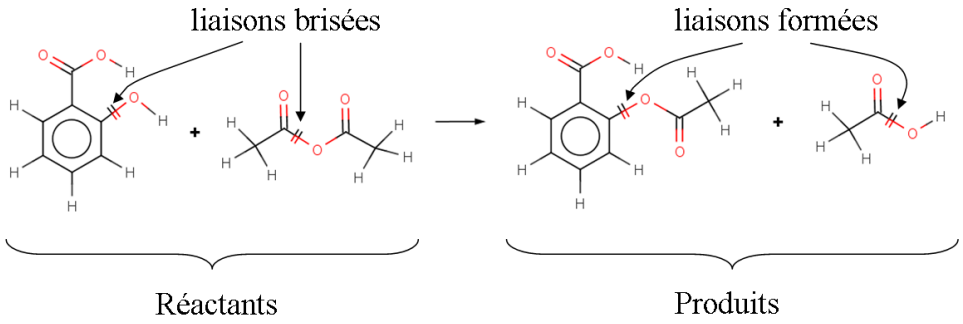
Le non-chimiste pourrait penser que la synthèse d'une molécule s'apparente à un jeu de construction : il suffit d'assembler physiquement les atomes les uns aux autres jusqu'à obtenir l'assemblage voulu[24]. Cette méthode s'avère toutefois irréalisable et la synthèse de molécules est un problème autrement plus compliqué. En effet, les transformations chimiques qui permettent de changer l'agencement des atomes et des liaisons au sein des molécules sont contraintes par des lois physiques très complexes. Du fait de ces contraintes, seul un faible nombre de toutes les transformations imaginables s'avère réalisable. Ces transformations sont appelées *réactions chimiques* ou simplement réactions. Le problème de la synthèse d'une molécule cible revient alors à trouver un enchaînement de réactions chimiques qui transforme un ensemble de *produits de départ*, c'est-à-dire de molécules disponibles (notamment dans la matière fossile telle que le pétrole, charbon ou gaz naturel) en la molécule cible.

Ainsi une réaction chimique est un processus de transformation qui modifie la structure d'une ou plusieurs molécules. Grâce à l'énergie apportée par le choc entre plusieurs, généralement deux, molécules, les électrons composant certaines liaisons entre atomes se réagencent différemment pour aboutir à de nouvelles molécules plus stables. Ce déplacement d'électrons peut provoquer différents effets : certaines liaisons se rompent, d'autres se créent, d'autres enfin voient leur type modifié, passant par exemple du type double au type aromatique[25]. Tout comme pour les molécules, les chimistes disposent de plusieurs modes de représentation des réactions. Dans la mesure où les molécules sont le plus souvent représentées par leurs formules structurales, une réaction se représente généralement par la juxtaposition des formules structurales des molécules de départs, ou *réactants* représentées à gauche de celles des molécules

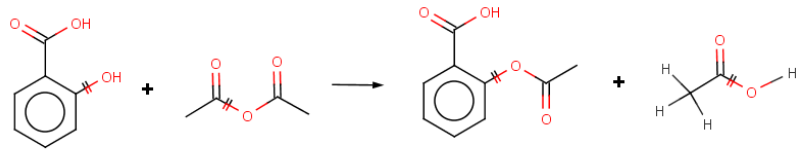
[24]Cette remarque, que les chimistes trouveront inepte, vise à dissiper une fausse idée de la synthèse, somme toute assez naturelle, que plusieurs informaticiens rencontrés lors de conférences ont pu avoir.

[25]Ces déplacements d'électrons peuvent aussi créer un excédent ou une lacune d'électrons sur un atome, créant ainsi une charge électrique tantôt négative tantôt positive sur l'atome, ou encore dissocier des doublets d'électrons, créant alors des radicaux libres instables. Ces deux effets ne modifiant pas fondamentalement la structure de la molécule, seront passés sous silence dans la mesure du possible.

obtenues, ou *produits*, les deux termes étant séparés par une flèche. L'ensemble résultant forme une *équation chimique* qui peut être donnée sous forme développée ou réduite selon que les formules structurales des réactants et des produits sont elles-mêmes spécifiées sous forme développée ou non. La figure 3.4 représente les équations développées et réduites d'une réaction chimique réalisant la synthèse de l'aspirine, dont on reconnaît le graphe moléculaire en tant que premier produit de la réaction. Les équations chimiques sont tout comme les formules



(a) Équation chimique développée



(b) Équation chimique réduite

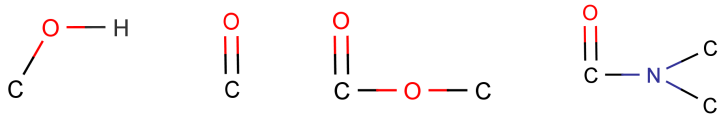
FIG. 3.4 – Réaction de synthèse de l'aspirine.

structurales, des modes de représentation non standardisés qui admettent de nombreuses variantes et sophistications. Dans la suite une équation chimique fait référence à la représentation « minimaliste » mentionnée précédemment, qui correspond aussi à la représentation usuelle des réactions en chémoinformatique. À ce sujet, les représentations d'équations chimiques figurant dans ce mémoire sont issues d'outils de visualisation graphique qui adoptent certaines conventions particulières. Les liaisons brisées par la réaction dans le terme gauche ainsi que les liaisons formées dans le terme droit, sont notamment représentées par des liaisons « doublement barrées ». Cette convention permet d'observer que chacun des deux réactants se brise en deux fragments qui se recombinent deux à deux pour donner deux produits, le premier étant le produit principal (l'aspirine), le second étant un produit secondaire (l'acide acétique).

Il est important de comprendre qu'une réaction chimique est un phénomène beaucoup plus complexe que ne le le laisse transparaître son équation chimique. Une réaction chimique se définit comme le phénomène dynamique qui commence à l'instant où les réactants entrent en collision et se termine au moment où les produits sont formés. Entre ces deux instants, la réactions passe par des états de transition qui ne sont pas représentés dans l'équation. Par ailleurs les réactions peuvent comporter plusieurs étapes intermédiaires, les produits formés lors d'une étape réagissant spontanément dans une étape suivante. Certaines molécules peuvent aussi réagir selon plusieurs réactions concurrentes, auquel cas il n'y a plus un seul

ensemble de produits mais plusieurs, alors pondérés par leur rendement respectif[26]. Enfin la mise en œuvre d'une réaction nécessite un environnement particulier en plus de la mise en présence des réactants : une réaction exige des conditions particulières de température, de lumière et de pression, ou encore la présence d'un certain type de solvants ou de catalyseurs. Là encore, ces aspects complexes et très éloignés de la problématique abordée par la fouille de graphes ont été volontairement occultés.

Enfin la *fonctionnalité* des molécules est un concept particulièrement important pour comprendre les réactions et qui mérite d'être mentionné ici puisque les travaux de S. Berasaluce (Berasaluce, 2002) ainsi que le chapitre 5 y font référence. Un *groupe fonctionnel* est un groupe caractéristique d'atomes qui induit des propriétés chimiques particulières aux molécules qui le contiennent. La figure 3.5 donne des exemples de groupes fonctionnels très courants. Les groupes fonctionnels sont importants dans la mesure où leur présence dans une



(a) Alcool (b) Carbonyle (c) Ester (d) Amide

FIG. 3.5 – Exemples de groupes fonctionnels.

molécule détermine en grande partie sa réactivité, c'est-à-dire l'ensemble des réactions que subit cette molécule – en tant que réactant – sous certaines conditions. Pour cette raison, les groupes fonctionnels servent souvent de clés d'indexation pour établir une classification des molécules et des réactions. Les groupes fonctionnels ne disposent toutefois pas de représentation formelle utilisable en chémoinformatique, si ce n'est sous forme de définition extensive qui consisterait à énumérer les groupes fonctionnels les plus importants sous forme de graphes. Pour cette raison, les travaux présentés dans Berasaluce (2002) introduisent la notion de *fonction chimique*, définie comme tout sous-graphe connexe maximal de liaisons multiples ou hétéroatomiques (i.e. de liaisons autre que $C-C$ ou $C-H$) inclus dans un graphe moléculaire. L'introduction des notions fondamentales relatives aux molécules et réactions permet maintenant d'aborder les problèmes de synthèse organique.

3.1.3 Les questions posées par la synthèse organique

Pour pouvoir tester expérimentalement une réaction, encore faut-il disposer de ses réactants. Dans le cas contraire, il faut trouver d'autres réactions qui produisent les réactants manquants à partir de molécules plus petites ou du moins plus simples jusqu'à obtenir par itération un plan de synthèse complet partant de molécules disponibles. La nature combinatoire du problème posé par la synthèse des molécules explique pourquoi ce problème exige de

[26]Le rendement d'une réaction $R \rightarrow P$ se définit dans des conditions données, comme la fraction du nombre de molécules P réellement produites au bout d'un temps indéfiniment long, sur le nombre total des mêmes molécules P que l'on pourrait obtenir si tous les réactants R étaient consommés par cette réaction. Un rendement peut être inférieur à 100 % en raison de réactions concurrentes $R \rightarrow P'$ aboutissant à des produits P' différents, voire à une réaction inverse $P \rightarrow R$ conduisant à un équilibre entre concentrations des réactants R et des produits P.

la part des chimistes plus encore qu'un savoir théorique, un savoir-faire : il n'existe pas en effet de théorie capable de produire de façon systématique les plans de synthèse de toutes les molécules.

Comme tout savoir-faire, l'art de la synthèse se fonde sur deux activités complémentaires : l'acquisition de nouvelles connaissances grâce à l'expérience passée et l'application de cette connaissance aux expériences futures. Dans le cas spécifique de la synthèse, ces connaissances se matérialisent par un ensemble de méthodes dites de synthèse : en simplifiant, une méthode de synthèse désigne une classe de réactions qui permet d'atteindre un objectif stratégique dans le domaine de la synthèse organique. En pratique ces méthodes de synthèse se décrivent par un ou plusieurs schémas de réaction génériques (au sens de réutilisables), observés par de nombreuses réactions réalisées expérimentalement, et qu'on peut espérer pouvoir appliquer à d'autres problèmes de synthèse. L'acquisition et la réutilisation des connaissances se traduisent donc pour les experts de la synthèse organique en deux tâches distinctes mais indissociables :

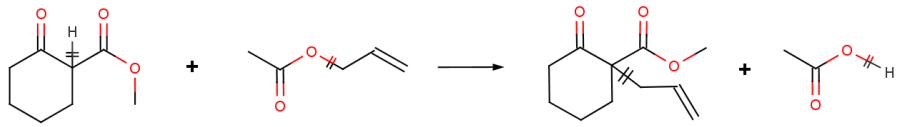
1. D'une part la découverte et la mise au point de nouvelles méthodes de synthèse ou à défaut, l'amélioration des connaissances relatives à des méthodes de synthèse déjà connues.
2. D'autre part l'identification à partir d'une molécule cible, de la ou des méthodes de synthèses qui permettent de produire la cible à partir de molécules plus simples à synthétiser[27].

Certains chimistes qualifient respectivement ces problématiques de *méthodologie de synthèse* et de *synthèse ciblée* (voir par exemple Gien (1998, page 86)). Les méthodes développées dans ce mémoire trouvant des applications dans l'une ou l'autre des problématiques, celles-ci sont développées dans les deux sections suivantes.

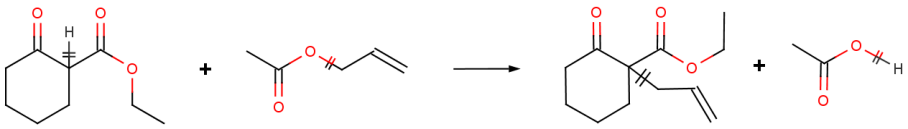
L'identification de méthodes de synthèse

Les chimistes ont de longue date observé que les réactions chimiques pouvaient être regroupées en familles ou classes de réactions selon qu'elles partagent le même schéma de réaction générique. Reproduisons cette découverte sur un exemple et considérons l'exemple de la réaction donnée sur la figure 3.6(a) réalisée dans certaines conditions omises ici. Si on modifie légèrement le premier réactant en lui ajoutant un atome supplémentaire de carbone (cf figure 3.6(b)) et qu'on se place dans les conditions équivalentes, on observe expérimentalement que le nouveau réactant réagit selon le même schéma de réaction que celui observé précédemment (cf figure 3.6(a)). En modifiant successivement d'autres fragments des réactants, on aboutit à la réaction de la figure 3.6(c) qui réagit toujours selon le même schéma alors qu'elle est foncièrement différente de la réaction initiale. En particulier un atome d'iode I joue le même rôle que l'atome d'oxygène O du second réactant. Si toutefois cet atome d'iode est lié à un atome de carbone tertiaire, c'est-à-dire qui est lui même lié à trois autres atomes de carbone (cf figure 3.6(d)), aucune réaction suivant le schéma de transformation des réactions précédentes ne se produit. On obtient ainsi de proche en proche un ensemble d'exemples et de contre-exemples de la famille de réactions que l'on tente de caractériser. Ce faisant, on peut établir précisément la structure minimale dans le terme gauche des réactants qui est présente dans les équations des exemples et absente dans celles des contre-exemples. Si on lui rajoute

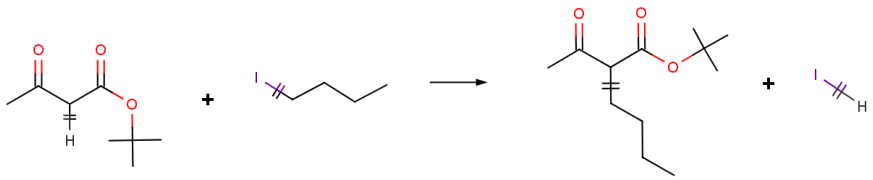
[27]La définition de la simplicité de synthèse d'une molécule dépend des objectifs de la synthèse elle-même. En général la synthèse est d'autant plus simple que le nombre ou le coût des étapes nécessaires à la fabrication de la dite molécule est faible.



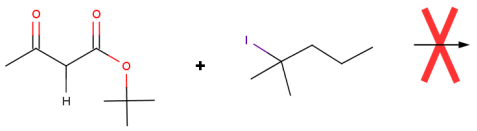
(a) Exemple 1



(b) Exemple 2



(c) Exemple 3



(d) Contre-exemple

FIG. 3.6 – Exemple d'une famille de réactions (synthèses à partir d'esters acétoacétiques).

comme terme droit le terme gauche transformé, on obtient le *schéma de réaction générique*, caractéristique de la famille de réactions considérée. Ce schéma générique est représenté sur la figure 3.7 : le symbole X désigne ici un atome d'oxygène ou un atome halogène de type

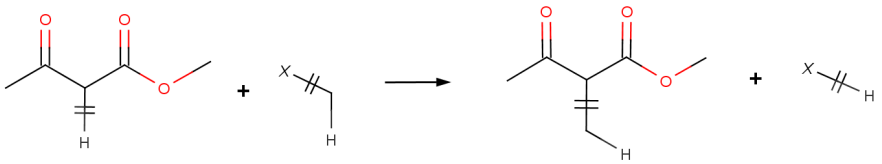
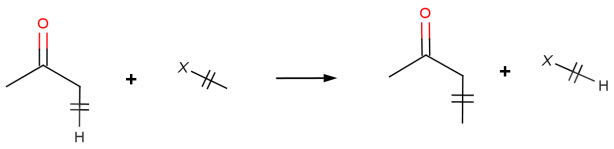
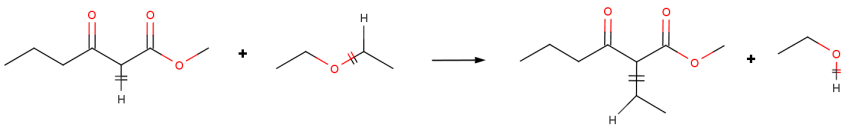


FIG. 3.7 – Schéma de réaction générique, caractéristique d'une méthode de synthèse

chlore (Cl), brome (Br) ou iode (I), représentant ainsi le fait que les schémas de réactions associés aux différentes valeurs que peut prendre l'atome X ne sont pas fondamentalement différents du point de vue des chimistes. La présence explicite d'un atome d'hydrogène (H) dans le second réactant $X - C - H$ indique que le carbone lié à cet atome d'hydrogène ne peut pas être tertiaire. Du point de vue de la terminologie, les expressions synonymes de « schéma de réaction » ou « schéma réactionnel » utilisées très fréquemment dans le reste du mémoire ne font pas nécessairement référence à un *schéma de réaction générique*, caractéristique d'une méthode de synthèse, mais doivent être comprises dans leur acception la plus large : un *schéma de réaction* ou *schéma réactionnel* est compris ici comme une structure syntaxique constituée de la juxtaposition d'un terme de départ et d'un terme d'arrivée représentant chacun des atomes reliés par des liaisons. Ainsi, si à chaque méthode de synthèse est associée à au moins un schéma de réaction (qui est alors générique), la réciproque n'est pas vraie : un schéma de réaction contenu dans l'équation d'une réaction peut être tantôt trop général, comme celui de la figure 3.8(a), tantôt inutilement trop spécifique comme celui de la figure 3.8(b), pour pouvoir représenter le schéma de la réaction générique sous-jacente déjà précisé sur la figure 3.7.



(a) Schéma trop général



(b) Schéma inutilement trop spécifique

FIG. 3.8 – Exemples de schémas de réactions non génériques

Cette démarche de généralisation satisfaisant exemples et contre-exemples n'est pas sans rappeler les méthodes d'apprentissage à partir d'exemples (Langley, 1996; Mitchell, 1997) en

particulier celui de l'espace des versions utilisé dans le domaine de l'apprentissage symbolique (cf chap. 2 de Mitchell (1997)). De manière analogue, le fait de pouvoir représenter par un schéma de réaction (générique ou non) l'ensemble (ou famille) de réactions qui ont en commun ce schéma fait référence à la classification conceptuelle (Michalski et Stepp, 1983; Gennari *et al.*, 1989) : les familles de réactions sont les équivalents de *concepts*, représentés à la fois par une description, ou *intension*, exprimée dans un langage de représentation formel, qui est ici celui des schémas de réactions, et à la fois par un ensemble d'instances, ou *extension*, qui sont ici les réactions dont les équations comprennent le schéma de l'intension. Certains concepts peuvent alors être vus comme des spécialisations d'autres concepts selon une relation de subsomption qui se traduit ici par la relation d'inclusion entre schémas de réactions. Cette analogie est reprise et définie formellement au chapitre 4. C'est aussi en raison de cette analogie que les modèles de représentation des connaissances (Napoli, 1992) ont trouvé en la synthèse organique un terrain d'expérimentation idéal, à travers des systèmes comme ReSyn développé à la section 3.2.2. En résumé, les schémas réactionnels sont pour les chimistes un langage essentiel de représentation de leurs connaissances en synthèse organique, capable de représenter les réactions, les classer en familles et d'organiser ces familles selon différentes classifications. En particulier la connaissance des schémas génériques caractéristiques des méthodes de synthèse permet aux chimistes de les réutiliser comme « patrons de conception » pour résoudre de nouveaux problèmes de synthèse.

La réutilisation des méthodes de synthèse

L'établissement de méthodes de synthèse n'a d'intérêt que si ces méthodes sont utilisées pour résoudre des problèmes de synthèse ciblée. Étant donnée une molécule cible que l'on cherche à synthétiser, la solution la plus simple consiste à trouver une réaction qui produise la cible à partir de produits de départ disponibles. Si toutefois une telle réaction ne peut pas être trouvée, il devient nécessaire d'établir un *plan de synthèse* qui transforme un ensemble de produits de départ en la cible selon plusieurs réactions chimiques successives. Un tel plan peut se représenter sous la forme d'un *arbre de synthèse*, illustré par la figure 3.9. Les nœuds d'un tel arbre représentent des molécules : la racine représente la cible, les feuilles sont les produits de départ et les nœuds intermédiaires représentent à la fois les produits principaux de réactions qui transforment les molécules des nœuds fils et à la fois les réactants de la réaction qui produit la molécule du nœud parent. Lorsque le plan de synthèse a pour produits de départ de petites molécules généralement issues de la pétro-chimie, on parle alors de synthèse totale. Lorsque le plan recourt à des molécules complexes généralement issues du monde végétal et extraites par un procédé biologique, on parle de synthèse partielle. Les plans de synthèse peuvent par ailleurs être de structure plutôt linéaire ou convergente. Dans le cas linéaire, on part d'un produit de départ, puis on cherche les réactions successives qui ajoutent à la molécule courante des fragments supplémentaires jusqu'à obtenir la cible voulue. L'arbre de synthèse résultant est alors profond et étroit. Les synthèses linéaires présentent l'inconvénient d'un faible rendement et donc d'un coût élevé, puisque le rendement global résulte du produit des rendements des nombreuses réactions intermédiaires. Les synthèses convergentes au contraire assemblent des molécules de taille similaire pour produire une molécule deux fois plus grande. L'arbre de synthèse est alors un arbre « bien balancé », peu profond et de rendement élevé.

Les réactions qui composent un plan de synthèse ne sont pas identifiées directement, mais comme des instances de méthodes de synthèse qui semblent applicables dans le contexte du problème considéré. Par ailleurs un plan de synthèse peut se concevoir soit dans le sens synthétique, en partant de produits de départ potentiels (i.e. des feuilles) pour tenter de converger

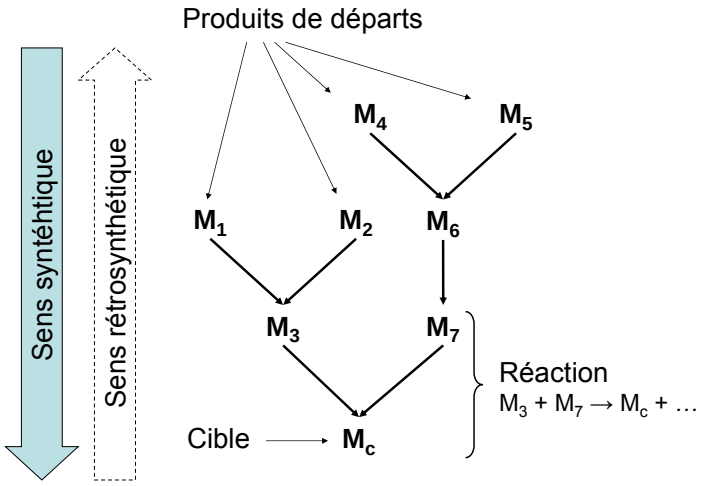


FIG. 3.9 – Plan de synthèse

vers la synthèse de la cible (i.e la racine), soit dans le sens rétrosynthétique, en cherchant une réaction qui aboutisse à la cible et en réitérant le processus sur les réactants de la réaction, soit encore, en mélangeant ces deux méthodes qui ne sont pas sans rappeler le principe de chaînages avant et arrière en intelligence artificielle. Les choix et donc les erreurs faites à un certain stade l'analyse se répercutent sur les choix ultérieurs. On le voit bien, la conception d'un plan de synthèse est une tâche extrêmement complexe et pourtant indispensable, comme l'a dit R. B. Woodward (récipiendaire du prix Nobel de 1965) : « synthesis must always be carried out by a plan ». Ce dernier a ainsi montré la nécessité et l'intérêt de procéder à un raisonnement de la plus grande minutie, pour pouvoir réussir des plans de synthèse ambitieux. Cette nécessité de prendre en compte toute la complexité du problème ainsi que la demande croissante de molécules cibles toujours plus complexes ont fait apparaître le besoin de se doter d'une méthodologie systématique adaptée au problème de la synthèse des molécules. Cette méthodologie devait ainsi être capable de clarifier et organiser les savoir-faires accumulés en synthèse organique, pour guider et donc accélérer la conception du plan de synthèse de molécules complexes.

3.1.4 La rétrosynthèse

Elias James Corey, récipiendaire du prix Nobel de chimie en 1990, a le premier formalisé une telle méthodologie qu'il a baptisée *rétrosynthèse* (Corey, 1971; Corey et Cheng, 1995). Cette méthodologie prévaut encore aujourd'hui dès qu'il s'agit de concevoir le plan de synthèse d'une *molécule cible* complexe. La rétrosynthèse est une méthode de résolution analytique et itérative qui opte pour un raisonnement en chaînage arrière partant de la cible pour remonter jusqu'aux produits de départ selon le *sens rétrosynthétique*, par opposition au *sens synthétique*, c'est-à-dire au déroulement naturel des réactions. Plus précisément la rétrosynthèse consiste à déduire de la structure de la molécule cible une réaction qui soit en mesure de la synthétiser, et ce à partir de réactants appelés *précurseurs* dont la synthèse est jugée être plus facile que celle de la cible. Si ces précurseurs ne sont pas des produits de départ disponibles, la procédure est alors réitérée, chaque précurseur devenant la cible courante. Ce

processus récursif s'interrompt lorsque tous les précurseurs ainsi obtenus sont des produits de départ, aboutissant ainsi à un plan de synthèse complet de la cible initiale. La figure 3.10 illustre la rétrosynthèse réalisée par Corey en 1957 de la molécule pourtant difficile à synthétiser du longifolène (car géométriquement complexe et peu fonctionnalisée). La double flèche

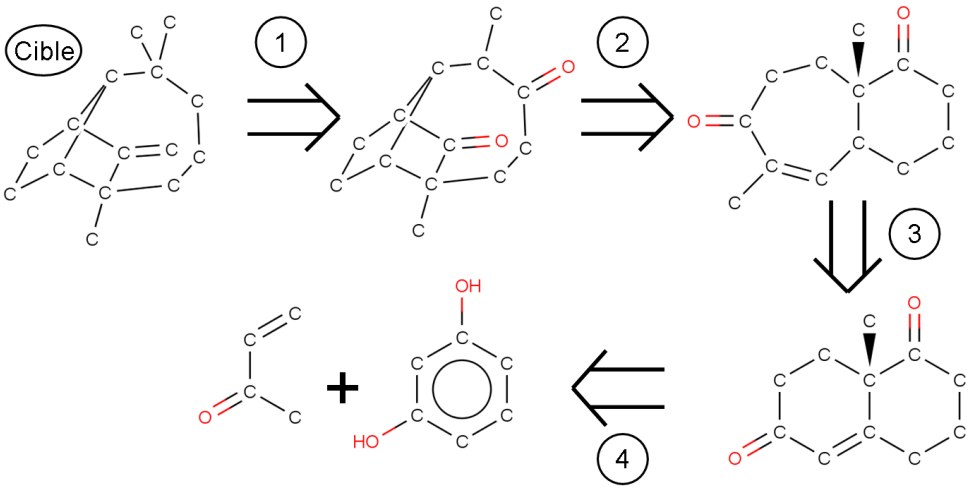


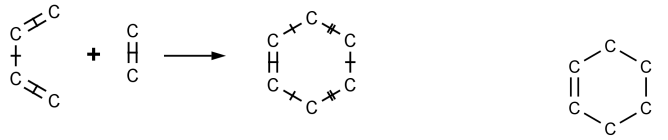
FIG. 3.10 – La rétrosynthèse réussie du longifolène par Corey.

représente conventionnellement les transformations exprimées dans le sens rétrosynthétique, pour les distinguer des réactions exprimant le sens synthétique par une simple flèche. Chaque étape permet soit de simplifier le problème de la synthèse en supprimant un cycle, soit d'aménager la cible par des groupes fonctionnels (ici $C = O$ puis $C - OH$) afin de permettre la simplification suivante.

La rétrosynthèse n'est pas sans rappeler le principe « diviser pour régner »– divide and conquer en anglais – bien connu en algorithmique, qui consiste à décomposer un problème complexe en plusieurs sous-problèmes analogues, indépendants et plus simples à résoudre puis à réitérer tant que nécessaire la décomposition sur ces sous-problèmes. Toutefois, contrairement aux algorithmes classiques du type « diviser pour régner » (quicksort ...), la décomposition d'un problème de synthèse ne garantit pas de converger vers une solution : certaines branches d'exploration peuvent en cours d'analyse être élaguées si l'expert pressent qu'elles conduisent à une impasse ou sont sous-optimales (i.e. rendement trop faible, coût trop élevé...). Une autre branche est alors explorée en reprenant l'analyse à un stade antérieur et en optant pour une réaction alternative à celle choisie initialement. Ce principe correspond à celui du « backtracking » en informatique, même si en pratique, la rétrosynthèse n'explore pas toutes les branches de synthèse possibles, qui sont trop nombreuses, et qu'elle recourt également à des raisonnements dans le sens synthétique. Une fois qu'un plan de synthèse complet et satisfaisant est établi, un processus de validation analyse le déroulement du plan dans le sens synthétique et si ce déroulement est vraisemblable, expérimente le plan en laboratoire. En cas d'échec, la rétrosynthèse est relancée pour explorer de nouvelles voies de synthèse. Pour éviter de découvrir tardivement qu'un plan de synthèse est défectueux à l'issue d'une validation expérimentale longue et coûteuse, il est essentiel que chaque étape de

la rétrosynthèse détermine une réaction aussi vraisemblable que possible – même si à ce stade de l'analyse, la réaction ne peut encore être testée expérimentalement – tout en débouchant sur des précurseurs aussi simples que possible.

Non seulement Corey a proposé une méthodologie rigoureuse de la synthèse ciblée, mais surtout il a rendu la rétrosynthèse opérationnelle, en répertoriant les sous-structures dans les molécules cibles qui permettent des transformations rétrosynthétiques efficaces. En particulier Corey identifie un certain nombre de méthodes de référence qui comportent des atouts stratégiques essentiels (création de cycle, haut rendement, stéréosélectivité ...). La méthode de Diels-Alder, qui servira d'exemple tout au long de ce mémoire, est une de ces méthodes de prédilection car elle construit un cycle à 6 atomes de carbone. Son schéma caractéristique est représenté sur la figure 3.11(a). De ce fait ces méthodes doivent, dans la mesure du possible,



(a) Schéma de réaction (b) Rétron

FIG. 3.11 – La méthode de Diels-Alder

être appliquées en priorité. Afin de déterminer quelles sont les méthodes de synthèse applicables à une molécule cible, sont recherchés dans le graphe moléculaire de la cible les rétrons des différentes méthodes de synthèse, c'est-à-dire, les empreintes caractéristiques laissées par ces méthodes dans les produits de leurs réactions. Le rétron caractéristique de la méthode de Diels-Alder est donné sur la figure 3.11(b). Dans le cas où une sous-structure de la cible s'apparente à un rétron sans lui être rigoureusement identique, le chimiste cherche à établir une réaction intermédiaire qui fasse apparaître le rétron dans le précurseur. L'exemple de la figure 3.12 illustre ce procédé : le rétron de la méthode de Diels-Alder (cf figure 3.11(b)) est d'abord identifié (ici parfaitement) dans la cible. L'application du schéma de la méthode dans le sens rétrosynthétique permet d'aboutir à des précurseurs plus simples.

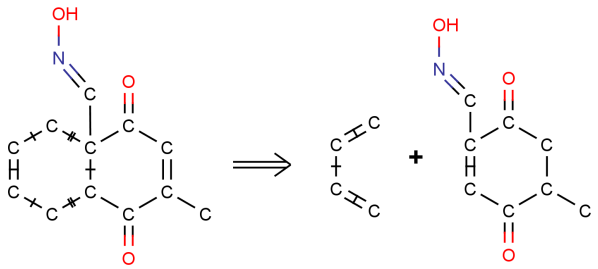


FIG. 3.12 – Une étape de rétrosynthèse, fondée sur les rétrons

Pour converger vers un plan de synthèse qui soit à la fois efficace et réalisable, les experts de la synthèse organique doivent déterminer chaque réaction en mobilisant toutes leurs connaissances, réflexion et intuition. L'analyse des différentes contraintes qu'impose un problème de synthèse est utile pour définir une méthodologie qui puisse guider l'expert dans cette

tâche complexe qu'est le choix de la réaction et ce, à chaque étape de la rétrosynthèse. Ainsi le choix d'une réaction doit être l'aboutissement logique de la propagation de contraintes globales. Cette propagation peut par exemple se décomposer en la succession de phases d'analyse liées à des considérations de nature politique, puis stratégique, tactique et enfin opérationnelle (cf page 11 de Laurenço (1998)). La *politique de synthèse* tout d'abord, est une donnée globale au problème considéré, qui découle du contexte dans lequel doit avoir lieu la synthèse. Cette politique se définit par l'importance accordée à différentes contraintes globales, comme l'efficacité, le coût, la sécurité, le respect de l'environnement ou encore l'originalité du plan de synthèse recherché. En fonction de la politique choisie, l'*analyse stratégique* identifie ensuite à partir de la structure de la molécule cible, des objectifs structuraux précis que doit remplir la réaction recherchée : par exemple, former tel cycle (c'est-à-dire le déconnecter dans le sens rétrosynthétique), créer tel stéréo-centre (c'est-à-dire le supprimer dans le sens rétrosynthétique) ou encore utiliser tel produit de départ comme réactant. Une fois les objectifs stratégiques définis, la *phase tactique* s'emploie à identifier précisément la réaction et les conditions réactionnelles qui permettront d'atteindre ces objectifs. En particulier la phase tactique étudie les aménagements nécessaires à l'application de la méthode de synthèse envisagée et à la protection des parties de la cible qui réagiraient sinon, une fois placées dans les conditions expérimentales envisagées. Enfin la *phase opérationnelle* met en œuvre la réaction, en spécifiant le mode opératoire et les conditions expérimentales de la réaction.

La définition des objectifs stratégiques constitue l'étape la plus déterminante de la rétrosynthèse, puisque c'est elle qui donne la direction que doit suivre le plan de synthèse en cours de conception. Cette stratégie cherche généralement à aboutir à un plan de synthèse opérationnel aussi court que possible pour éviter des étapes intermédiaires qui feraient baisser inutilement le rendement global. Afin de raccourcir le plan de synthèse, chaque étape de la rétrosynthèse doit réduire autant que possible la complexité du problème, c'est-à-dire déterminer en priorité une réaction éliminant au moins une des structures présentes dans la molécule cible et dont la synthèse pose de réelles difficultés. Pour ce faire, Corey identifie cinq grandes classes de stratégies (cf page 16 de Corey et Cheng (1995)) qui peuvent être éventuellement combinées :

- Les *stratégies fondées sur des transformations* caractéristiques de méthodes de synthèse. Cette stratégie déjà évoquée, consiste à reconnaître dans la cible le rétron d'une méthode de synthèse caractéristique. La phase tactique s'emploiera alors à établir une réaction associée à cette méthode de synthèse qui produise la cible.
- Les *stratégies fondées sur les structures* consistent à reconnaître dans la structure de la molécule cible un produit de départ dont on dispose ou tout au moins un produit intermédiaire qu'on sait synthétiser. La phase tactique consiste alors à trouver une réaction qui synthétise la cible et dont un des réactants est le produit de départ ou intermédiaire identifié.
- Les *stratégies topologiques* consistent à trouver une transformation simplifiant la topologie de la cible, en déconnectant une ou plusieurs liaisons dites *stratégiques*. La simplification de la topologie peut consister à réduire le nombre ou la complexité des cycles, à exploiter les symétries de la cible et/ou à décomposer la cible en réactants de taille comparable pour réaliser des plans de synthèse convergents.
- Les *stratégies stéréochimiques* sont similaires aux stratégies topologiques mais cherchent à simplifier la stéréochimie (i.e. la géométrie) de la cible plutôt que sa topologie.
- Enfin les *stratégies fondées sur les groupes fonctionnels* utilisent les groupes fonctionnels comme d'un index pour trouver à partir des fonctions présentes dans la cible, les transformations rétrosynthétiques qui leur sont associées. Une voire plusieurs réactions

intermédiaires consistant en l'ajout, le retrait ou l'échange de groupes fonctionnels, sont alors souvent nécessaires pour faire apparaître le rétron de la transformation identifiée.

La rétrosynthèse de Corey présente une analogie avec les travaux en intelligence artificielle sur la résolution de problèmes et plus particulièrement les démonstrateurs logiques : la molécule cible joue le rôle de théorème à démontrer, les règles d'inférence sont les réactions chimiques, les axiomes sont les produits de départ et les démonstrations sont les plans de synthèses. Cette analogie avec la logique n'est pas fortuite : Corey a intitulé son livre de référence sur la rétrosynthèse « The Logic of Chemical Synthesis » (Corey et Cheng, 1995). Pour cette raison, les travaux de Corey ont encouragé le développement de systèmes d'aide à la synthèse fondés sur la rétrosynthèse et présentés par ailleurs à la section 3.2.2.

3.2 La chémoinformatique pour la synthèse organique

Les travaux exposés dans ce mémoire présentent des applications en chémoinformatique, branche scientifique dont l'objet est de traiter l'information chimique par des moyens informatiques. Pour cette raison, la section 3.2.1 dresse en quelques paragraphes un panorama relativement large de la chémoinformatique[28] avant que les sections 3.2.2 et 3.2.3 traitent des branches spécifiques de la chémoinformatique auxquelles les applications abordées dans ce mémoire se rattachent.

3.2.1 L'émergence de la chémoinformatique

La chimie organique est probablement un des terrains d'application les plus anciens et les plus fertiles de l'informatique en général et de l'intelligence artificielle en particulier. Ainsi le système `Dendral` d'identification de la structure des molécules à partir de leur spectrogramme de masse est souvent cité comme un des premiers systèmes experts (Buchanan et Feigenbaum, 1978). En dehors de ce système qui peut paraître anecdotique, les chimistes ont très tôt développé des outils logiciels pour répondre à leurs divers besoins. Ainsi la complexité des équations de la chimie quantique ont rapidement encouragé au développement d'outils de simulation numérique pour calculer les états d'énergie ou la dynamique des molécules. L'apport évident de l'informatique en tant que puissance de calcul a ainsi permis l'émergence dès les années 50 d'une branche de la chimie, appelée *chimie computationnelle*.

Mais l'application de l'informatique à la chimie ne s'est pas limitée à la simulation et de nombreux autres champs d'applications ont émergé progressivement pour répondre à de nouveaux besoins. Ainsi, il a d'abord fallu modéliser et représenter informatiquement les molécules et réactions chimiques, puis se donner les moyens de les visualiser en 2D puis en 3D. Le besoin d'interagir facilement avec la structure des molécules a créé une demande constante pour disposer d'interfaces graphiques ergonomiques puis d'outils de réalité augmentée. Par ailleurs, l'archivage informatique d'importantes quantités de molécules et de réactions a vite nécessité le développement de *systèmes d'information chimique* adaptés munis de langages de requêtes dédiés, dont la description est développée à la section 3.2.3. La modélisation des molécules s'est progressivement enrichie de nombreux descripteurs physico-chimiques (solubilité par rapport à différents solvants, capacité calorifique, enthalpies et autres grandeurs thermodynamiques ...) ou biologiques (toxicité, activités biologiques spécifiques ...). Ces descripteurs peuvent être calculés à l'aide de modèles formels lorsqu'ils existent, ou sinon prédits

[28] Toutefois cette section ne prétend pas présenter de façon exhaustive toutes les applications de la chémoinformatique et sa lecture ne saurait remplacer la consultation d'ouvrages de synthèse entièrement dévolus à ce sujet, comme par exemple Gasteiger et Engel (2004).

à partir d'exemples, à l'aide de *méthodes QSAR/QSPR*[29] d'apprentissage numérique. Cette dernière approche revient à estimer selon une régression à partir d'exemples, une fonction associant à la structure d'une molécule la valeur d'une grandeur physico-chimique (QSPR) ou d'une activité biologique (QSAR) de cette molécule. Ces fonctions s'obtiennent généralement en représentant la structure de la molécule considérée par un vecteur de descripteurs scalaires, notamment topologiques (i.e. histogrammes de séquences d'atomes, de groupes fonctionnels ...) ou géométriques (i.e. histogrammes de conformations spatiales ...) puis à appliquer une méthode de régression numérique (régression linéaire, logistique, SVR ...) pour prédire la valeur du descripteur cible à partir du vecteur de grande dimension. Par ailleurs, les outils de *chimie combinatoire* ont permis de produire de très grandes collections de molécules virtuelles à partir de grammaires adaptées. Ces collections sont ensuite exploitées par les techniques de *criblage virtuel* pour sélectionner à l'aide des prédictions QSAR/QSPR, les molécules présentant un intérêt potentiel, notamment en terme d'activités biologiques. Ce faisant, l'industrie pharmaceutique est en mesure de tester virtuellement un grand nombre de substances actives potentielles avant de les tester expérimentalement par criblage haut-débit. Le nombre potentiellement infini de molécules à tester et les calculs toujours plus complexes de descripteurs font du criblage virtuel une excellente application pour les grilles de calcul distribué. Enfin un domaine en plein développement est celui des outils de *docking* permettant à l'aide de modèles géométriques, de mesurer l'affinité des ligands à se fixer sur les récepteurs de protéines. Le docking met ainsi en évidence la jointure entre chimie et biologie – l'interface étant la biologie moléculaire – et par voie de conséquence, la passerelle qui relie *bioinformatique* à ce qui a été baptisé très récemment chémoinformatique et qui précédemment était appelée chimie informatique par certains.

À ce sujet, il est intéressant de constater que la biologie a réussi en quelques années ce que la chimie peine à réaliser en des décennies : imposer l'idée d'une discipline scientifique transverse, cristallisée sous un terme unique « bioinformatique », quand bien même cette discipline regroupe des problèmes tout aussi indépendants que peuvent l'être ceux de la chimie informatique. Ce n'est en effet qu'en 1998 que le terme *chémoinformatique*[30] apparaît dans un article de F. Brown (Brown, 1998) pour désigner selon l'auteur, le traitement informatique de l'information chimique afin d'améliorer la conception de nouveaux médicaments. Depuis, l'acception du terme chémoinformatique s'est étendue pour désigner le traitement informatique de l'information chimique en général. Aujourd'hui le terme chémoinformatique semble remporter une certaine adhésion de la part de la communauté scientifique concernée, suite notamment à la rédaction d'ouvrages de synthèse sur le sujet (Gasteiger et Engel, 2004; Leach et Gillet, 2003), même si la définition exacte des contours de la discipline est toujours sujette à polémique. La définition initiale donnée par F. Brown mérite d'être mentionnée car elle fait apparaître un lien fort entre sa vision de la chémoinformatique et l'extraction de connaissances : « chemo-informatics is the mixing of information resources to transform **data into information** and **information into knowledge**, for the intended purpose of making decisions faster in the arena of drug lead identification and optimisation ». Si cette définition peut sembler a posteriori trop réductrice, elle suggère que la fouille de données pourrait être un des axes les plus prometteurs du développement de la chémoinformatique.

Dans la mesure où une présentation exhaustive de toutes les applications précitées de la chémoinformatique est impossible, seuls les sujets ayant un lien évident avec le reste de

[29] De l'anglais, Quantitative Structure-Activity Relationship et Quantitative Structure-Property Relationship

[30] De l'anglais, chemo-informatics par analogie avec le terme bioinformatics.

ce mémoire sont développés dans ce qui suit. Ainsi la section 3.2.2 présente succinctement les systèmes d'aide à la résolution de problèmes de synthèse dans la mesure où ces derniers constituent les fondations historiques à partir desquelles la chémoinformatique s'est intéressée aux réactions chimiques. La section 3.2.3 présente ensuite très brièvement les systèmes d'information chimique et plus particulièrement les bases de données de réactions, dans la mesure où les méthodes de fouille de graphes proposées dans ce mémoire ont comme première application la fouille des données issues de ces systèmes. Enfin les travaux antérieurs de fouille de données appliqués à la chémoinformatique sont résumés dans la section 3.3 et en particulier la thèse de Sandra Berasaluce (Berasaluce, 2002) sur l'extraction de connaissances à partir de bases de données de réactions et dont cette thèse peut être vue comme le prolongement d'un point de vue applicatif. D'autres problèmes plus spécifiques de chémoinformatique ne sont pas abordés ici mais dans les chapitres auxquels ces problèmes se rattachent le plus. Ainsi le problème de la classification non supervisée des réactions chimiques est abordé au chapitre 6 alors que celui de l'accessibilité synthétique et de la détection des liaisons stratégiques est traité au chapitre 7.

3.2.2 Les systèmes d'aide à la résolution de problèmes de synthèse

Sous l'impulsion initiale de Vleduts (Vleduts, 1963), plusieurs systèmes informatiques d'aide à la synthèse ont vu le jour. Ces systèmes ont eu le temps d'évoluer et de se multiplier au cours des bientôt quarante dernières années si bien qu'il est exclu d'en faire ici une description exhaustive et précise. Les lecteurs intéressés peuvent se reporter aux articles récents (Pfoertner et Sitzmann, 2003; Ott, 2004; Hanessian, 2005; Todd, 2005; Chen, 2006) ou aux thèses (Laurenço, 1985; Gien, 1998) consacrés à ce sujet.

Les systèmes d'aide à la synthèse servent essentiellement deux applications. La première est la prédiction de réactions dans le sens synthétique : étant donné un ensemble de molécules mises en présence l'une de l'autre sous certaines conditions expérimentales, est-il possible de prédire si une réaction se déclenche, et le cas échéant, quels en seront les produits et avec quel rendement ? La seconde application est la détermination d'un plan de synthèse d'une molécule cible à partir de la rétrosynthèse de cette molécule. Bien évidemment, les deux applications sont complémentaires puisqu'un plan de synthèse une fois déterminé nécessite d'être validé dans le sens synthétique. Certains systèmes d'aide à la conception de plans de synthèse intègrent un module de prédiction des réactions pour combiner des raisonnements dans les deux sens synthétique et rétrosynthétique. Indépendamment de l'application visée, les systèmes se répartissent selon Ugi (Ugi *et al.*, 1988) en deux grandes catégories que sont les *systèmes formels ou logiques* et les systèmes *fondés sur la connaissance ou empiriques*. Ces deux catégories s'avèrent en réalité les deux positions extrêmes entre lesquelles se situent les systèmes développés selon qu'ils mettent plus ou moins en avant les principes de l'une ou de l'autre catégorie.

À l'une des extrémités se trouvent donc les systèmes formels caractérisés par une génération combinatoire systématique d'un grand nombre de réactions potentielles, que ce soit dans le sens synthétique pour la prédiction de réactions ou dans le sens rétrosynthétique pour la conception de plans de synthèses. Ces derniers produisent ainsi des plans de synthèse complets sans que l'expert puisse intervenir pour guider la recherche. L'exemple le plus extrême est peut-être le système `RAIN` reposant sur le modèle de Dugundji-Ugi (Ugi *et al.*, 1994). Ce modèle représente les molécules par la matrice d'adjacence de ses électrons de valence : le coefficient c_{ij} représente le nombre de doublets d'électron qui participent à la liaison entre les atomes i et j. Les réactions sont elles modélisées par des matrices de transition représentant la

redistribution des électrons de valence entre atomes. De cette manière la matrice représentant les produits d'une réaction s'obtient en faisant la somme de la matrice des réactants et de la matrice de transition associée à la réaction. À partir d'un ensemble de matrices de transitions et d'une matrice représentant une molécule de départ (dans le sens synthétique) ou une molécule cible (dans le sens rétrosynthétique), `RAIN` explore alors l'espace d'état des molécules accessibles en appliquant de façon systématique les matrices de transitions. Si le système `RAIN` a donné lieu à des résultats intéressants, notamment pour élucider le déroulement d'une réaction composée d'une séquence de réactions élémentaires, il présente l'inconvénient majeur de générer de façon aveugle un trop grande nombre de solutions dont la plupart sont irréalistes ou inintéressantes.

Pour éviter l'« égarement » des méthodes purement formelles telles que `RAIN`, plusieurs méthodes proposent de guider leur processus de génération combinatoire de réaction ou transformation, par l'intégration de règles ou de lois modélisant la connaissance qu'ont les chimistes de la réactivité. Ainsi le système `CAMEO` de prédiction de réaction intègre l'essentiel des modèles théoriques de « mécanique réactionnelle » afin de ne prédire que les réactions les plus réalistes (Jorgensen *et al.*, 1990). De même le système `SYNGEN` (Hendrickson et Toczko, 1989) d'analyse rétrosynthétique considère lors d'une phase stratégique un grand nombre de décompositions possibles de la molécule cible en déconnectant successivement différentes liaisons jusqu'à obtenir des fragments proches de produits de départ. Àvant d'être retenue, chaque décomposition doit toutefois être validée lors d'une phase d'analyse tactique qui vérifie que chaque étape de la décomposition peut se traduire en une réaction plausible.

Contrairement aux systèmes précédents fondés sur un processus de génération combinatoire de plan de synthèse, les systèmes empiriques privilégient une représentation explicite des connaissances en synthèse organique afin de guider la recherche vers quelques plans de synthèse les plus plausibles. Ces systèmes empiriques sont donc construits autour d'une base de connaissance constituée de schémas de transformation et utilisent un module de *perception* chimique pour analyser la structure de la molécule cible (i.e. détection de cycles, de groupes fonctionnels, de stéréocentres dans le graphe moléculaire de la cible). La plupart de ces systèmes s'inspirent des travaux remarquables de formalisation de la rétrosynthèse réalisés par Corey (cf section 3.1.4). Parmi eux, on compte `OCCS` (Corey et Wipke, 1969) qui est le plus ancien, `LHASA` (Corey, 1971), `SECS` (Wipke, 1974), `WODCA` (Gasteiger *et al.*, 1990, 1992), Resyn (Vismara *et al.*, 1992) et bien d'autres encore. Le plus célèbre et peut-être aussi le plus mature de ces systèmes est `LHASA` dérivé de `OCCS` dont le développement conduit par Corey (Corey, 1971) lui-même, se poursuit depuis plus de 30 ans. Ce système s'appuie sur une base de connaissances répertoriant de façon structurée plus de 2000 transformations. Le système expert détecte ainsi dans la cible les rétrons, même partiels, de ces transformations qu'il applique dans le sens rétrosynthétique selon un chaînage arrière. Les transformations en mesure d'être appliquées servent une stratégie choisie par l'utilisateur parmi les cinq possibles identifiées par Corey (voir la liste exacte donnée dans la section 3.1.4). `SECS` (Wipke, 1974) adopte une conception très similaire à celle de `LHASA`.

Contrairement aux systèmes précédents, le système `CHIRON` (Hanessian *et al.*, 1990) plus récent ne permet pas d'établir un plan de synthèse complet mais seulement d'identifier des produits de départ intéressants permettant potentiellement d'aboutir à la synthèse de la cible. D'autre part sa stratégie n'est pas globale mais essentiellement fondée sur des considérations stéréochimiques. Son idée consiste – puisque la formation des stéréocentres d'une molécule est un problème délicat qu'il est préférable d'éviter – à décomposer la cible en des fragments isolant ses stéréocentres de sorte que chacun de ces fragments soit structurellement très proche d'un produit de départ connu et répertorié dans une base de données. Un des derniers systèmes

en date est `WODCA` (Gasteiger *et al.*, 1992; Pfoertner et Sitzmann, 2003) développé par l'équipe de Gasteiger. L'observation selon laquelle les systèmes antérieurs ont tendance à imposer une logique inflexible trop éloignée du raisonnement naturel de l'expert est à l'origine du développement de `WODCA` qui veut privilégier un haut niveau d'interactivité avec l'expert et qui lui vaut, selon ses auteurs, le titre de système de seconde génération. En pratique `WODCA` propose un peu comme `CHIRON`, de retrouver à partir des caractéristiques structurelles de la cible les produits de départ les plus pertinents parmi une base de produits commerciaux. Lorsqu'aucun produit de départ convenable n'est trouvé, un module complémentaire propose de déconnecter les liaisons qu'il a identifié comme stratégiques (cf définition de la section 3.1.4) pour réitérer la recherche des produits de départs sur les fragments de précurseurs. L'identification de liaisons stratégiques étant en relation directe avec le chapitre 7, n'est pas développée ici mais dans la section 7.1.1. L'interactivité se situe dans la possibilité pour l'expert de choisir lui-même les critères qualifiant le degré de pertinence d'un produit de départ par rapport à la cible. Toutefois, l'établissement du plan de synthèse reliant les produits de départ à la molécule cible reste entièrement à la charge de l'expert – tout comme pour `CHIRON` – et son titre de système de seconde génération peut pour cette raison, paraître excessif (cf détails p. 70-71 de Gien (1998)).

Il est remarquable que les systèmes les plus récents comme `WODCA` et `CHIRON` apparaissent également moins ambitieux que les systèmes précédents comme `SYNGEN` ou `LHASA`, alors qu'on pourrait s'attendre au contraire. À l'opposé des premiers systèmes monolithiques qui privilégiaient l'autonomie et la prise de décision, les derniers systèmes semblent davantage conçus comme des outils destinés à répondre à des problèmes spécifiques. L'expert reste la pièce maîtresse au cœur de la conception du plan de synthèse et l'informatique n'est qu'une boîte à outils lui permettant d'accéder à des informations complémentaires de ses connaissances. Cette évolution dans la conception du rôle que peuvent jouer les systèmes d'aide à la conception de plans de synthèse se retrouve dans l'histoire du développement du système Resyn (pour RÉtroSYNthèse) suivi de celui de Resyn Assistant (Vismara *et al.*, 1992; Vismara, 1995; Vismara *et al.*, 1998; Gien, 1998; Laurenço, 1998). Ce projet initié par la société Roussel Uclaf et réalisé dans le cadre du Gdr Ticco[31] revêt ici une importance toute particulière puisque il existe un lien de filiation reliant Resyn à cette thèse, en passant par l'outil ReSyn Assistant puis la thèse de Sandra Berasaluce (cf section 3.3.2). Au départ, ReSyn était pensé comme un outil d'aide à la conception de plans de synthèse fondé sur le principe de transformation rétrosynthétique utilisé dans `LHASA`. Une des originalités du projet était la volonté d'intégrer des travaux de recherche réalisés dans le domaine de la représentation des connaissances et plus spécifiquement des systèmes de représentations d'objets et de raisonnement par classification (Napoli, 1992; Napoli *et al.*, 1994), afin de faciliter l'acquisition et la maintenance de la base de connaissances de ReSyn. Grâce aux enseignements que le développement du projet ReSyn a pu apporter, un nouveau projet fut démarré en 1997 sous le nom de ReSyn Assistant. L'objectif de celui-ci n'était plus tant de fournir un système d'aide à la synthèse « autonome » manipulant et raisonnant sur des connissances, mais de proposer une plate-forme logicielle d'expérimentation écrite en Java destinée à faciliter l'analyse de problèmes de synthèse. Pour ce faire, un système de perception original des molécules a été développé dans ReSyn Assistant, avant d'être étendu aux réactions. Cette perception est fondée sur une modélisation multi-niveaux des graphes moléculaires (Vismara et Laurenço, 2000) et a été notamment exploitée par S. Berasaluce pour fouiller les BdR (cf détails du système à la section 3.3.2 p. 69). Ces dernières font l'objet de la section suivante.

[31] Gdr 1093 du CNRS « Traitement Informatique de la Connaissance en Chimie Organique », 1993 – 2001.

3.2.3 Les systèmes d'information chimique

Les systèmes d'information chimique sont nés à la fois, du besoin de recenser un nombre toujours croissant des molécules et des réactions chimiques connues et à la fois, de la possibilité d'archivage et d'indexation de l'information que proposent les moyens informatiques modernes. En effet, les progrès de la chimie et de son industrie reposent sur une connaissance collective des millions de molécules déjà synthétisées et des millions de réactions chimiques qui interviennent dans ces synthèses. Avant l'ère informatique, les chimistes n'avaient d'autres possibilités, pour trouver l'information qu'ils recherchaient, que de consulter des recueils d'indexation comme les Chemical Abstracts (CAS) ou des ouvrages de compilation thématique à partir desquels ils retrouvaient les publications scientifiques pertinentes. Les premiers systèmes de bases de données ont permis d'indexer l'information à partir de mots-clés et de la nomenclature des molécules. Mais un progrès décisif a été marqué lorsqu'au début des années 80, certaines bases de molécules ont pu être interrogées à l'aide de requêtes structurales. Ces systèmes ont ainsi pu bénéficier des progrès accomplis dans le cadre des systèmes d'aide à la synthèse, pour représenter de façon adéquate les molécules par des graphes et pour y rechercher des sous-structures. Ainsi le système `DARC` créé en 1966 par J. E. Dubois (Dubois et Sobel, 1985) sera le premier à interroger des CAS à partir de requêtes structurales spécifiées qui plus est, dans un terminal graphique. Par ailleurs, la société MDL propose dès 1979 le premier système commercial d'interrogation de bases de molécules appelé `MACCS`, suivi en 1982 du premier système d'interrogation de bases de réactions appelé `REACCS`.

Aujourd'hui les systèmes d'information chimique modernes permettent d'accéder en un temps raisonnable à des millions de molécules et de réactions selon des langages de requêtes très riches en possibilités. Quelques unes des plus importantes bases de données de réactions sont décrites sur la figure 3.13. Chaque base de données présente des caractéristiques

Nom de la base de données	Société éditrice	Nombre de réactions	Période d'acquisition	Remarques
CASREACT	CAS®	+17 M	1840 – ...	Très grande, issue des CHEMICAL ABSTRACTS®
BEILSTEIN	Elsevier	+15 M	1771 – ...	Variée, couvrant une longue période.
ORGSYN	Symyx®/MDL®	+5 K	1981 – ...	Petite mais très fiable dont le contenu a été vérifié expérim.
CHEMINFORM RX	Symyx®/MDL®	+1,2 M	1991 – ...	Variée, orientée méthodologie de synthèse (i.e. orientée réactions plus que produits).
RX-JSM	Symyx®/MDL®	+90 K	1980 – ...	Sélective, insistant sur le caract. innovant des réactions.
REFLIB	Symyx®/MDL®	+200 K	1900 – 1991	Compilation d'anciennes bases sélectives.

FIG. 3.13 – Principales bases de données de réactions

particulières et répond ainsi à des besoins différents. Les méthodes développées ont pu ainsi être testées expérimentalement sur les bases ORGSYN, CHEMINFORM RX, JSM et REFLIB accessibles à travers le portail TITANESCIENCES de l'Inist (cf titanesciences.inist.fr).

Si le contenu entre ces bases de données diffère du point de vue de la chimie, la modélisation informatique des réactions reste identique dans son principe d'une base à l'autre. La figure 3.14 reproduit une copie écran de la représentation d'une réaction issue d'une de

ces bases de données. L'information essentielle est représentée par l'équation de la réaction

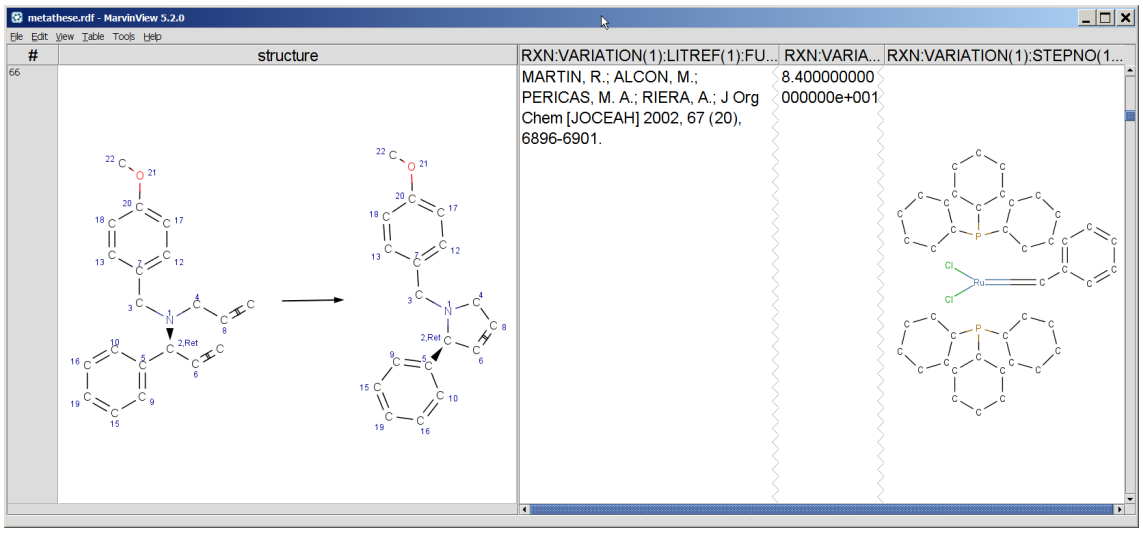
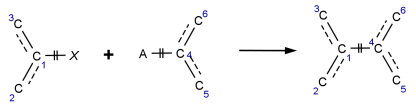


FIG. 3.14: Représentation d'une réaction extraite de CHEMINFORM RX. Les colonnes représentent respectivement l'équation de la réaction, la référence bibliographique associée, le rendement (ici 84 %) et le ou les catalyseurs.

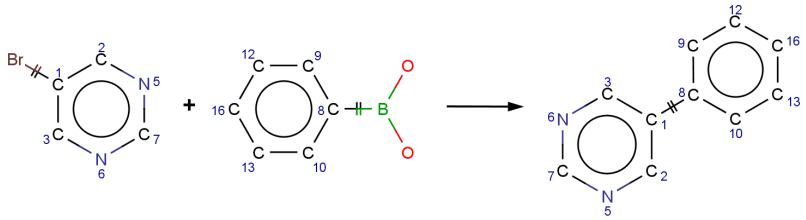
(colonne structure). Chaque atome est identifié dans le terme des réactants comme dans celui des produits par un entier unique appelé dans ce qui suit indice d'appariement. Les liaisons brisées, formées ou modifiées par la réaction sont barrées par un ou deux traits perpendiculaires. D'autres informations complémentaires sont disponibles, comme la référence bibliographique où cette réaction est décrite (seconde colonne), le rendement moyen dans les conditions expérimentales décrites par ailleurs (troisième colonne), le graphe moléculaire du ou des catalyseurs (quatrième colonne) et bien d'autres informations non représentées sur cet exemple comme les conditions réactionnelles en terme de température, solvants...

Les bases de données de réactions ne seraient toutefois que des collections de documents électroniques si elles ne disposaient pas d'un langage de requêtes pour permettre un accès rapide et pertinent à l'information que recherchent les spécialistes de la synthèse organique. Du point de vue purement informatique, les bases de données de molécules ou de réactions présentent de nombreux problèmes originaux. La principale originalité vient du fait que les molécules et les réactions sont des objets qui ne sont pas seulement représentés par une juxtaposition de propriétés, comme dans les bases de données relationnelles, mais essentiellement par des graphes étiquetés. De ce fait, les interrogations que se posent en pratique les chimistes, peuvent souvent se formuler en terme d'agencement relatif de sous-structures. Un langage de requêtes adapté doit donc pouvoir spécifier en plus des opérateurs logiques classiques, des opérateurs de sélection fondés sur des sous-structures. Le traitement de ces opérateurs spécifiques nécessite la recherche dans les données de sous-graphes isomorphes – de façon exacte ou approchée – à la sous-structure spécifiée par l'opérateur (cf section 2.3.1). Le problème étant NP-complet, le temps de calcul que nécessite une telle requête peut s'avérer très long si cette requête est traitée séquentiellement sur des millions de graphes moléculaires. Les techniques d'indexation efficace des données et d'optimisation dans les moteurs de requêtes sont donc essentielles. Parmi les langages de requêtes structurales, certains se sont inspirés de la notation linéaire des graphes moléculaires proposée par Wipke dans `SECS` (Wipke, 1974), pour pouvoir spécifier les requêtes en mode texte (i.e. sous forme de chaîne de caractères).

C'est notamment le cas de SMARTS/SMILES (Weininger, 1988) ou plus récemment du langage MQL (Proschak *et al.*, 2007). Mais la plupart des interrogations se font maintenant sous forme graphique, en spécifiant la requête sous la forme d'une combinaison logique de sous-structures à rechercher dans les bases de molécules ou de réactions. La figure 3.15(a) donne un aperçu des possibilités offertes par ce type de requêtes graphiques.



(a) Requête



(b) Exemple de réponse

FIG. 3.15: Exemple d'une requête simple (a) posée à un système de gestion d'une BdR et d'une réaction (b) satisfaisant la requête. Le symbole X dans la requête permet de représenter tout halogène (i.e. Br, Cl...). Le symbole A représente tout atome qui n'est pas d'hydrogène. Les liaisons doubles en pointillé représentent les liaisons aromatiques.

Aujourd'hui, les bases de molécules et de réactions constituent un outil de travail quotidien pour les experts de la synthèse organique, bien plus que ne l'ont jamais été les outils d'aide à la synthèse présentés à la section 3.2.2. Cependant, si les systèmes d'information chimique permettent de répondre efficacement à des interrogations précises formulables dans leur langage de requête, ils ne peuvent répondre aux questions plus ouvertes abordées par la fouille de données : quelle est la distribution des groupes fonctionnels dans les données et quelles sont les règles sous-jacentes ? Quelles sont les caractéristiques structurelles propres à telle famille de réactions ou telle famille de molécules ? ... Certains travaux de recherche allant dans ce sens commencent à apparaître et sont présentés à la section suivante.

3.3 L'extraction de connaissances en chimie organique

Les méthodes générales de fouille de données apparues au début des années 90 puis les méthodes de fouille de graphes depuis le début des années 2000 commencent à être exploitées dans le cadre d'applications de chémoinformatique et leur utilisation est certainement amenée à se généraliser. La section 3.3.1 fait mention des principaux travaux en chémoinformatique qui ont eu recours à la fouille de données et plus particulièrement aux méthodes de fouille de graphes ou de programmation logique inductive introduites au chapitre 2. Toutefois la plupart de ces travaux traitent d'ensembles de molécules et non de réactions chimiques. La section 3.3.2 fait le point concernant l'utilisation de méthodes de fouille de données appliquées spécifiquement aux BdR. En particulier cette section développe les travaux de Sandra

Berasaluce (Berasaluce, 2002) dont les résultats ont motivé le sujet de cette thèse.

3.3.1 Applications de la fouille de données en chémoinformatique

Les problèmes QSAR/QSPR (voir à ce sujet le chapitre 10 de Gasteiger et Engel (2004)) font un usage intensif des méthodes d'analyse de données et d'apprentissage automatique, afin de prédire certaines grandeurs physico-chimiques associées aux molécules. Cependant, les méthodes QSAR/QSPR ne traitent généralement pas de problèmes d'extraction de connaissances – i.e. dont le but n'est plus seulement de prédire un phénomène mais surtout de le comprendre – dans la mesure où les modèles (SVM, réseaux de neurones, etc) que ces méthodes produisent sont difficilement interprétables. En comparaison, les méthodes dévolues à l'extraction de connaissances, comme les méthodes d'extraction de règles d'association fréquentes (cf section 2.1.2), restent peu utilisées en chémoinformatique. Une des rares exceptions est le travail de S. Berasaluce décrit à la section suivante. Plus récemment, Auer et Bajorath (2008) utilisent les motifs émergents (Dong et Li, 1999; Bailey *et al.*, 2002) dérivés des motifs d'attributs fréquents (cf section 2.1.4) pour discriminer les conformations de certaines molécules dans leur état biologiquement actif des conformations non actives de ces mêmes molécules (i.e. dans leur conformation d'énergie minimale). Si les méthodes de fouille de données sont encore relativement peu utilisées en chémoinformatique, cette dernière aborde la problématique de l'extraction de connaissances à travers certaines méthodes plus anciennes d'apprentissage à partir d'exemples. Certaines méthodes de classification assortissent en effet leurs décisions par des éléments d'explication qui, par leur interprétation, peuvent contribuer à extraire des éléments de connaissance. Dans certains cas, ces éléments d'explications prennent la forme de motifs structuraux de molécules, voire de réactions. Ces méthodes procèdent alors par généralisation grâce au calcul des sous-graphes maximaux communs aux graphes moléculaires des exemples (cf SGCM à la sect. 2.3.1). Il en est ainsi de la méthode `CNN` (Régin *et al.*, 1995; Régin, 1995) pour prédire les liaisons stratégiques, à laquelle se réfère le chapitre 7, des travaux de Cuissart (2004) pour la prédiction de la « biodégradabilité facile » des molécules, ou encore du système `grams` de classification de réactions (Jauffret *et al.*, 1995), évoqué à la section suivante.

Une des principales applications produisant des motifs structuraux de molécules est la classification de molécules selon leur niveau de toxicité ou d'activité biologique, dans le cadre d'une application de criblage virtuel. Cette application se fonde généralement sur deux ensembles de molécules spécifiées par leur graphes moléculaires, regroupant respectivement les exemples positifs, c'est-à-dire les molécules avérées actives, et les exemples négatifs, c'est-à-dire les molécules avérées inactives. L'objectif du problème est de déterminer un modèle de prédiction de molécules actives. Ce problème est intéressant car il a déjà pu être abordé comme un problème de fouille de graphes. En effet, les biochimistes savent depuis longtemps que certaines formes de toxicité ou d'activité biologiques peuvent s'expliquer par la présence de sous-structures et donc de sous-graphes dans les graphes moléculaires. Dès lors, la prédiction des molécules actives est un problème de fouille de graphes qui consiste à determiner les motifs structuraux discriminants (i.e. beaucoup plus fréquents dans les exemples positifs que dans les exemples négatifs et inversement). Les difficultés algorithmiques posées par la fouille de graphes ont restreint les premiers programmes à ne tenir compte que des séquences d'atomes, comme par exemple `CASE` (Klopman, 1984) ou plus récemment MOLFEA (Helma *et al.*, 2004) pour prédire la mutagénicité[32] de molécules.

[32]La mutagénicité est la propriété qu'ont certains composants chimiques de provoquer des mutations génétiques.

Pour cette raison, les méthodes de recherche de sous-graphes fréquents (cf section 2.4.2) ont dès leur apparition au début des années 2000, trouvé en la chémoinformatique un terrain d'application fertile, comme le souligne Fischer et Meinl (2004). Les résultats obtenus par certains de ces travaux ont été reconnus par la communauté chémoinformatique à travers des publications dans des journaux de référence du domaine comme *ACS Journal Of Chemical Information Modeling* (JCIM). Un des exemples les plus représentatifs est probablement les travaux décrits dans Kazius *et al.* (2006) s'appuyant sur l'algorithme Gaston (cf section 2.4.2). Leur approche consiste à extraire les sous-graphes fréquents des exemples positifs puis pour chacun de ces motifs noté M, faire un test d'hypothèse (grâce au calcul d'une « p-value ») associée à l'hypothèse nulle selon laquelle la probabilité d'observer un exemple positif selon un tirage aléatoire est indépendant du fait qu'il contienne le motif M ou non. Le motif associé à la plus petite « p-value » est sélectionné comme le motif le plus discriminant puis le traitement est réitéré sur l'ensemble des exemples ne contenant pas ce motif. Cette méthode non seulement donne de bons résultats mais surtout se distingue des méthodes précédentes en cela qu'elle tient compte de tous les motifs structuraux possibles (au delà d'une certaine fréquence) tout en étant capable de traiter plusieurs milliers d'exemples.

3.3.2 L'extraction de connaissances à partir des bases de données de réactions

Le problème de l'extraction de connaissances à partir d'un ensemble de réactions a été initialement abordé par des méthodes de classification non supervisé de réactions (Wilcox et Levinson, 1986; Fujita, 1986; Rose et Gasteiger, 1994; Jauffret *et al.*, 1995; Hendrickson, 1997; Satoh *et al.*, 1998; Wang *et al.*, 2001; Zhang et Aires-de Sousa, 2005). L'objet de ces travaux est de regrouper les réactions jugées similaires, parfois d'en faire des classes de réactions caractérisées par des schémas réactionnels, et parfois même d'extraire automatiquement des données une hiérarchie de classes de réactions. Certaines de ces approches (Wilcox et Levinson, 1986; Fujita, 1986; Jauffret *et al.*, 1995) se fondent notamment sur la notion de graphes condensés ou superposés de réactions (Vladutz, 1986), sur laquelle s'appuie également la méthode de recherche des schémas de réactions fréquents proposée au chapitre 4. Intuitivement, les graphes de réactions consistent à superposer les graphes moléculaires des réactants et des produits d'une réaction. Ces graphes sont définis formellement à la section 4.5.

Les approches précédentes présentent différents inconvénients. Ainsi, certaines de ces méthodes (Fujita, 1986; Rose et Gasteiger, 1994; Hendrickson, 1997; Wang *et al.*, 2001) proposent des modèles de classification ad hoc, reposant sur des choix restrictifs guidés par les connaissances qu'ont les chimistes des réactions. L'intégration des connaissances du domaine, qui peut être vue comme un élément positif, se fait toutefois au dépens de l'information contenue dans les données : ces méthodes n'extraient en effet des graphes moléculaires que l'information qui répond directement au besoin du modèle. D'autres approches sont fondées sur des méthodes sophistiquées de généralisation à partir d'exemples. Ainsi le système `grams` de Jauffret *et al.* (1995) construit une hiérarchie de schémas de réactions à l'aide de généralisations successives de graphes condensés de réactions. Ces calculs lourds et complexes ne semblent pas conçus pour traiter des grands ensembles de réactions tels que les BdR. Enfin, d'autres méthodes (Satoh *et al.*, 1998; Zhang et Aires-de Sousa, 2005) plus récentes se fondent sur les méthodes de voisinage ou les cartes auto-organisatrices, et en particulier les cartes de Kohonen pour produire une « cartographie » des réactions. Ces méthodes originales ne produisent toutefois pas des classes de réactions décrites par des schémas réactionnels mais seulement des regroupements (i.e. clusters) de réactions définis relativement à une distance définie de

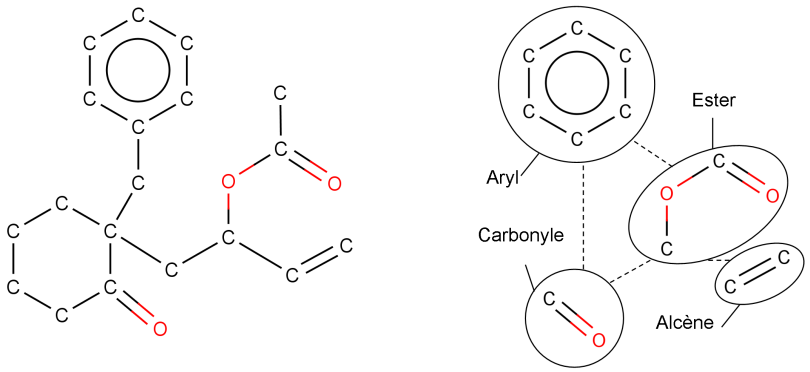
façon ad hoc. En outre, l'interprétation qui peut être faite des regroupements de réactions n'est pas évidente. En comparaison, la fouille des graphes contenus dans les BdR peut permettre un meilleur passage à l'échelle que les méthodes d'apprentissage à partir d'exemples, sans pour autant réaliser des simplifications ad hoc de l'information topologique. Enfin la fouille de motifs (i.e. de schémas réactionnels) dans les BdR permet de qualifier des classes de réactions, contrairement aux méthodes de voisinage.

Alors que les méthodes de recherche de sous-graphes fréquents ont permis de fouiller des ensembles de graphes moléculaires, ces mêmes méthodes n'ont pas encore, à notre connaissance, été appliquées à un ensemble d'équations de réactions chimiques (ce problème fait d'ailleurs l'objet du chapitre 4). Les méthodes de recherche d'attributs fréquents et d'extraction des règles d'association (cf section 2.1.2) ont toutefois pu être appliquées aux bases de données de réactions, au cours de la thèse de Sandra Berasaluce (Berasaluce, 2002; Berasaluce *et al.*, 2004). Ainsi S. Berasaluce aborde le problème général de l'extraction de connaissances à partir de bases de données de réactions selon un processus en deux temps : i) en modélisant certaines sous-structures essentielles de graphes moléculaires, appelées *blocs* par des attributs booléens ii) puis en extrayant les règles d'association fréquentes dans les ensembles de blocs représentant les réactions. Cette méthode a ainsi l'avantage de s'appuyer sur les nombreux algorithmes de recherche motifs d'attributs fréquents, tout en tenant compte de la structure de graphe moléculaire présente dans les équations de réactions.

La définition des blocs fait partie du modèle de perception multi-niveaux des graphes moléculaires (Vismara et Laurenço, 2000), implémenté dans ReSyn Assistant. Ce modèle distingue trois niveaux qualifiés de micro, méso et macro dont le degré d'abstraction va croissant. Le niveau *micro* correspond à rien d'autre que les graphes moléculaires eux-mêmes. Le niveau *méso* correspond à la représentation de la molécule par des blocs et le niveau *macro* par des groupes liés de blocs appelés systèmes de blocs. Les blocs sont des sous-graphes connexes du graphe moléculaire qui sont perçus comme remarquables du point de vue des chimistes. Ces blocs peuvent être de nature topologique, fonctionnelle ou stéréochimique. Les blocs topologiques définissent le squelette de la molécule, à travers des blocs regroupant des cycles, des liens entre cycles (bridges), des chaînes... Les blocs stéréochimiques correspondent aux différents stéréocentres de la molécule. Enfin les blocs fonctionnels sont les *fonctions* définies formellement comme les sous-graphes connexes maximaux ne comportant que des liaisons multiples ou des liaisons hétéroatomiques. Les fonctions correspondent à une des définitions formelles qu'il est possible de donner à la notion plus floue de groupe fonctionnel. Les noms des fonctions incluses dans chaque bloc fonctionnel sont ensuite déterminés en naviguant dans une hiérarchie des fonctions les plus courantes ordonnées par inclusion. Les blocs voisins de même nature (i.e. topologiques, fonctionnels...) sont ensuite reliés par une arête pour former une représentation abstraire de la molécule appelée *graphe de blocs*, dont la figure 3.16 donne un exemple.

S. Berasaluce étend le principe de la décomposition par blocs aux équations de réactions chimiques. La perception par blocs est réalisée séparément sur les graphes moléculaires des réactants et des produits. Les indices d'appariement (ou mapping) entre atomes des réactants et des produits sont ensuite utilisés pour apparier les blocs entre eux. Cet appariement permet d'identifier trois catégories de blocs :

- Les *blocs inchangés* sont ceux présents dans les deux termes de l'équation et constitués du même ensemble d'atomes (i.e. des mêmes indices d'appariements).
- Les *blocs créés* sont les blocs présents dans le terme droit des produits mais qui ne rentrent pas dans la catégorie des blocs inchangés.
- Les *blocs détruits* sont les blocs présents dans le terme gauche des réactants mais qui



(a) Molécule initiale (b) Graphe de blocs

FIG. 3.16: Exemple de décomposition en blocs fonctionnels

ne rentrent pas dans la catégorie des blocs inchangés.
La figure 3.17 donne un exemple de décomposition d'une équation chimique en blocs fonctionnels. Chaque bloc est identifié par le nom de la fonction reconnue par ReSyn Assistant. Chaque bloc de nom b identifié dans l'équation est ensuite représenté par un des trois attributs b_d, b_c et b_s selon l'état détruit, créé ou stable qui lui est associé. Chaque graphe moléculaire est décrit par l'ensemble des attributs, ou motif, des blocs présents dans le graphe. Le motif de la réaction est égal à l'union des motifs de chacun de ses réactants et produits. Ainsi l'équation de la figure 3.17 est représentée par le motif :

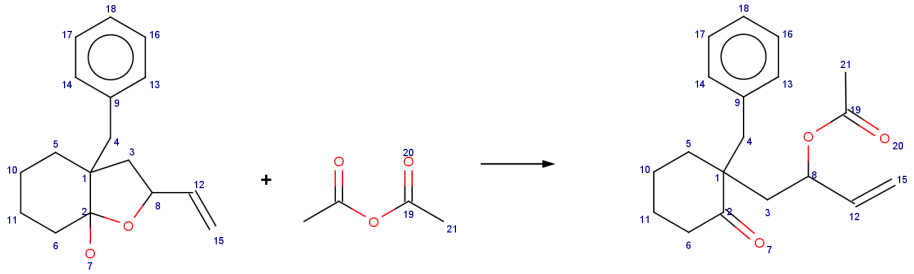
$$\textbf{hémiacétal}_d, \textbf{anhydride}_d, \textbf{aryl}_s, \textbf{alcène}_s, \textbf{carbonyle}_c, \textbf{ester}_c$$

Un algorithme de recherche des motifs fréquents et d'extraction de règles d'association permet ensuite de mettre en évidence un certain nombre de phénomènes, comme la sur-représentation de telle fonction créée ou encore de mettre en relation certaines fonctions. Une règle de confiance élevée permet par exemple d'exprimer le fait que telle fonction se créée souvent à partir de la destruction de telles autres fonctions, à l'image de la règle suivante :

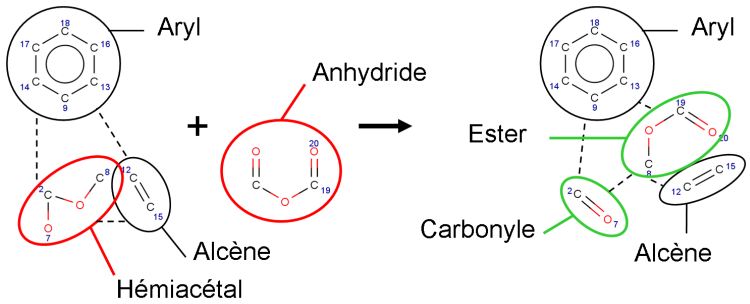
$$\textbf{alcool}_d, \textbf{anhydride}_d \rightarrow \textbf{ester}_c$$

L'extraction des motifs et des règles fréquentes permet d'apprendre sur le contenu des BdR au niveau symbolique et de permettre une indexation sémantique de celle-ci. Cependant, comme le constate S. Berasaluce (cf pages 189 et 190 de Berasaluce (2002)), les résultats restent limités car l'information topologique que renferment les graphes moléculaires est en grande partie perdue : les blocs sont certes identifiables à des graphes mais les motifs de graphes existant à l'intérieur de ces blocs ou entre ces blocs ne sont pas pris en compte. D'autre part, les résultats issus de la fouille de données dépendent entièrement de la hiérarchie des blocs passée en paramètre, c'est à dire d'une forme de connaissances a priori. Si l'intégration de connaissances peut être vue comme un atout, elle peut aussi biaiser les résultats en imposant une projection particulière des équations de réaction et ce faisant, peut occulter des phénomènes révélateurs. En particulier il est impossible à partir d'une règle d'association représentant une méthode de synthèse d'en déduire son schéma de réaction caractéristique. Par exemple le motif :

$$\textbf{alcène}_d, \textbf{diène}_d, \textbf{alcène}_c$$



(a) Équation initiale



(b) Équation de blocs fonctionnels

FIG. 3.17: Décomposition d'une équation en blocs fonctionnels. Les blocs inchangés, créés et détruits apparaissent respectivement en noir, vert et rouge sur la figure.

obtenu à partir de la décomposition en blocs fonctionnels du schéma général de la méthode de Diels-Alder (cf figure 3.11(a)) ne permet absolument pas de reconstruire ce schéma, ni même d'en donner une ébauche.

En pratique, cette approche permet davantage de mieux connaître la réactivité des groupes fonctionnels (notamment les relations de préséance entre groupes : quel groupe réagit prioritairement sur tel autre ?) que d'extraire des connaissances sur les méthodes de synthèse. Pour ce faire la prise en compte des graphes moléculaires au niveau micro, c'est-à-dire au niveau des atomes et des liaisons sans réduction aucune de l'information, semble nécessaire.

Cette conclusion est à l'origine des travaux réalisés dans le cadre de ce mémoire : dans quelle mesure les méthodes de recherche de graphes fréquents et plus généralement de fouille de graphes peuvent-elles aider à extraire des connaissances des bases de données de réactions, de la même manière que ces méthodes ont pu être appliquées à la fouille de graphes moléculaires (cf section 3.3.1). Les réactions étant des instances de méthodes de synthèse qui sont elles-mêmes représentées par certains schémas de réactions (cf section 3.1.3), la notion de schémas réactionnels est ici centrale. La première question à se poser est de savoir s'il est possible d'extraire l'ensemble des schémas de réactions fréquents – selon une définition de la fréquence à préciser – grâce aux méthodes existantes de recherche de sous-graphes fréquents. La seconde est de savoir comment déterminer les schémas qui représentent le mieux les méthodes de synthèse. Ces deux questions ouvertes font l'objet des deux chapitres suivants.

Chapitre 4

La recherche des schémas de réactions fréquents et le prétraitement des bases de données de réactions

Sommaire

4.1 Introduction

L'application qui sous-tend ce chapitre 4 comme les chapitres 5 et 6, est l'extraction à partir de BdR, de connaissances relatives aux méthodes de synthèses. Pour ce faire, le présent chapitre traite du problème de la recherche des schémas de réactions fréquents dans les BdR et du prétraitement des données associé. Comme l'a expliqué la section 3.1.3, les méthodes de synthèse jouent un rôle central en synthèse organique. La connaissance d'un expert en synthèse organique se jauge ainsi au nombre de méthodes de synthèse qu'il connaît et pour chacune de ces méthodes, la précision avec laquelle il maîtrise les conditions nécessaires à son application. Cette expertise ne se réduit cependant pas à un catalogue de méthodes : idéalement, l'expert doit avoir une compréhension globale intégrant les relations et interactions entre ces méthodes,

notamment pour prédire les phénomènes de concurrence entre réactions (i.e. une réaction se produisant à la place de celle escomptée). L'organisation des connaissances en méthodologie de synthèse est donc un problème extrêmement complexe et il est illusoire de penser pouvoir extraire automatiquement toute cette connaissance des BdR à l'aide d'un simple algorithme de fouille de données. Même en imaginant disposer un jour d'un ordinateur aussi puissant et intuitif que le cerveau humain, on voit mal comment un tel ordinateur pourrait acquérir le savoir d'un expert uniquement à partir d'un ensemble de réactions, aussi grand soit-il, sans avoir préalablement bénéficié d'une solide formation en chimie organique. Qui plus est, les BdR actuelles sont imparfaites et n'apportent qu'une information partielle (la variété des réactions est immense et seule une infime partie de cette diversité peut se représenter dans une BdR), imprécise (la description des réactions est limitée) et parfois même erronée (certaines spécifications de réactions comportent des erreurs). Enfin les bases de données sont par nature déclaratives et donc statiques : l'ordinateur ne peut contrairement à l'expert, lancer une expérimentation pour répondre à une interrogation à laquelle les données disponibles ne sont d'aucune utilité.

D'un autre côté, les ordinateurs ont des atouts que l'être humain n'a pas : un simple ordinateur de bureau pourra bientôt mémoriser et interroger une collection d'un million de réactions sans commettre la moindre erreur ou négligence, ni éprouver un quelconque sentiment de lassitude. Un système d'extraction de connaissances à partir de BdR doit donc en premier lieu être pensé comme un outil de fouille de données efficace, destiné à assister l'expert dans les tâches qui lui sont pénibles ou inaccessibles, avant éventuellement, d'évoluer vers un système plus complexe capable d'extraire des BdR et de mettre à jour un modèle intégrant les différentes connaissances relatives aux réactions chimiques. Un tel projet, qui dépasse largement le cadre de cette thèse, pourrait se développer en résolvant consécutivement les problèmes suivants :

1. Identifier automatiquement les schémas de réactions génériques à partir de très grandes bases de données de réactions, rattacher ces schémas génériques à des méthodes de synthèse, et ainsi regrouper les réactions associées à une même méthode.
2. À partir de ces regroupements de réactions, améliorer la connaissance de chacune de ces méthodes de synthèses, en particulier extraire le domaine d'application de la méthode de synthèse (en terme de catalyseurs, solvants ...).
3. À partir de ces méthodes caractérisées isolément, proposer un modèle global permettant d'organiser de façon intégrée les connaissances relatives à toutes les méthodes. Éventuellement exploiter ce modèle pour prédire le cours d'une réaction ou proposer une aide à la résolution de problèmes de synthèse.

Les travaux présentés dans ce chapitre (et dans les deux chapitres suivants) n'apportent des éléments de réponse qu'au premier problème. Ces travaux traitent plus précisément de l'extraction de schémas génériques caractéristiques de méthodes de synthèse (cf section 3.1.3) à partir de BdR.

Le concept de schéma de réaction joue donc un rôle central. Étant donné un tel schéma et une BdR, le nombre de réactions de la BdR qui contiennent ce schéma équivaut à la notion de fréquence des motifs d'attributs (cf section 2.1.2), et à ce titre définit la fréquence d'un schéma de réaction, dont on donnera une définition rigoureuse dans la section suivante. La fréquence d'un schéma est une information essentielle qui permet de savoir si la transformation exprimée par ce schéma est couramment réalisée et a fortiori réalisable, ou au contraire rare car originale ou difficile à mettre en œuvre, ou enfin inexistante car tout simplement infaisable. L'interprétation qui peut être faite de la fréquence d'un schéma de réaction est toutefois

délicate dans la mesure où elle dépend étroitement de la façon dont les données fouillées ont été confectionnées. Un schéma de fréquence élevée dans une BdR ne traduit pas forcément une transformation chimique inévitable mais peut-être simplement une pratique très courante en synthèse organique. Au contraire un schéma de réaction bien connu des chimistes peut très bien être absent d'un ensemble de réactions spécifiques à une application.

Quoiqu'il en soit, il est évident que les schémas de réactions les plus fréquents ne sont pas les plus représentatifs de méthode de synthèse. La figure 4.1 donne l'exemple d'un schéma très général et très fréquent qui n'est pas celui d'une méthode de synthèse. Ce schéma exprime

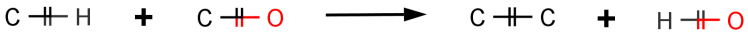


FIG. 4.1: Exemple de schéma de réaction non représentatif d'une méthode de synthèse

simplement le fait que deux liaisons C-H et C-O se brisent pour former une liaison carbone carbone – ou liaison CC – en recombinant l'hydrogène restant avec l'atome de d'oxygène. Mais ce schéma n'explique pas comment cette transformation est rendue possible. En particulier ce schéma omet les groupes fonctionnels qui se situent à proximité des atomes figurant sur le schéma et qui jouent en faveur de la transformation. Même si la fréquence d'un schéma de réaction ne répond pas directement au problème posé, elle demeure un facteur essentiel à prendre en compte : un schéma de fréquence faible, qui n'apparaît presque jamais dans une BdR aura peu de chance d'être celui d'une méthode de synthèse pour peu que cette base de données soit suffisamment grande pour être représentative de la plupart des méthodes de synthèse connues. À partir de la fréquence des schémas de réactions fréquents et de leur fréquence, il devient envisageable de déterminer les schémas des méthodes de synthèse grâce à une procédure de sélection de motifs. Ce problème est développé aux chapitres 5 et 6. L'objet de ce chapitre est à ce stade du développement, d'extraire les schémas de réactions fréquents dans une BdR, non pas comme un objectif en soi mais comme le préalable à une sélection efficace des motifs dont le problème est abordé ultérieurement au chapitre 5.

Les schémas de réactions fréquents se présentent sous la forme de deux termes constitués de graphes. Il est donc naturel de se tourner vers les algorithmes existants de recherche de sous-graphes fréquents (cf section 2.4), par ailleurs très efficaces. Deux problèmes empêchent toutefois l'application directe de ces algorithmes aux BdR :

- D'une part la manière dont les chimistes représentent les réactions par des graphes ne permet pas aux techniques de fouille de graphes d'extraire les schémas de réactions fréquents. Si ces méthodes peuvent s'appliquer immédiatement et avec succès à la fouille de graphes moléculaires (Fischer et Meinl, 2004), leur application directe aux graphes moléculaires contenus dans les BdR ne conduirait à aucun résultat pertinent : tout au plus pourrait-on mettre en évidence les fragments de graphes moléculaires qui sont fréquents dans les produits ou dans les réactants de réactions sans qu'aucun schéma de réaction, c'est à dire, aucun schéma de transformation entre graphes moléculaires, ne puisse s'en déduire.
- D'autre part les bases de données contiennent des descriptions de réactions souvent incomplètes, ambiguës ou erronées. Leur fouille sans autre filtrage conduirait à des résultats trop bruités et donc inexploitables.

Une étape de prétraitement s'avère donc indispensable d'une part, pour améliorer la qualité des données fouillées et d'autre part, pour exprimer les données au sein d'un modèle

exploitable par les algorithmes de fouille de graphes. Le reste du chapitre présente comment certains principes de chimie organique ont permis de concevoir un prétraitement des BdR qui satisfasse aux deux objectifs précédents. À cette fin, la section 4.2 définit formellement une représentation informatique des graphes moléculaires, des schémas de réactions et de la relation d'inclusion entre ces schémas. La section 4.3 explicite un certain nombre de propriétés des molécules et des réactions chimiques qui sont utilisées ultérieurement lors du prétraitement des BdR. La section 4.4 présente le processus global de fouille de schémas de réactions tel qu'il a été mis au point. La section 4.5 introduit le modèle des graphes condensés de réactions, au cœur du processus de prétraitement. La section 4.6 présente ensuite les imperfections que comportent les BdR ainsi que les différentes étapes du prétraitement qu'il a été nécessaire de mettre au point pour corriger ces défauts. Enfin la section 4.7 détaille les tests qui ont été réalisés pour d'une part, évaluer l'efficacité du prétraitement et pour d'autre part, connaître les limites de la recherche des schémas de réactions fréquents. Ces travaux sont résumés par ailleurs dans l'article Pennerath *et al.* (2008c).

4.2 Formalisation du problème

Les notions introduites au chapitre 1 de graphes moléculaires, d'équations chimiques ou de schémas de réactions sont définies formellement dans ce qui suit. Ces définitions sont très proches des représentations informatiques qui ont été faites de ces objets pour les manipuler.

Définition 4.2.1. Le *graphe moléculaire* d'une molécule est un graphe g simple, étiqueté et connexe[33] dont les ensembles de sommets $V(g)$ et d'arêtes $E(g)$ représentent respectivement les atomes et les liaisons de covalence de la molécule. Chaque sommet $s \in V(g)$ représente un atome et est étiqueté par le couple $(e(s), c(s))$ où $e(s)$ et $c(s)$ sont respectivement le numéro atomique de l'élément chimique ($H = 1$ pour l'hydrogène, $C = 6$ pour le carbone, $N = 7$ pour l'azote, $O = 8$ pour l'oxygène ...) et la charge électrique (en multiple entier de la charge élémentaire, de -3 à +3) portés par l'atome. Chaque arête $a \in E(g)$ est étiquetée par la multiplicité $m(a)$ de la liaison associée (1,2,3 ou 4 pour une liaison respectivement simple, double, triple ou aromatique).

Les BdR décrivent principalement chaque réaction par son équation chimique (cf section 3.2.3).

Définition 4.2.2. Une *équation chimique* ou plus généralement un *schéma de réaction* est spécifié par un 4-uplet $(\mathcal{R}, \mathcal{P}, \lambda_\mathcal{R}, \lambda_\mathcal{P})$, tel que :

- $\mathcal{R}$ et $\mathcal{P}$ sont les graphes représentant respectivement les membres gauche et droit de l'équation ou du schéma de réaction. Les composantes connexes de $\mathcal{R}$ (resp. $\mathcal{P}$) correspondent aux graphes moléculaires des différents réactants (resp. produits).
- $\lambda_\mathcal{R} : \mathcal{D}_{\lambda_\mathcal{R}} \subseteq V(\mathcal{R}) \to \mathbb{N}$ et $\lambda_\mathcal{P} : \mathcal{D}_{\lambda_\mathcal{P}} \subseteq V(\mathcal{P}) \to \mathbb{N}$ sont des fonctions d'annotation des sommets de $\mathcal{R}$ et de $\mathcal{P}$ respectivement, associant à certains sommets un *indice d'appariement*.

La figure 4.2 représente une équation chimique. En règle générale, on ne s'intéresse qu'aux schémas de réactions ou aux équations qui expriment une transformation, c'est à dire, pour lesquels il existe une différence entre les membres gauche et droit. Dans le cas contraire, on

[33] Un graphe moléculaire est nécessairement connexe dans la mesure où les molécules considérées forment des assemblages d'atomes dont la cohésion physique est assurée exclusivement par des liaisons de covalence, représentées par ailleurs par les arêtes des graphes moléculaires.

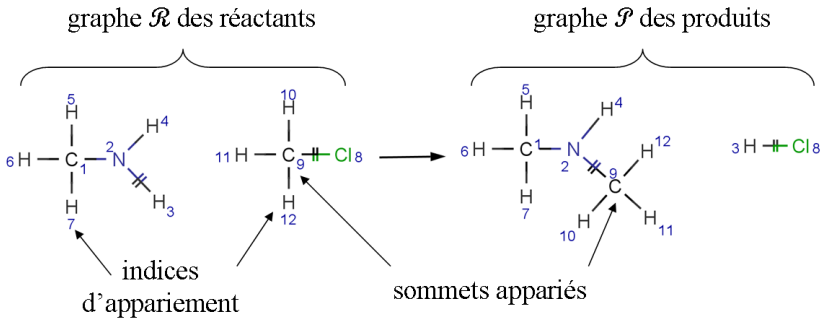


FIG. 4.2: Exemple de spécification d'une équation de réaction.

qualifiera le schéma ou l'équation de *dégénéré*. Un sommet s de $\mathcal{R}$ (resp. $\mathcal{P}$) est *apparié* si $\lambda_{\mathcal{R}}$ (resp. $\lambda_{\mathcal{P}}$) lui associe un indice d'appariement. Dans une équation telle que celle représentée sur la figure 4.2, les indices d'appariement sont les entiers qui jouxtent les atomes appariés. Deux sommets $s_1 \in \mathcal{R}$ et $s_2 \in \mathcal{P}$ sont *appariés* l'un à l'autre s'ils sont annotés par le même indice (i.e. $\lambda_{\mathcal{R}}(s_1) = \lambda_{\mathcal{P}}(s_2)$). Ils sont alors censés représenter un et un seul même atome. Une *base de données de réactions chimiques* (BdR) est donc formellement équivalente à un ensemble de 4-uplets $\{(\mathcal{R}_i, \mathcal{P}_i, \lambda_{\mathcal{R}_i}, \lambda_{\mathcal{P}_i}\})_{1 \leq i \leq n}$.

Un schéma de réaction n'est ici qu'une structure syntaxique qui permet aussi bien de représenter l'équation chimique d'une véritable réaction que le schéma d'une transformation conceptuelle, commune à plusieurs réactions. Le schéma de la figure 4.3 est ainsi un des nombreux schémas généraux contenus dans l'équation de la figure 4.2. Cette relation d'inclusion

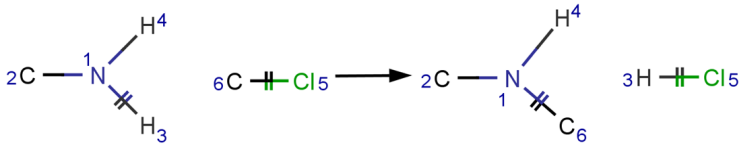


FIG. 4.3: Exemple d'un schéma de réaction inclus dans l'équation de la figure 4.2.

se définit formellement :

Définition 4.2.3. Le schéma de réaction $S_1 = (\mathcal{R}_1, \mathcal{P}_1, \lambda_{\mathcal{R}_1}, \lambda_{\mathcal{P}_1})$ est inclus (ou est plus général que) l'équation ou le schéma de réaction $S_2 = (\mathcal{R}_2, \mathcal{P}_2, \lambda_{\mathcal{R}_2}, \lambda_{\mathcal{P}_2})$ (et on note $S_1 \subseteq_S S_2$) si :

- Les graphes $\mathcal{R}_1$ et $\mathcal{P}_1$ sont isomorphes à des sous-graphes partiels[34] de $\mathcal{R}_2$ et $\mathcal{P}_2$.
- Les injections correspondantes de sommets $\theta_R : V(\mathcal{R}_1) \to V(\mathcal{R}_2)$ et $\theta_P : V(\mathcal{P}_1) \to V(\mathcal{P}_2)$ sont compatibles avec les relations d'appariement :

$$\theta_R(\mathcal{D}_{\lambda_{\mathcal{R}_1}}) \subseteq \mathcal{D}_{\lambda_{\mathcal{R}_2}}, \theta_P(\mathcal{D}_{\lambda_{\mathcal{P}_1}}) \subseteq \mathcal{D}_{\lambda_{\mathcal{P}_2}} \text{ et } \lambda_{\mathcal{R}_1}(s_1) = \lambda_{\mathcal{P}_1}(s_2) \Leftrightarrow \lambda_{\mathcal{R}_2}(\theta_R(s_1)) = \lambda_{\mathcal{P}_2}(\theta_P(s_2))$$

[34]Le choix de considérer des sous-graphes partiels plutôt que des sous-graphes induits permet de couvrir davantage de schémas, les sous-graphes induits formant un sous-ensemble des sous-graphes partiels.

L'inclusion entre schémas de réactions peut se comprendre comme une relation de subsomption $\sqsubseteq_S = _S \supseteq$ entre les schémas de réactions définis à un isomorphisme près (i.e. modulo la numérotation des indices de sommets). Cette relation de subsomption définit la fréquence d'un schéma de réaction : le *support d'un schéma de réaction* S relativement à un ensemble E de réactions est le nombre de réactions de E généralisées par S. Le *problème de la recherche des schémas de réactions fréquents* dans une BdR consiste à déterminer le support de tous les schémas de réactions fréquents dont le support est supérieur ou égal à un seuil f_{min}.

4.3 Formalisation des connaissances du domaine

Les équations contenues dans les BdR comportent souvent des lacunes : ainsi les chimistes représentent systématiquement les molécules sous une forme abrégée (cf formule structurale de la figure 3.3(b)) et prennent rarement la peine de décrire tous les appariements d'atomes entre sommets. Les produits secondaires jugés inintéressants sont tout simplement omis de l'équation et il en va de même de certains réactants lorsque ceux-ci sont jugés évidents. Ces négligences sont tolérées dans la mesure où les chimistes n'ont aucune difficulté à réinterpréter correctement les données l'aide des connaissances qu'ils ont du domaine. Dans le cadre d'un processus automatisé d'extraction de connaissance, il devient nécessaire d'identifier les propriétés particulières que présentent les graphes $(\mathcal{R}, \mathcal{P})$ et que le chimiste utilise implicitement. La démarche adoptée consiste à reformuler ces propriétés en axiomes exprimés exclusivement à partir de concepts propres à l'informatique et à la théorie des graphes de manière à ce que les algorithmes de prétraitement puissent être spécifiés formellement :

Conservation des atomes Tout atome se conservant au cours d'une réaction, il existe une bijection ν des sommets du graphe $\mathcal{P}$ vers ceux de $\mathcal{R}$ qui identifie un et même atome. Cela implique en particulier que les fonctions $\lambda_\mathcal{R}$ et $\lambda_\mathcal{R}$ soient injectives et que $\lambda_\mathcal{R}$ et $\lambda_\mathcal{P}$ soient cohérentes avec ν :

$$\forall s_1 \in V(\mathcal{R}), \forall s_2 \in V(\mathcal{P}), \lambda_\mathcal{R}(s_1) = \lambda_\mathcal{P}(s_2) \Rightarrow \nu(s_2) = s_1 \tag{4.1}$$

Règle de valence La règle de valence (ou règle de l'octet) impose que le nombre d'électrons d'un atome s qui participent aux liaisons de s est fonction uniquement de l'élément chimique de s et sa charge électrique[35]. Ce nombre d'électrons est la *valence* de l'atome s. A titre d'exemples, dans le cas d'atomes non chargés électriquement, les valences d'un atome d'hydrogène, d'oxygène, d'azote et de carbone sont respectivement de $\text{val}(H = 1) = 1$, $\text{val}(O = 8) = 2$, $\text{val}(N = 7) = 3$ et $\text{val}(C = 6) = 4$. Certains éléments chimiques peuvent admettre plusieurs états de valence : un atome de soufre peut avoir une valence de 2, 4 ou 6. Dans le cas d'un atome chargé, d'élément chimique e et de charge c, sa valence devient $\text{val}(e - c)$. Ainsi un atome de carbone chargé négativement aura une valence de $\text{val}(C^-) = \text{val}(6 - (-1)) = \text{val}(N) = 3$.

Si on omet ici le cas plus complexe des liaisons aromatiques, la règle de valence entraîne une propriété caractéristique des graphes moléculaires : le degré pondéré $\deg_p$ de tout atome (i.e. la somme des multiplicités $m(a)$ des liaisons a simples, doubles ou triples

[35] À des fins de simplification, on omet ici de mentionner le cas des radicaux libres instables qui sont du reste, relativement rares dans les BdR.

incidentes à s) égale sa valence et est donc borné :

$$\forall s \in V(\mathcal{R}), \deg_p(s) = \sum_{a \in E(\mathcal{R}) | s \in a} m(a) = \text{val}(e(s) - c(s)) \tag{4.2}$$

$$\forall s \in V(\mathcal{P}), \deg_p(s) = \sum_{a \in E(\mathcal{P}) | s \in a} m(a) = \text{val}(e(s) - c(s)) \tag{4.3}$$

La valence d'un atome correspond donc au nombre maximal de liaisons simples que peut avoir cet atome. L'égalité précédente se généralise au cas des cycles aromatiques si on admet qu'une liaison aromatique présente dans la formule précédente une multiplicité $m(a, s)$ variable, fonction du type du cycle aromatique et de l'atome incident s : ainsi dans un cycle aromatique à 6 carbones, un atome de carbone engage un électron supplémentaire en plus des deux électrons participant déjà aux deux liaisons du cycle incidente à s. La multiplicité par liaison aromatique est alors de $m(a) = (2+1)/2 = 1,5$. Dans un cycle aromatique à 4 carbones et un atome d'oxygène, ce dernier engage un doublet libre non comptabilisé dans sa valence, de sorte que la multiplicité des liaisons aromatiques reste égale à $m(a) = (2 + 0)/2 = 1$.

Conservation des électrons de valence Les *électrons de valence* sont les électrons de la dernière couche électronique d'un atome s. Ces électrons participent soit aux liaisons de s, soit à des doublets d'électrons locals à s, selon une répartition qui satisfasse la règle de valence. Au cours d'une réaction, non seulement les liaisons entre atomes peuvent se modifier mais aussi les doublets d'électrons peuvent se scinder ou au contraire se former pour modifier la valence d'un atome (par exemple le doublet d'un atome d'azote N de valence 3 peut céder un de ses électrons de valence à un autre atome pour donner naissance à un atome d'ammonium N^+ de valence 4). Quel que soit le devenir des électrons de valence des atomes participant à une réaction, leur nombre total doit se conserver au cours de la réaction tout comme les atomes d'un même élément chimique. Il en résulte, si $d(s)$ désigne le nombre de doublets d'électrons de l'atome s, l'égalité suivante :

$$\sum_{s \in V(\mathcal{R})} 2 \times d(s) + 2 \times \sum_{a \in E(\mathcal{R})} m(a) = \sum_{s \in V(\mathcal{P})} 2 \times d(s) + 2 \times \sum_{a \in E(\mathcal{P})} m(a) \tag{4.4}$$

En posant pour toute paire $\{s_1; s_2\}$ de sommets de $\mathcal{R}$ (resp. de $\mathcal{P}$), $m(\{s_1; s_2\}) = 0$ si $\{s_1; s_2\} \notin E(\mathcal{R})$ (resp. si $\{s_1; s_2\} \notin E(\mathcal{P})$) et en rappelant la bijection $\nu : \mathcal{P} \to \mathcal{R}$ due à la conservation des atomes, il vient :

$$\sum_{\{s_1;s_2\} \subseteq V(\mathcal{R})} m(\{\nu^{-1}(s_1); \nu^{-1}(s_2)\}) - m(\{s_1; s_2\}) = \sum_{s \in V(\mathcal{R})} d(s) - d(\nu^{-1}(s)) \tag{4.5}$$

Dans la plupart des réactions, le nombre de doublets d'électron se conserve au cours des réactions de sorte que le membre droit de l'équation précédente s'annule.

Minimalité de la distance d'édition Une réaction transforme ses réactants en ses produits en suivant statistiquement la séquence (t_j) de transformations élémentaires qui minimise l'énergie thermodynamique nécessaire. Cette énergie est proportionnelle en première approximation à la distance d'édition $d(\mathcal{R}, \mathcal{P}) = \sum c(t_j)$ pour passer de $\mathcal{R}$ à $\mathcal{P}$ en supposant que le coût $c(t_j)$ soit l'énergie nécessaire à la transformation t_j. D'après l'axiome précédent, les transformations élémentaires consistent uniquement à diminuer ou augmenter la multiplicité des arêtes (cf membre gauche de l'équation 4.5)

en plus de modifier le nombre de doublets d'électron (cf membre droit). Par ailleurs si les réactants et les produits d'une réaction sont connues, le membre droit de l'équation 4.5 est un terme calculable puisque les modifications des doublets d'électrons se déduisent directement de la structure de $\mathcal{R}$ et $\mathcal{P}$. Le terme variable de la distance d'édition s'identifie donc à $d(\mathcal{R},\mathcal{P}) = \sum_{\{s_1;s_2\}\subseteq V(\mathcal{R})} c(\{s_1;s_2\})$ où le coût $c(\{s_1;s_2\})$ correspond à l'énergie nécessaire pour modifier la multiplicité de la liaison $\{s_1;s_2\}$. En notant $r(\{s_1;s_2\}) = m(\{\nu^{-1}(s_1);\nu^{-1}(s_2)\}) - m(\{s_1;s_2\})$ la variation de multiplicité de la liaison $\{s_1;s_2\}$, le coût $c(\{s_1;s_2\})$ est nul si $r(\{s_1;s_2\}) \geq 0$ puisque la formation d'une liaison libère de l'énergie, et peut être supposé proportionnel en première approximation à $|r(\{s_1;s_2\})|$ lorsque $r(\{s_1;s_2\}) < 0$. Finalement :

$$d(\mathcal{R},\mathcal{P}) = \min_r \left(\sum_{a\in E(\mathcal{R}),r(a)<0} |r(a)| \right) \tag{4.6}$$

Cette équation exprime que pour passer des graphes moléculaires des réactants aux graphes des produits qui sont supposés connus, la réaction sous-jacente tend à briser le moins possible de liaisons.

Ces différentes propriétés sont exploitées lors de la phase de prétraitement du processus de fouille des schémas de réactions présenté dans la section suivante.

4.4 Le processus de fouille des schémas de réactions

La figure 4.4 présente les différentes étapes du processus de fouille de schémas de réactions fréquents tel qu'il a été développé. Ce processus entièrement opérationnel comprend toutes les étapes du processus d'extraction de connaissances tel que décrit par Fayyad *et al.* (1996b) : l'expert commence tout d'abord par sélectionner les réactions qu'il souhaite fouiller dans une BdR à l'aide d'un langage de requêtes spécifique puis sauvegarde la réponse de sa requête dans un fichier au format *Reaction Data File* ou RDF[36]. L'étape de prétraitement développée en section 4.6, filtre et corrige l'ensemble $\{E_i\}$ des équations de départ en un ensemble $\{E'_j\}$ d'équations à demi ou totalement appariées. Cet ensemble peut alors être transformé en l'ensemble des graphes de réactions équivalents $\{\mathcal{G}(E'_j)\}$ dont le modèle est développé en section 4.5. Les descriptions de ces graphes étiquetés sont sauvegardées dans un fichier au format GBDM afin d'être exploités par un algorithme de fouille de graphes, comme `Gaston` (Nijssen et Kok, 2004) ou `gSpan` (Yan et Han, 2002) pour la recherche des sous-graphes fréquents ou `Forage` (Pennerath et Napoli, 2007) pour l'extraction des schémas de réactions les plus informatifs (cf chapitre 5). Dans tous les cas, l'algorithme produit un fichier GBDM contenant un ensemble $\{(g_k, f_k, \dots)\}$ de motifs g_k associés à leur fréquence f_k plus éventuellement d'autres propriétés (score, information, etc). L'étape de post-traitement permet de convertir l'ensemble des graphes de réactions g_k en l'ensemble $\{S(g_k)\}$ des schémas de réactions équivalents, qui sont ensuite triés selon les valeurs décroissantes d'une des propriétés spécifiée par l'expert (par exemple le score ou la fréquence), avant d'être sauvegardés dans un fichier RDF. L'expert peut alors analyser les schémas obtenus et leurs propriétés à l'aide d'un logiciel standard de visualisation de schémas de réactions. Avant de décrire les étapes du prétraitement dans le détail, la section suivante développe le modèle des graphes condensés

[36]Ce format de fichier est un des formats les plus répandus pour l'échange de descriptions de réactions. Il n'est aucunement relié au langage *Resource Description Framework* du Web sémantique.

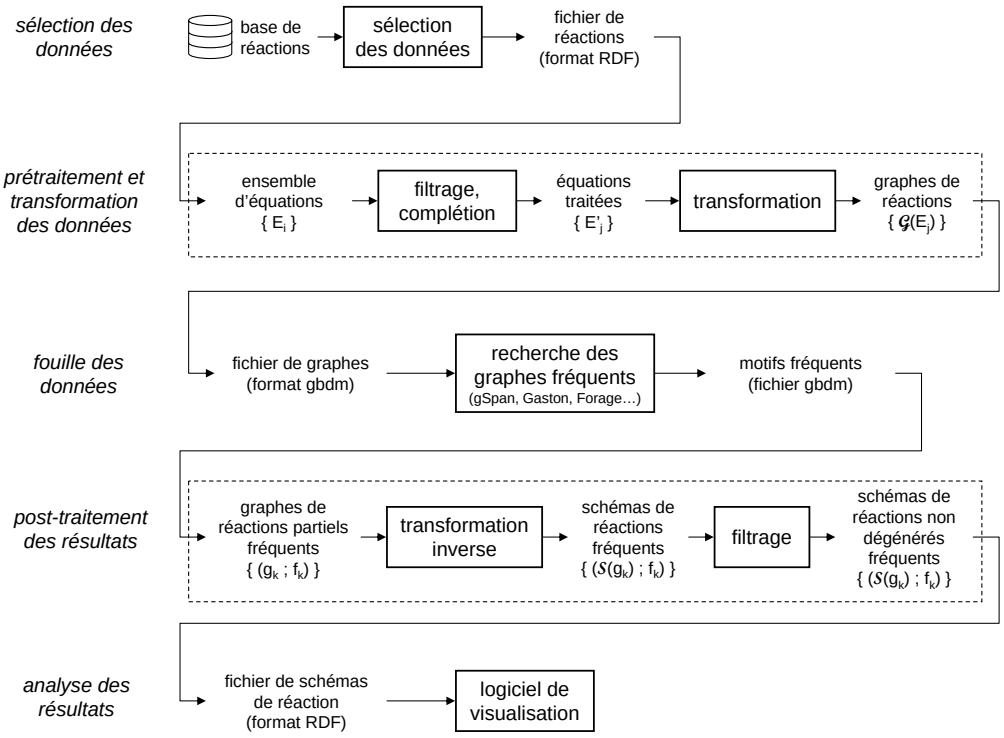


FIG. 4.4: Étapes du processus de fouille des schémas de réactions.

de réactions qui permet de traiter la recherche de schémas réactionnels fréquents comme un problème de recherche des sous-graphes fréquents connexes.

4.5 La transformation des données : les graphes condensés de réactions

Tout algorithme de recherche des motifs fréquents exploite la propriété de monotonicité de la relation de subsomption entre motifs qui correspond en l'occurrence à la relation d'inclusion $\subseteq_S$. Les méthodes existantes de fouille de graphes sont incapables de s'adapter à cette relation d'inclusion $\subseteq_S$ pour deux raisons essentielles : d'une part ces méthodes ne génèrent, pour des raisons de réduction combinatoire, que des motifs de graphes connexes, ce qui n'est pas le cas des schémas de réactions. D'autre part ces méthodes n'intègrent pas naturellement la relation d'ordre $\subseteq_S$ entre schéma de réactions puisque cette relation repose sur la notion étrangère d'appariement entre sommets (i.e. les fonctions $\lambda_{\mathcal{R}}$ et $\lambda_{\mathcal{P}}$). Le modèle des *graphes condensés de réactions* (GCR), ou simplement graphes de réactions, permet de résoudre simultanément ces deux problèmes (Pennerath et Napoli, 2006) et ainsi, de traiter le problème de la recherche de schémas réactionnels fréquents comme un problème de recherche des sous-graphes fréquents connexes. Le graphe condensé de réaction d'un schéma de réaction repose sur le principe déjà évoqué de *conservation des atomes* puisqu'il revient à superposer les graphes des réactants et des produits d'un schéma de réaction, en fusionnant les sommets appariés représentant le

même atome. Les GCR ont initialement été introduits par Vladutz (1986) afin de stocker les réactions dans un format plus compact et ont été adoptés par des méthodes d'apprentissage par classification pour représenter des exemples de réactions (Wilcox et Levinson, 1986; Fujita, 1986; Jauffret *et al.*, 1995). Au delà de leur compacité, les GCR détiennent des propriétés qui en font un modèle idéal dans le cadre de la fouille de schémas de réactions. La suite de cette section développe un modèle formel pour les graphes de réactions et met en rapport les propriétés du modèle avec le problème de la recherche des schémas de réactions fréquents. Ce modèle est présenté ici dans le cadre restrictif des schémas de réactions, même s'il peut être utilisé de manière plus générale, pour fouiller des transformations abstraites de graphes.

Le graphe de réaction associé à une équation ne peut se définir qu'à partir d'une équation $(\mathcal{R}, \mathcal{P})$ dite *à demi appariée* c'est à dire dont tout sommet de $\mathcal{P}$ est apparié à un et un seul sommet de $\mathcal{R}$ (i.e. l'application $\nu = \lambda_{\mathcal{R}}^{-1} \circ \lambda_{\mathcal{P}}$ de $V(\mathcal{P})$ vers $V(\mathcal{R})$ est injective). Une équation à demi appariée devient *totalement appariée* si tous les sommets du membre gauche et du membre de droite sont appariés (i.e. l'application ν devient bijective). Une équation qui n'est pas à demi appariée mais qui dispose d'atomes appariés est dite *partiellement appariée*. Étant donné un schéma de réaction $S = (\mathcal{R}, \mathcal{P}, \lambda_{\mathcal{R}}, \lambda_{\mathcal{P}})$ à demi apparié, le *graphe condensé de réaction* noté $\mathcal{G}(S)$ du schéma S est égal au graphe $\mathcal{R}$ dans lequel les liaisons formées par la réaction (i.e. dans $E(\mathcal{P})$ mais pas dans $E(\mathcal{R})$) ont été ajoutées en tenant compte de l'appariement des atomes et où toutes les arêtes de $\mathcal{G}(S)$ ont été étiquetées par le couple $(l_{\mathcal{R}}, l_{\mathcal{P}})$ des types de la liaison dans $\mathcal{R}$ et $\mathcal{P}$ respectivement (en utilisant un type spécial 0 pour désigner une liaison non existante dans $\mathcal{R}$ ou $\mathcal{P}$). Enfin les sommets de $\mathcal{G}(S)$ sont étiquetés par l'élément chimique de l'atome qu'il représente et ses charges électriques avant et après la réaction. Formellement :

Définition 4.5.1. Étant donné un schéma de réaction $S = (\mathcal{R}, \mathcal{P}, \lambda_{\mathcal{R}}, \lambda_{\mathcal{P}})$ à demi apparié, selon une injection $\nu : V(\mathcal{P}) \to V(\mathcal{R})$ (i.e. $\nu = \lambda_{\mathcal{R}}^{-1} \circ \lambda_{\mathcal{P}}$), le graphe condensé de réaction $\mathcal{G}(S)$ de S est égal au graphe $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}}, l_{\mathcal{G}}^v, l_{\mathcal{G}}^e)$ où :

- $V_{\mathcal{G}} = V(\mathcal{R})$
- $E_{\mathcal{G}} = E(\mathcal{R}) \cup \{\{\nu(v_1); \nu(v_2)\} | \{v_1; v_2\} \in E(\mathcal{P})\}$
- $\forall v \in V_{\mathcal{G}}$,
  - Si $v \in \{\nu(v') | v' \in V_{\mathcal{P}}\}$ alors $l_{\mathcal{G}}^v(v) = (e(v), c(v), c(\nu^{-1}(v)))$
  - Sinon $l_{\mathcal{G}}^v(v) = (e(v), c(v), c(v))$
- $\forall a \in E_{\mathcal{G}}$ (en notant $\nu^{-1}(a) = \{\nu^{-1}(v_1); \nu^{-1}(v_2)\}$ pour $a = \{v_1; v_2\} \in E_{\mathcal{G}}$),
  - Si $a \in E(\mathcal{R})$ et $\nu^{-1}(a) \in E(\mathcal{P})$ alors $l_{\mathcal{G}}^e(a) = (l_{\mathcal{R}}^e(a), l_{\mathcal{P}}^e(\nu^{-1}(a)))$
  - Si $a \in E(\mathcal{R})$ et $\nu^{-1}(a) \notin E(\mathcal{P})$ alors $l_{\mathcal{G}}^e(a) = (l_{\mathcal{R}}^e(a), 0)$
  - Si $a \notin E(\mathcal{R})$ et $\nu^{-1}(a) \in E(\mathcal{P})$ alors $l_{\mathcal{G}}^e(a) = (0, l_{\mathcal{P}}^e(\nu^{-1}(a)))$

La définition du graphe de réaction tient lieu d'algorithme de construction. Le graphe de réaction de l'équation de la figure 4.2 est ainsi représenté sur la figure 4.5. Un graphe de réaction associé à une équation chimique (resp. à un schéma de réaction général) sera qualifié de *spécifique* (resp. de *général*).

Inversement il est possible d'associer à tout graphe G dont les étiquettes sont semblables à celles des graphes de réactions définis précédemment, un schéma de réaction $\mathcal{S}(G)$ totalement apparié :

Définition 4.5.2. Étant donné un graphe connexe $G = (V_G, E_G, l_G^v, l_G^e)$ dont les étiquettes ont le type de celles des graphes de réaction, le schéma de réaction $\mathcal{S}(G)$ associé à G est le schéma $S = (\mathcal{R}, \mathcal{P}, \lambda_{\mathcal{R}}, \lambda_{\mathcal{P}})$ où

- $V(\mathcal{R}) = V(\mathcal{P}) = V_G$

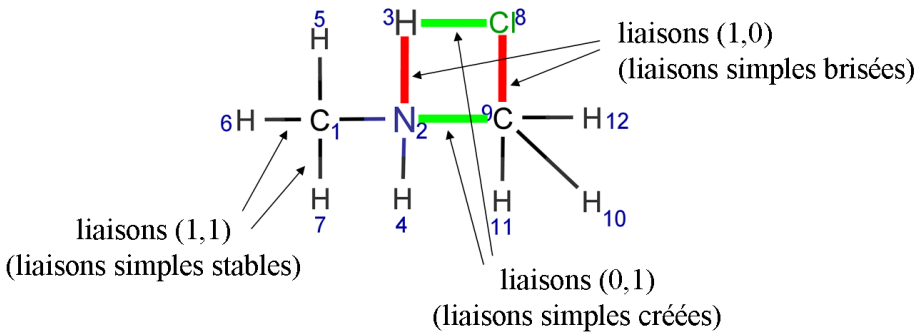


FIG. 4.5: Graphe de réaction équivalent à l'équation totalement appariée de la figure 4.2.

- $E(\mathcal{R}) = \{e \in E_G | l^e_G(e) = (l_\mathcal{R}, l_\mathcal{P}) \text{ et } l_\mathcal{R} \neq 0\}$
- $E(\mathcal{P}) = \{e \in E_G | l^e_G(e) = (l_\mathcal{R}, l_\mathcal{P}) \text{ et } l_\mathcal{P} \neq 0\}$
- $\forall v \in V_G, \lambda_\mathcal{R}(v) = \lambda_\mathcal{P}(v) =$ l'indice de v dans G.

Là encore cette définition tient lieu d'algorithme de construction du schéma $\mathcal{S}(G)$. La notion de schéma de réaction dégénéré est étendu à celle des graphes de réactions : un graphe de réaction G est dit *dégénéré* lorsque chacune de ses arêtes est telle que son type $(l_\mathcal{R}, l_\mathcal{P})$ vérifie $l_\mathcal{R} = l_\mathcal{P}$ et lorsque chaque atome conserve sa charge électrique au cours de la transformation.

Les graphes de réactions détiennent quatre propriétés utiles dans le cadre de la fouille de données :

Propriété 4.5.3. *Un graphe de réaction G est dégénéré si et seulement si le schéma $\mathcal{S}(G)$ de réaction associé l'est aussi.*

Cette propriété triviale résulte de la définition des schémas et graphes de réaction dégénérés.

Propriété 4.5.4. *La transformation $S \mapsto \mathcal{G}(S)$ qui associe à un schéma totalement apparié son graphe de réaction est bijective (modulo les indices d'appariement et les isomorphismes de graphes) et sa bijection réciproque est $G \mapsto \mathcal{S}(G)$.*

Un graphe de réaction est donc un graphe rigoureusement équivalent à un schéma de réaction totalement apparié. Dans le cas d'un schéma S de réaction à demi apparié comme illustré par l'exemple de la figure 4.6, l'obtention du schéma de réaction $\mathcal{S}(\mathcal{G}(S))$ revient à compléter le membre droit de S par certains fragments moléculaires constitués des atomes présents dans le terme des réactants mais absents dans le terme initial des produits.

Propriété 4.5.5. *Tout graphe de réaction est un graphe connexe.*

Cette propriété est une conséquence immédiate de la définition des graphes de réaction. En effet, s'il existait deux composantes connexes G_1 et G_2 dans un graphe de réaction, les schémas de réactions associés $\mathcal{S}(G_1)$ et $\mathcal{S}(G_2)$ représenteraient non pas un mais deux schémas de réactions indépendants et donc dissociables. La quatrième propriété est la plus fondamentale :

Propriété 4.5.6. *L'ensemble des schémas de réactions ordonné par la relation d'inclusion de schémas $\subseteq_S$ (cf définition 4.2.3) est équivalent (rigoureusement est isomorphe) à l'ensemble*

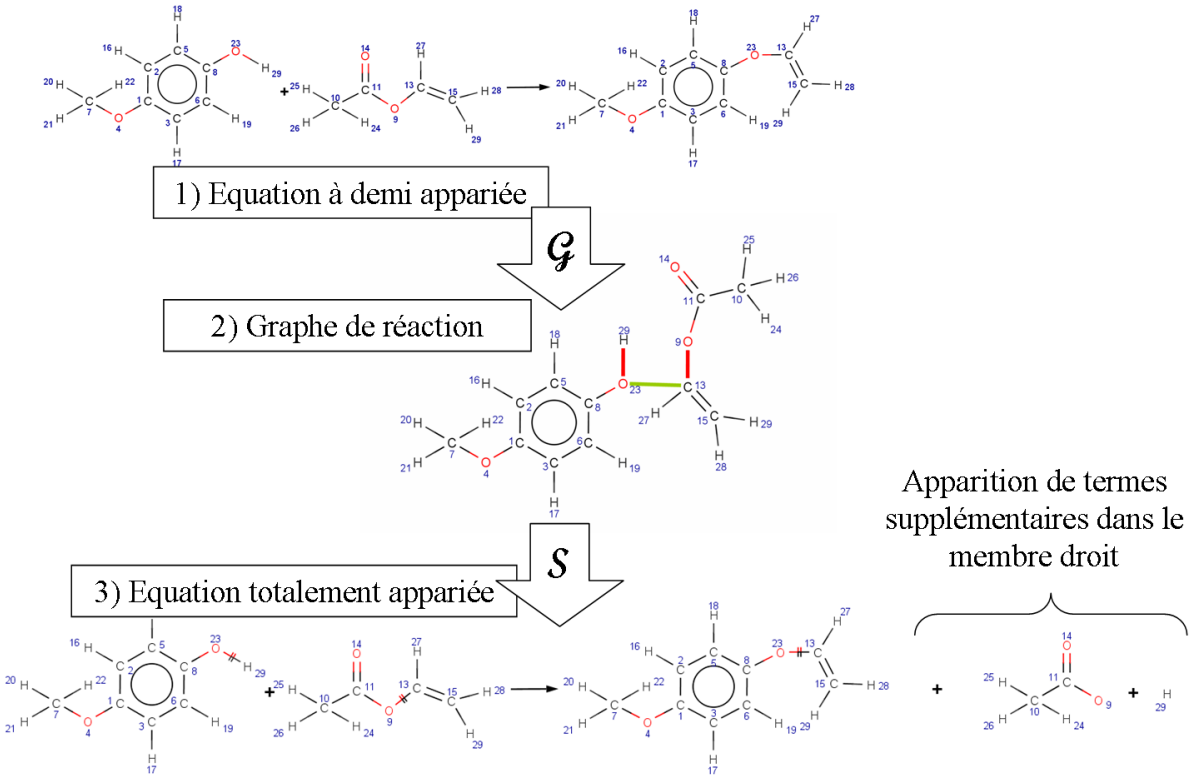


FIG. 4.6: Complétion d'une équation à demi appariée.

des graphes de réactions ordonné par la relation de sous-graphe partiel isomorphe :

$$S_1 \subseteq_S S_2 \Leftrightarrow \mathcal{G}(S_1) \subseteq_G \mathcal{G}(S_2)$$

La figure 4.7 illustre l'équivalence des deux relations d'ordre sur l'exemple du schéma de réaction de la figure 4.3 inclus dans l'équation de la figure 4.2. Étant donné un schéma S_1

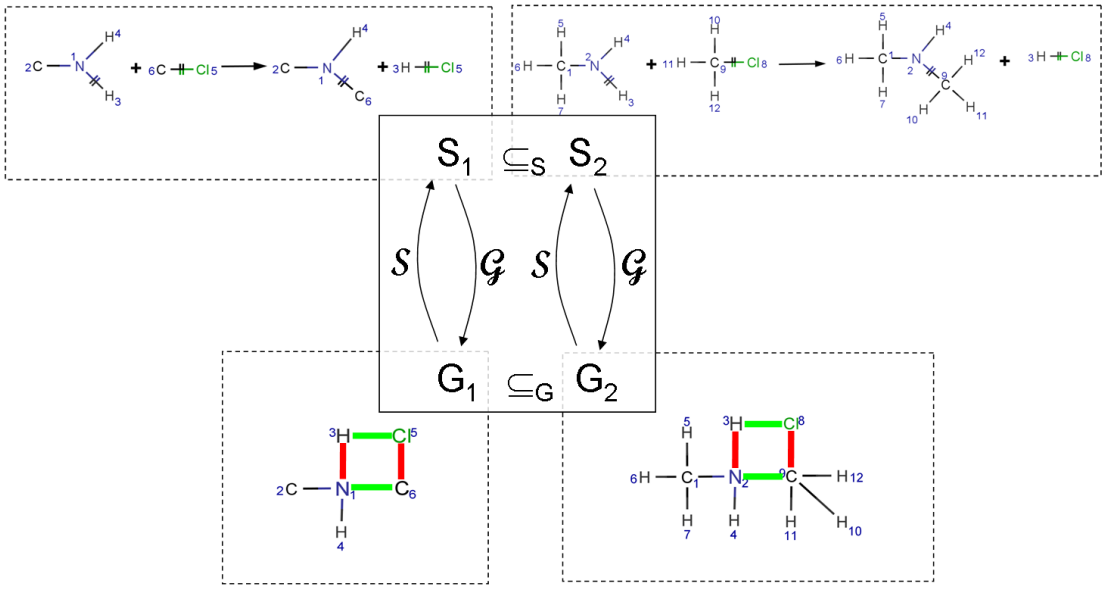


FIG. 4.7: Equivalence des relations d'ordre $\subseteq_S$ et $\subseteq_g$.

inclus au sens de $\subseteq_S$ dans un schéma S_2, le GCR G_1 du schéma S_1 apparaît bien isomorphe à un sous-graphe partiel du GCR G_2 du schéma S_2.

Proposition 4.5.7. *Le problème de la recherche des schémas de réactions non dégénérés fréquents relativement à un support minimal f_{min} dans un ensemble $(E_i)_{1\leq i\leq n}$ d'équations de réactions est équivalent à celui de la recherche des sous-graphes connexes non dégénérés fréquents relativement au même support minimal f_{min} dans l'ensemble des graphes de réactions équivalents $(\mathcal{G}(E_i))_{1\leq i\leq n}$.*

Cette proposition est une conséquence des quatre propriétés précédentes. 4.5.3, 4.5.4, 4.5.5 et 4.5.6. L'intérêt de cette proposition est d'exhiber une méthode pour traiter le problème de la recherche des schémas de réactions non dégénérés fréquents en utilisant les algorithmes existants de recherche de sous-graphes fréquents comme `gSpan` (Yan et Han, 2002) ou `Gaston` (Nijssen et Kok, 2004) (cf section 2.4). La méthode se décompose en quatre étapes :

1. Prétraitement : transformer les équations $(E_i)_{1\leq i\leq n}$ des réactions en leurs graphes de réaction équivalents $(\mathcal{G}(E_i))_{1\leq i\leq n}$
2. Fouille des données : extraire les sous-graphes $(g_j)_{1\leq j\leq m}$ connexes fréquents de $(\mathcal{G}(E_i))_{1\leq i\leq n}$ à l'aide par exemple de `Gaston`.
3. Post-traitement (filtrage) : ne conserver que les sous-graphes $(g_{j_k})_{1\leq k\leq l}$ non dégénérés des motifs fréquents $(g_j)_{1\leq j\leq m}$.

4. Post-traitement (transformation) : transformer les motifs fréquents $(g_{j_k})_{1 \leq k \leq l}$ en leurs schémas de réactions équivalents $(\mathcal{S}(g_{j_k}))_{1 \leq k \leq l}$.

Le seul inconvénient de cette méthode est de ne pouvoir éviter la génération de sous-graphes dégénérés fréquents lors de l'étape de fouille des données, du moins si cette dernière utilise les algorithmes existants de recherche de sous-graphes fréquents. Il est alors nécessaire d'éliminer les motifs dégénérés lors de la phase de post-traitement. Par ailleurs cette méthode n'est applicable que si les équations chimiques initiales sont à demi appariées et sans autre défaut. La section suivante explique comment se ramener à un tel ensemble d'équations à partir d'un extrait quelconque de BdR.

4.6 Le prétraitement des données

4.6.1 Les imperfections des bases de données de réactions

Une BdR décrit une équation chimique par un 4-uplet $(\mathcal{R}, \mathcal{P}, \lambda_{\mathcal{R}}, \lambda_{\mathcal{P}})$. Ce 4-uplet respecte très rarement tous les axiomes de la section 4.3. Les équations telles que celles des figures 4.9 et 4.11 peuvent être qualifiées différemment selon la nature de leurs non conformités (Berasaluce, 2002). Les défauts ont pu ainsi être répartis en différentes catégories. Une équation est ainsi qualifiée de :

Non saturée quand les atomes d'hydrogène ne sont pas explicités et que le principe de la valence n'est pas respecté (i.e. quand $\{s \in V(\mathcal{R}) \cup V(\mathcal{P}) | deg_p(s) < \text{val}(l(s))\} \neq \emptyset$). En pratique c'est le cas de toutes les équations chimiques contenues dans les bases de données, comme celle de la figure 4.8.

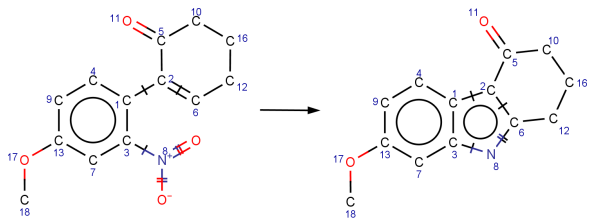


FIG. 4.8: Exemple d'équation non saturée.

Non déterministe quand les produits d'une équation ne sont pas produits simultanément mais concurremment selon des rendements statistiques respectifs (i.e. quand il existe au moins deux sommets s_1 et s_2 de deux composantes connexes distinctes de $\mathcal{P}$ portant le même indice d'appariement $\lambda_{\mathcal{P}}(s_1) = \lambda_{\mathcal{P}}(s_2)$, cf figure 4.9). Cette convention oblige à omettre les produits secondaires de chaque réaction concurrente : une réaction non déterministe est donc nécessairement incomplète (cf point suivant).

Non équilibrée mais équilibrable quand certains réactants ou produits doivent être dupliqués pour que le principe de conservation des atomes et donc de leurs éléments chimiques puisse être respecté. Ainsi sur l'équation de la figure 4.10 n'est pas équilibrée (il y a par exemple deux atomes de chlore dans le produit alors qu'il y en a qu'un seul dans le réactant). L'équation est toutefois équilibrable puisque le produit est de toute évidence constitué de deux exemplaires du même réactant. Formellement soit la matrice $H_{\mathcal{R}}$

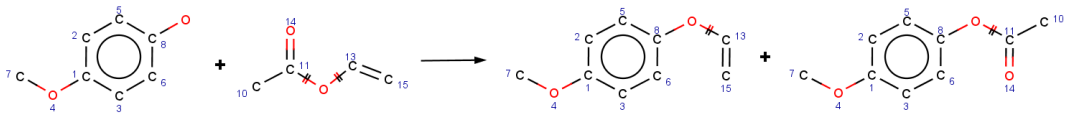


FIG. 4.9: Exemple d'équation non déterministe.

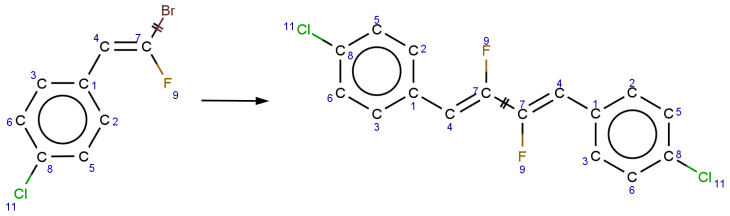


FIG. 4.10: Exemple d'équation non équilibrée.

(resp. $H_{\mathcal{P}}$) qui a pour coefficients h_{ij} les nombres d'atomes de numéro atomique i dans le j$^{\text{ème}}$ réactant $\mathcal{R}_j$ (resp. j$^{\text{ème}}$ produit $\mathcal{P}_j$)). La réaction est *équilibrée* si les sommes ligne par ligne des coefficients de $H_{\mathcal{R}}$ et $H_{\mathcal{P}}$ sont égales. Elle est dite *équilibrable* s'il existe des vecteurs de pondération $C_{\mathcal{R}}$ et $C_{\mathcal{P}}$ dont les coefficients sont des entiers strictement positifs satisfaisant l'équation linéaire :

$$H_{\mathcal{R}} \times C_{\mathcal{R}} = H_{\mathcal{P}} \times C_{\mathcal{P}}$$

En pratique il est rare qu'une équation soit équilibrable exactement : sur l'exemple de la figure 4.10, même en dupliquant le réactant, les deux atomes de brome (Br) manquent dans le membre droit de l'équation.

Erronée quand l'équation n'est pas équilibrable et que certains éléments chimiques sont plus présents dans les produits que dans les réactants (i.e. quand il existe un élément chimique davantage présent dans $\mathcal{P}$ que dans $\mathcal{R}$). Dans certains cas, certaines réactions erronées le sont à un point qui laisse perplexe, tel que l'exemple de la figure 4.11(a). Mais dans la plupart des cas, les lacunes sont plus subtiles : l'équation de la figure 4.11(b) est ainsi erronée du fait qu'elle n'est pas équilibrable et contient respectivement 2 et 3 atomes d'oxygène dans les membres gauche et droit.

Incomplète quand l'équation n'est ni équilibrable ni erronée parce que certains produits secondaires sont omis de l'équation (i.e. quand il existe une élément chimique davantage présent dans $\mathcal{R}$ que dans $\mathcal{P}$). Ainsi l'équation de la figure 4.12) est incomplète car elle n'est ni équilibrable, ni erronée et l'atome de silicium et ses trois carbones voisins ne se retrouvent pas dans le membre droit de l'équation.

Ambiguë quand l'équation n'est pas à demi appariée parce que certains sommets des produits ne sont pas appariés (i.e. quand $\mathcal{D}_{\lambda_{\mathcal{P}}} \neq \mathcal{P}$). En effet les appariements des atomes ne sont généralement pas spécifiés par les chimistes. Les équations des BdR sont alors

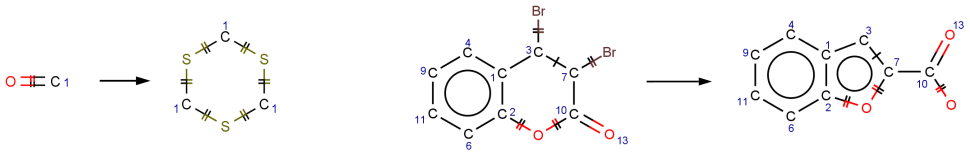


FIG. 4.11: Deux exemples d'équations erronées.

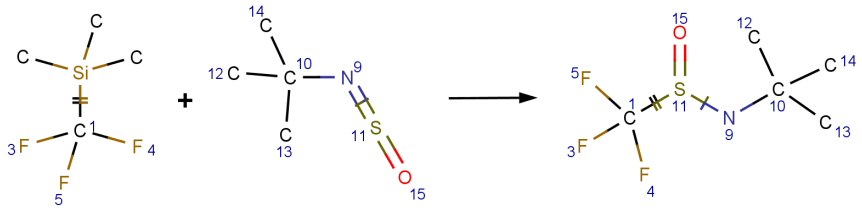


FIG. 4.12: Exemple d'équation incomplète.

appariées automatiquement par un algorithme fondé sur une minimisation de la distance d'édition entre les graphes $\mathcal{R}$ et $\mathcal{P}$. Dans certaines configurations, l'algorithme est incapable de déterminer l'origine de certains atomes des produits dans le membre des réactants. Ainsi sur l'équation de la figure 4.13, l'atome d'oxygène aromatique du membre droit n'est pas apparié car il peut aussi bien provenir du premier comme du second réactant.

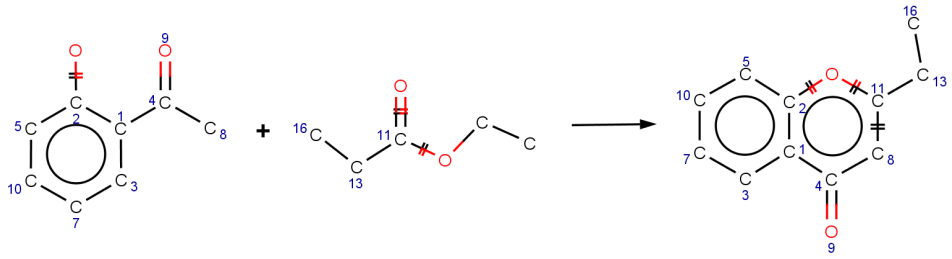


FIG. 4.13: Exemple d'équation ambiguë.

4.6.2 Les étapes du prétraitement

Le prétraitement des données est construit comme une succession d'étapes. Chaque étape est une fonction qui transforme tout 4-uplet qu'elle reçoit d'un flux en entrée en une suite éventuellement vide de 4-uplets qu'elle émet dans un flux en sortie. Tout 4-uplet en sortie de e_i devient ensuite un 4-uplet en entrée de e_{i+1}. L'ordre des étapes est défini de telle sorte qu'une étape résout une catégorie particulière de défauts sans jamais introduire un type de défaut résolu lors d'une étape précédente. Les étapes sont dans l'ordre :

La scission des équations non déterministes : si l'équation $(\mathcal{R}, \mathcal{P}, \lambda_{\mathcal{R}}, \lambda_{\mathcal{P}})$ est non déterministe, toute composante connexe $\mathcal{P}_k$ de $\mathcal{P}$ dont le rendement associé dépasse un seuil configurable donne lieu à une équation déterministe $(\mathcal{R}, \mathcal{P}_k, \lambda_{\mathcal{R}}, \lambda_{\mathcal{P}_k})$ où $\lambda_{\mathcal{P}_k}$ est la restriction de $\lambda_{\mathcal{P}}$ sur $\mathcal{P}_k$.

La saturation des molécules qui consiste simplement à connecter à chaque sommet s de $\mathcal{R}$ et de $\mathcal{P}$, $\text{val}(e(s)) - c(s)) - \deg_p(s)$ atomes d'hydrogène par des liaisons simples pour satisfaire l'axiome de *valence des atomes*. La saturation de l'équation chimique de la figure 4.8 produit l'équation de la figure 4.14.

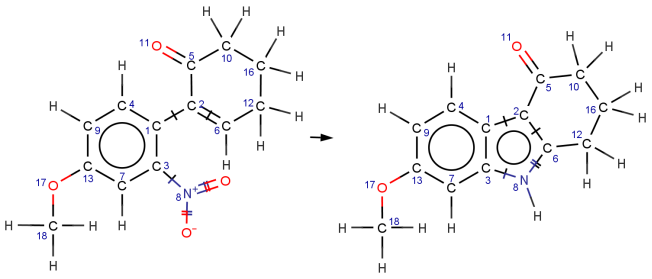


FIG. 4.14: Saturation d'une équation chimique.

L'élimination des équations erronées : s'il existe un sommet de $\mathcal{P}$ dont l'élément chimique n'est associé à aucun sommet de $\mathcal{R}$, l'équation candidate est erronée et est éliminée. C'est par exemple le cas des atomes de soufre dans l'équation de la figure 4.11(a).

La pondération des équations non équilibrées est un problème théorique complexe de programmation linéaire en nombres entiers (Sena *et al.*, 2006). En pratique toutefois les équations comptent rarement plus de trois réactants et trois produits. L'étape de pondération se contente donc pour toute équation non équilibrée de tester toutes les duplications évidentes de réactants et/ou de produits jusqu'à obtenir une équation équilibrée ou presque. Une réaction « presque équilibrée » est une réaction pour laquelle seulement quelques atomes sont manquants dans le membre des produits. L'équation résultante après équilibrage est alors incomplète. La détection des équations « presque équilibrées » se fait en vérifiant que les coefficients de $H_{\mathcal{R}} \times C_{\mathcal{R}}$ sont égaux ou légèrement plus grands que ceux de $H_{\mathcal{P}} \times C_{\mathcal{P}}$. L'unique réactant de la figure 4.10 est ainsi dupliqué pour donner l'équation « presque équilibrée » de la figure 4.15 (cette équation n'est pas rigoureusement équilibrée puisqu'il manque les 2 atomes de brome dans le membre droit). Si toutes les tentatives de pondération échouent, on distingue deux cas : s'il existe une étiquette de sommet plus représentée dans $\mathcal{P}$ que dans $\mathcal{R}$, l'équation candidate est considérée erronée et est éliminée. Dans le cas contraire l'équation est jugée incomplète mais peut passer à l'étape suivante.

La complétion des appariements : A ce stade du prétraitement, l'équation obtenue est presque toujours ambiguë (ne serait-ce du fait que les atomes d'hydrogène ajoutés par saturation ne sont pas appariés). Un rejet systématique des équations ambiguës est donc impossible. La solution adoptée consiste à produire l'ensemble des équations totalement appariées les plus plausibles qui se déduisent de l'équation ambiguë. Cet ensemble contient vraisemblablement l'unique réaction se produisant réellement en plus d'éventuelles équations factices. L'effet néfaste introduit par la présence de ces artefacts est toutefois limité car les équations surnuméraires ainsi produites restent minoritaires

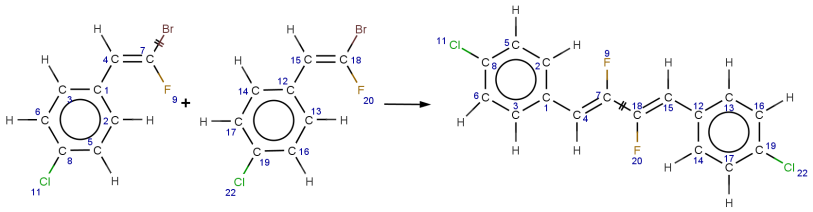


FIG. 4.15: Équilibrage de l'équation de la figure 4.10.

(les tests présentés à la section 4.7.1 montrent que ces réactions représentent moins de 9 % des équations fouillées) et ainsi, influent peu sur le filtrage statistique qu'opère la recherche des motifs fréquents en ne gardant que les schémas de réactions les plus fréquents.

Pour désambiguïser une équation, il est nécessaire d'apparier tous les sommets non appariés de $\mathcal{P}$ à des sommets non appariés de $\mathcal{R}$. Compte tenu du nombre exponentiel d'appariements possibles, il est nécessaire de restreindre la procédure aux appariements les plus plausibles. On introduit à cette fin la notion de compatibilité entre sommets : deux sommets $s_1 \in V(\mathcal{P})$ et $s_2 \in V(\mathcal{R})$ sont *compatibles* si aucun des deux n'est déjà apparié, s'ils sont de même élément chimique (i.e. $e(s_1) = e(s_2)$), s'ils sont tous deux adjacents à un sommet apparié et s'ils partagent le même ensemble maximal de sommets voisins appariés : s_1 est incompatible avec s_2 s'il existe un sommet $s_3 \in V(\mathcal{R})$ compatible avec s_1 qui a plus de voisins appariés avec des voisins de s_1 que s_2 n'en a avec s_1. En notant $\mathcal{V}(s)$ l'ensemble des sommets voisins du sommet s, cette dernière condition équivaut à :

$$|\lambda_{\mathcal{R}}(\mathcal{V}(s_3) \cap \mathcal{D}_{\lambda_{\mathcal{R}}}) \cap \lambda_{\mathcal{P}}(\mathcal{V}(s_1) \cap \mathcal{D}_{\lambda_{\mathcal{P}}})| > |\lambda_{\mathcal{R}}(\mathcal{V}(s_2) \cap \mathcal{D}_{\lambda_{\mathcal{R}}}) \cap \lambda_{\mathcal{P}}(\mathcal{V}(s_1) \cap \mathcal{D}_{\lambda_{\mathcal{P}}})|$$

L'appariement de deux sommets n'est plausible que si ces derniers sont compatibles, selon le principe de *minimalité de la distance d'édition*. L'élimination des appariements incompatibles permet d'élaguer l'arbre de recherche dans l'espace d'état des appariements possibles. La procédure consiste alors en un algorithme de backtracking qui effectue à chaque étape de sa progression l'appariement d'un sommet de $\mathcal{P}$ non encore apparié avec un sommet de $\mathcal{R}$ qui lui est compatible puis met à jour les fonctions $\lambda_{\mathcal{R}}$ et $\lambda_{\mathcal{P}}$. Un retour arrière se produit soit dès qu'un sommet non apparié de $\mathcal{P}$ n'est plus compatible avec aucun sommet de $\mathcal{R}$ soit dès que tout sommet de $\mathcal{P}$ est apparié, l'équation associée étant alors passée à l'étape de traitement suivante. La complétion des appariements d'une de deux équations déterministes extraites de l'équation 4.9 conduit à 4 équations à demi appariées dont l'une d'elles est représentée sur la figure 4.16. On y observe l'appariement des atomes d'hydrogène mais surtout de l'atome d'oxygène numéro 23.

La construction du graphe de réaction : A ce stade du prétraitement, toutes les équations sont à demi appariées. Chaque équation E est donc remplacée par son graphe de réaction $\mathcal{G}(E)$, conformément à la section 4.5. Le graphe de réaction construit à partir de l'équation à demi appariée de la figure 4.16 est représenté sur la figure 4.17.

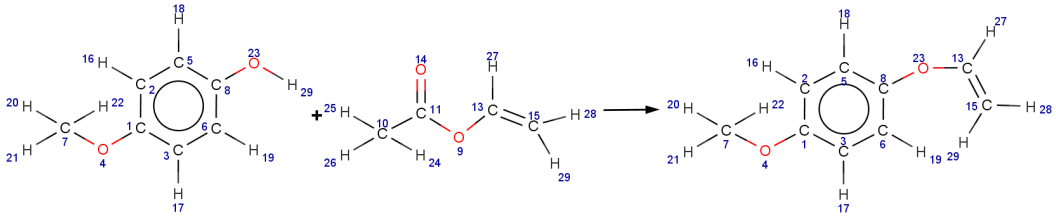


FIG. 4.16: Équation appariée à partir de l'équation de la figure 4.9.

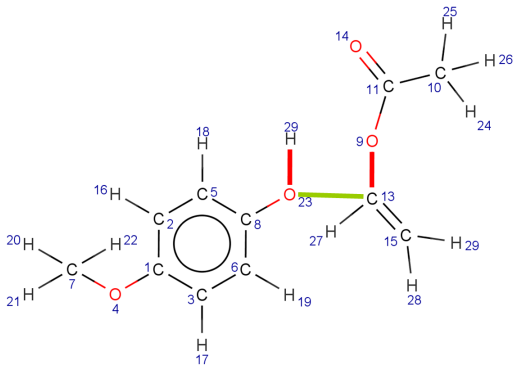


FIG. 4.17: Graphe de réaction de la figure 4.16

L'élimination des graphes de réactions non réalistes : L'étape de complétion des appariements transforme une équation de départ en un nombre limité d'équations E_i à demi appariées qui n'ont pas nécessairement le même degré de plausibilité. De ce fait, le nombre n_i d'arêtes brisées (étiquetées $-$) de chaque graphe de réaction $\mathcal{G}(E_i)$ est calculé. Tout graphe $\mathcal{G}(E_i)$ dont le nombre n_i n'est pas égal à $n_{min} = \min(n_i)$ peut alors être éliminé au nom du principe de *minimalité de la distance d'édition*. Après élimination, toute équation initiale E aboutit à un ensemble d'équations à demi appariées de même n_i minimal. Le nombre de ces équations est le plus souvent égal à 1, parfois nul lorsque E s'avère erronée, parfois égal à 2 ou 3 (mais rarement plus) lorsque E peut valablement être interprétée selon différents appariements de sommets. Lorsque le nombre d'appariements possibles est supérieur à un seuil (fixé à 4), on estime que l'appariement de l'équation initiale est insuffisant et que les équations qui en découlent sont trop incertaines et doivent être éliminées.

La complétion des arêtes créées : Dans le cas d'une réaction E incomplète, les sous-graphes présents initialement dans $\mathcal{R}$ mais pas dans $\mathcal{P}$ impliquent la présence de produits secondaires dans $\mathcal{S}(\mathcal{G}(E))$. Certains atomes de ces produits secondaires ne sont pas saturés du fait que certaines liaisons formées manquent. Dans le cas particulier où seuls deux sommets du graphe produit ne sont pas saturés, il est possible de compléter le graphe de réaction par la liaison formée manquante. Ce faisant, l'équation équivalente du graphe de réaction fait apparaître le produit secondaire. Ainsi alors que l'équation équivalente du graphe de réaction de la figure 4.17 fait apparaître au bas de la figure 4.6, deux fragments de produits dans le membre de droite (en plus du produit principal), la

complétion des arêtes créées sur la figure 4.18, a pour effet de lier ces deux fragments résiduels et ainsi de faire apparaître le produit secondaire de la réaction.

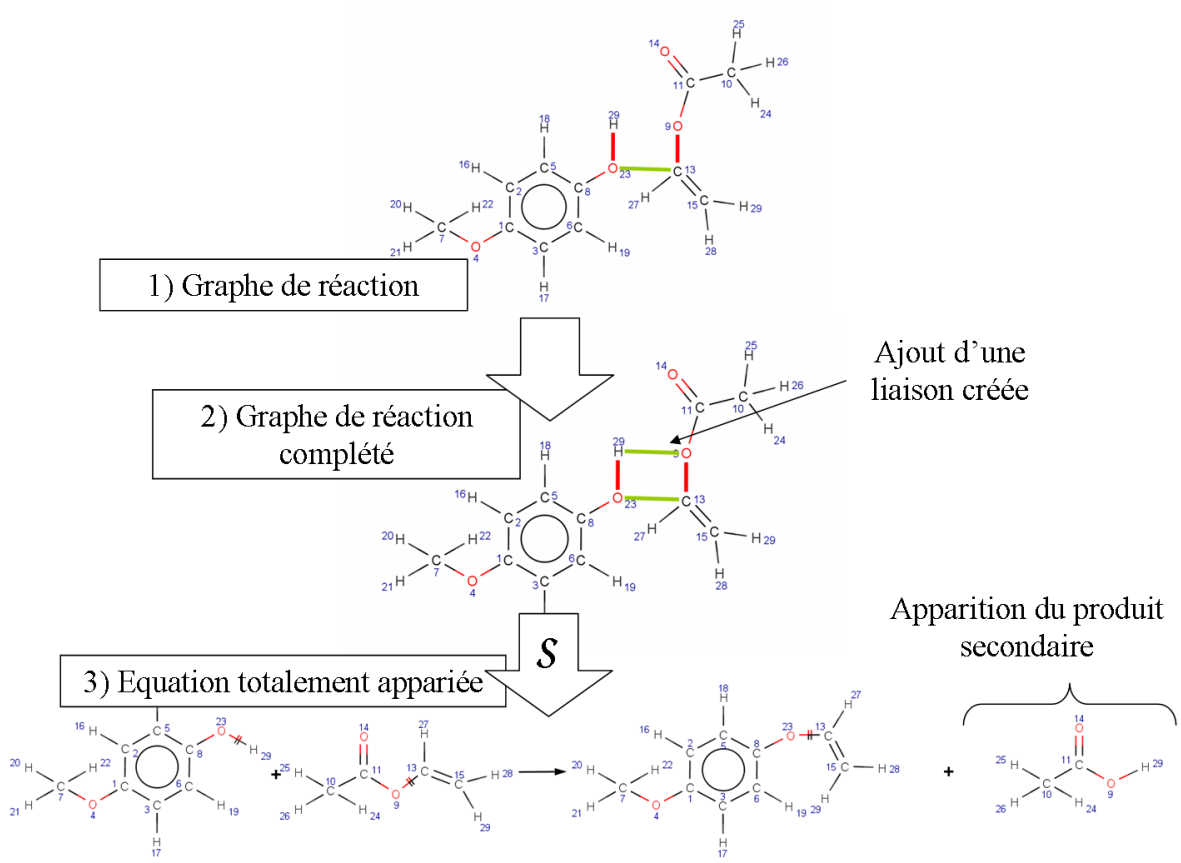


FIG. 4.18: Apparition du produit secondaire après complétion du graphe de réaction

En résumé, la mise au point du prétraitement a permis de filtrer, compléter et convertir les équations chimiques contenues dans les BdR en GCR. Les sous-graphes connexes fréquents ont ensuite pu être extraits de cet ensemble de GCR à l'aide d'algorithmes existants de fouille de graphes. Ces motifs fréquents sont ensuite été départagés en motifs dégénérés et non dégénérés. Les motifs dégénérés permettent de considérer l'ensemble des sous-graphes connexes fréquents, inclus dans les graphes moléculaires des réactants et produits des BdR. Les motifs non dégénérés permettent, une fois reconvertis en schémas réactionnels, de considérer l'ensemble des schémas fréquents (non dégénérés) inclus dans les équations des BdR. Cette procédure a donné lieu à plusieurs expérimentations présentées dans la section suivante.

4.7 Expérimentation

Les tests réalisés se répartissent selon deux objectifs : évaluer la qualité du prétraitement et des données d'une part (section 4.7.1) et tester la recherche des schémas de réactions fréquents d'autre part (section 4.7.2).

4.7.1 Tests relatifs au prétraitement

L'algorithme de prétraitement a été implémenté et testé sur plus de 125000 réactions mono-étapes issues de l'intégralité des deux BdR Orgsyn et Derwent Journal of Synthetic Methods (ou JSM) ainsi que sur un extrait d'environ 30000 entrées récentes de la base ChemInform. Ces bases de données couvrent ensemble une grande variété de méthodes de synthèse. Les résultats des tests sont résumés par le tableau de la figure 4.19.

base	taille initiale	taux t_p de perte	taux t_c de complétude	taux t_a d'ambiguïté	taux t_e d'échec	taux t_g de conversion	taille en sortie
Orgsyn	4856	45 %	13 %	11 %	2 %	54 %	2608
ChemInform	34027	17 %	20 %	8 %	11 %	73 %	25889
JSM	86246	28 %	21 %	10 %	4 %	70 %	61176
Total	125129	25 %	20 %	9 %	6 %	70 %	89673

Fig. 4.19: Résultats du prétraitement sur les BdR Orgsyn et JSM.

Le contenu de ce tableau repose sur la définition de différents taux indicateurs :

Taux de perte Soit $\mathcal{I}$ l'ensemble initial des équations de la base de données. Le prétraitement commence par écarter l'ensemble $\mathcal{E} \subseteq \mathcal{I}$ des équations erronées, comme les réactions non équilibrables ou comportant des représentations non normalisées (i.e. pour mettre en évidence des catalyseurs, des complexes organo-métalliques, etc). Le *taux de perte* des données est défini comme la proportion $t_p = \frac{|\mathcal{E}|}{|\mathcal{I}|}$ des données comportant des défauts non corrigibles. La plupart des réactions rejetées l'étant parce que certains réactants manquent et rendent l'équation non équilibrable, le taux de perte permet d'apprécier la précision avec laquelle les réactants sont spécifiés.

Taux de complétude L'ensemble $\mathcal{I} \setminus \mathcal{E}$ des équations non erronées conduit à un ensemble d'équations déterministes $\mathcal{D}$. Si l'ensemble $\mathcal{C}$ désigne le sous-ensemble de $\mathcal{D}$ regroupant les équations complètes, le *taux de complétude* des données est défini comme $t_c = \frac{|\mathcal{C}|}{|\mathcal{D}|}$. Ce taux permet d'apprécier la précision avec laquelle les produits secondaires ont été spécifiés.

Taux d'échec La procédure d'appariement appliquée à $\mathcal{D}$ ne réussit que sur un sous-ensemble $\mathcal{A}$ de $\mathcal{D}$. Le *taux d'échec* du prétraitement est défini comme la proportion $t_e = 1 - \frac{|\mathcal{A}|}{|\mathcal{D}|}$.

Taux d'ambiguïté En fonction de la précision des appariements initiaux dans $\mathcal{I}$, chaque équation de $\mathcal{D}$ alimente l'ensemble résultat $\mathcal{G}$ d'un nombre variable de graphes de réactions. Le *taux d'ambiguïté* des données est défini par $t_a = \frac{|\mathcal{G}|}{|\mathcal{D}|} - 1$. Globalement 92 % des équations correctement converties (i.e. de $\mathcal{A}$) conduisent à un seul graphe de réaction, 7 % à deux, 0,5 % à 3 et 0,3 % à 4. Le taux d'ambiguïté et le taux d'échec permettent d'apprécier la précision avec laquelle les indices d'appariement ont été spécifiés.

Taux de conversion Le taux global de conversion est le produit $t_g = (1 - t_p) \times (1 - t_e)$. Ce taux permet d'apprécier la proportion des données qui sont finalement fouillés.

Si le taux moyen de conversion de 70 % peut paraître relativement faible, un examen approfondi atteste que la plupart des échecs sont souvent inévitables : en effet, soit les équations fautives introduisent des atomes d'origine inconnue dans leur produit, soit elles recourent à des représentations non normalisées (i.e. pour mettre en évidence des catalyseurs, des complexes organo-métalliques, etc), soit enfin elles présentent des appariements de sommets très pauvres voire inexistants.

De façon inattendue mais néanmoins intéressante, l'outil de prétraitement a permis d'évaluer précisément le soin apporté à la spécification des données dans les BdR. La courbe de la figure 4.20 représente ainsi l'évolution année après année des indicateurs de qualité des différents suppléments annuels de la base JSM entre 1980 et 2006. Comme en témoigne l'aspect

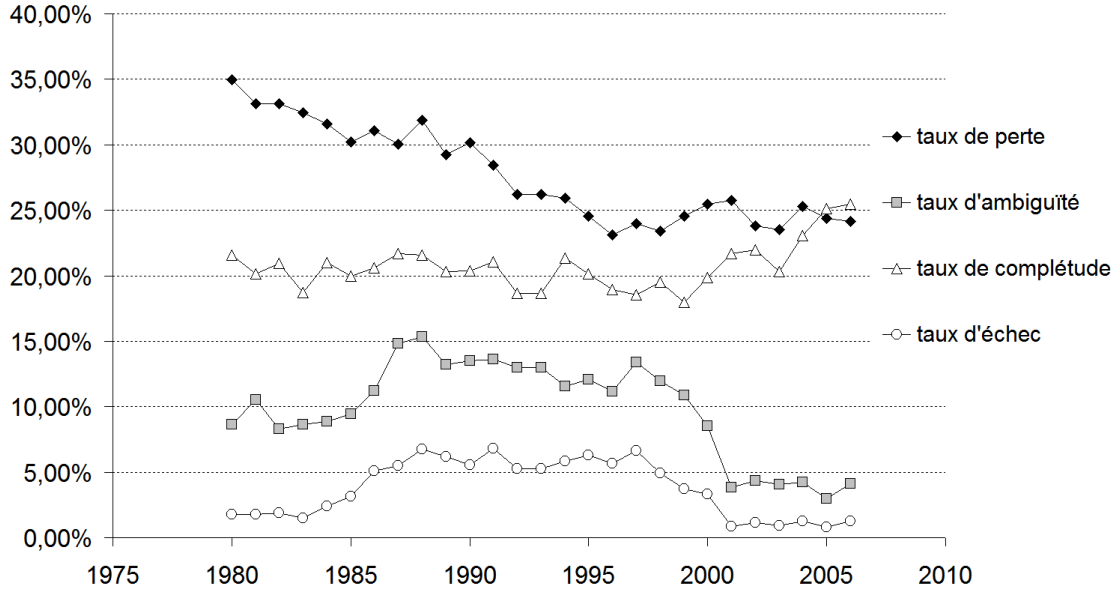


FIG. 4.20: Évolution des indicateurs de qualité sur la base JSM.

relativement lissé des courbes, la production annuelle de JSM de 2500 réactions en moyenne est suffisamment représentative pour établir des tendances. Le taux de perte t_p a ainsi observé une diminution régulière depuis la création de JSM avant de se stabiliser autour de 25 %, manifestement due à une spécification de plus en plus rigoureuse des réactants. Le taux de complétude t_c longtemps resté constant, connaît une hausse légère et récente. Cette amélioration est confortée par la décroissance brutale du taux d'ambiguïté t_a sur la même période. Ces deux évolutions traduisent conjointement une nette augmentation de la qualité des données, en particulier pour ce qui concerne les appariements d'atomes. Enfin les évolutions du taux d'échec et du taux d'ambiguïté apparaissent logiquement corrélées dans la mesure où ces deux taux apprécient la qualité d'appariement des atomes. Au delà de l'estimation de la qualité des données dans les BdR, le prétraitement a surtout permis d'extraire les schémas de réactions fréquents qui fait l'objet de la section suivante.

4.7.2 Tests sur la recherche de schémas de réactions fréquents

Le processus de fouille des BdR exposé à la section 4.4 a été appliqué au problème de la recherche des schémas de réactions fréquents afin d'une part, de vérifier qu'en pratique, une quantité significative de schémas de réactions fréquents peut être extrait en un temps raisonnable, de différentes BdR variant par leur taille et contenu et d'autre part, d'estimer la distribution statistique des schémas dans les BdR. Ce dernier point permet de mettre en évidence le besoin d'une procédure sélective de schémas informatifs présentée au chapitre suivant. Les résultats présentés reposent sur trois jeux de données aux caractéristiques différentes :

1. Le premier jeu de donnée est constitué de l'intégralité des 2608 réactions issues du prétraitement de la base de données ORGSYN. Cette base de données est connue pour être une description rigoureuse de méthodes de synthèses soigneusement sélectionnées et où les protocoles expérimentaux ont été vérifiés par des laboratoires indépendants. Ce jeu de données présente donc une grande variété de méthodes de synthèse sans (ou avec peu de) répétition (une méthode n'étant représentée que par une seule réaction).
2. Le second jeu de données est un extrait de 7029 réactions issues des bases de données CHEMINFORM et REFLIB [37]. L'extrait se concentre sur des réactions à haut rendement et ne contenant que des atomes dont l'élément chimique est courant[38]. Contrairement au jeu précédent, ce jeu présente une certaine redondance, plusieurs réactions pouvant illustrer une même méthode de synthèse.
3. Le troisième jeu est une sélection de 3248 réactions illustrant la méthode de Diels-Alder qui est une des plus célèbres méthodes de synthèse, permettant de construire des cycles à 6 chaînons tel qu'illustré par son schéma général donné sur la figure 4.21. Les équations chimiques contenues dans ce jeu de données présentent donc une forte redondance structurelle.

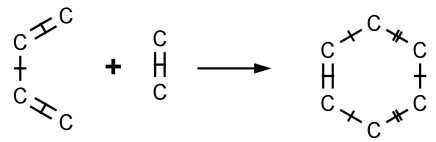


FIG. 4.21: Schéma de la méthode de synthèse de Diels-Alder.

Le tableau de la figure 4.22 présente les résultats obtenus suite à l'extraction par l'algorithme `Gaston` – qui a l'avantage d'être un outil à la fois très efficace et téléchargeable librement – des schémas de réactions fréquents dans les différents jeux de données[39]. Ces jeux

base	taille	fréquence minimale	temps de calcul (s)	nombre de schémas fréquents	nombre de schémas non dégénérés fréquents	taille moyenne
A (ORGSYN)	2608	50 (2 %)	521	319261	10200 (3,2 %)	34
B (REFLIB)	7029	140 (2 %)	306	158122	47359 (30 %)	25
C (Diels-Alder)	3248	150 (4,5 %)	274	685441	658346 (96 %)	25

FIG. 4.22: Résultats des tests

de données permettent d'observer, à travers le tableau 4.22, des résultats bien connus de la recherche des motifs fréquents en général :

– Tout d'abord le temps de calcul n'est évidemment pas fonction uniquement du nombre d'exemples fouillés. Une autre grandeur influente est le nombre de motifs fréquents à extraire, fixé indirectement par le choix du seuil minimal de fréquence. Ainsi la recherche des motifs dont la fréquence relative est supérieure ou égale à 2 % prend presque deux fois plus de temps pour fouiller les 2608 exemples du jeu A que les 7029 exemples du

[37] Ces deux bases de données sont des produits commerciaux de la société Symyx®/MDL®.

[38] Ces éléments sont H, C, O, N, F, Cl, Br, I, P, B, S et Si.

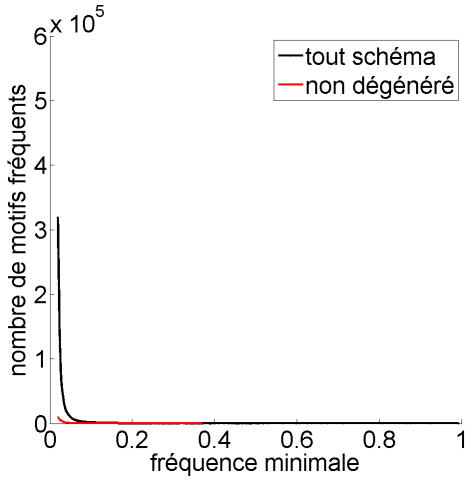
[39] Le nombre de motifs fréquents rapporté ne décompte pas les motifs réduits à un seul sommet, qui ne sont pas fouillés par Gaston.

jeu B. Ce facteur 2 s'explique a posteriori par le fait que le jeu A présente près de deux fois plus de motifs fréquents que le jeu B.

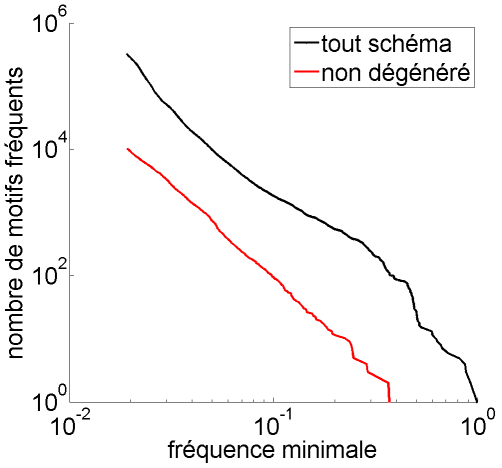
- Cependant le nombre de motifs fréquents et la taille des données ne suffisent pas pour déterminer, même approximativement, le temps de calcul. Ainsi la fouille des réactions du jeu C est plus rapide que celle du jeu A alors que le jeu C compte plus d'exemples et produit deux fois plus de motifs fréquents que le jeu A. Un des autres facteurs influant sur le temps de calcul est sans doute la taille moyenne du motif exprimée dans la dernière colonne du tableau (définie ici comme la somme des nombres de sommets et d'arêtes du motif). En effet certains traitements ralentissent lorsque le motif courant grandit (comme le calcul du représentant canonique). Cette explication hypothétique semble plausible compte tenu de la taille moyenne des motifs fréquents dans le jeu A qui est bien supérieure à celle du jeu C.
- Le nombre de motifs fréquents n'est pas corrélé dans l'absolu avec le seuil minimal de fréquence f_{min}, même si ce nombre est une fonction décroissante du seuil f_{min} pour un jeu de données fixé : un seuil de fréquence relative de 2 % pour le jeu B (soit un support minimal de 140 exemples) produit 4 fois moins de motifs fréquents qu'un seuil minimal supérieur égal à 4,5 % pour le jeu C (soit un support minimal de 150 exemples). Ces différences s'expliquent par la densité variable des données qui peuvent présenter un nombre plus ou moins élevé de motifs fréquents pour une fréquence donnée. La valeur minimale de f_{min} en deçà de laquelle un algorithme de recherche des motifs fréquents n'arrive plus à extraire les motifs fréquents en un temps raisonnable dépend donc fortement de la densité des données fouillées.

Les différences de densité se manifestent clairement sur les courbes de la figure 4.23 dont les axes sont gradués de manière identique pour en faciliter la comparaison. Ces courbes résument l'évolution du nombre de motifs fréquents en fonction du seuil minimal de fréquence pour chacun des trois jeux de données. La décroissance de ces courbes illustre le caractère anti-monotone des motifs fréquents. On observe en particulier une augmentation du nombre de motifs fréquents extrêmement brusque lorsque le seuil de fréquence minimale décroît suffisamment (cf courbes 4.23(a), 4.23(c) et 4.23(e)). Cette augmentation se produit toutefois à des seuils de fréquence différents : ainsi les 10000 motifs les plus fréquents sont produits à des fréquences minimales de 5 %, 6 % et 27 % respectivement pour les jeux de données A, B et C. Le jeu C est donc plus dense que le jeu B qui lui-même est plus dense que le jeu A. Cette différence de densité s'explique par la plus grande concentration de réactions similaires dans le jeu C ayant de nombreux motifs en commun. La caractéristique linéaire décroissante des courbes sur une échelle log-log (cf courbes 4.23(b), 4.23(d) et 4.23(f)) semble indiquer une loi inversement polynomiale lorsque le seuil f_{min} s'approche de 0, approximativement inversement cubique (i.e. $N(f_{min}) \sim \frac{1}{f_{min}^3}$).

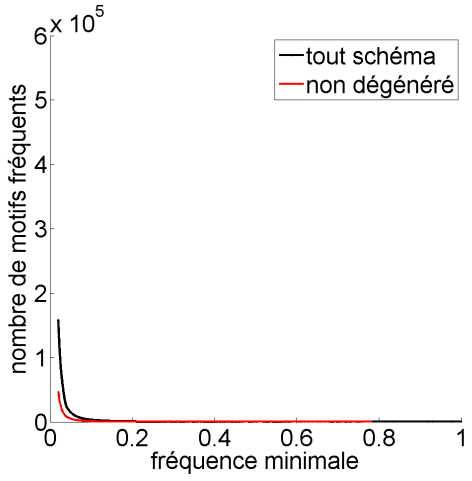
L'échelle log-log permet également de montrer que le nombre de schémas non dégénérés tend à être proportionnel au nombre total de schémas fréquents lorsque f_{min} tend vers 0 puisque les courbes respectives des schémas dégénérés et non dégénérés forment deux droites parallèles. Toutefois le facteur de proportionnalité varie fortement d'un jeu de données à l'autre et tend à augmenter lorsque la densité des données diminue au point de s'inverser complètement : ainsi 96 % des schémas fréquents du jeu A sont dégénérés alors que seulement 3 % des schémas fréquents du jeu C le sont encore. Ce résultat permet d'évaluer empiriquement le gaspillage, déjà mentionné à la section 4.5, du temps de calcul utilisé pour fouiller les schémas de réactions dégénérés. Le rendement de la méthode proposée passe ainsi d'un niveau excellent pour des jeux de réactions fortement redondantes (96 % pour le jeu C) à un niveau



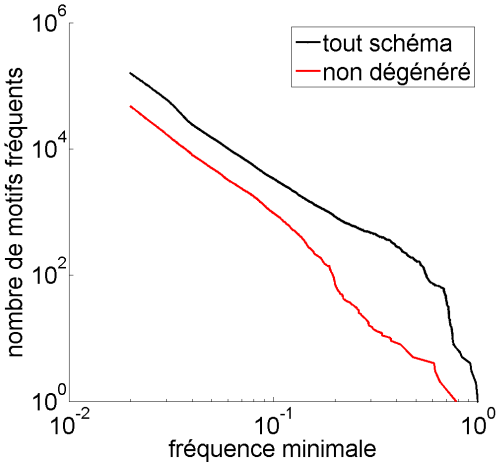
(a) Données A, échelle linéaire



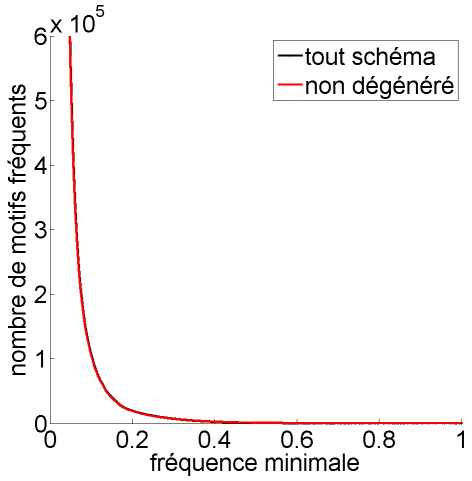
(b) Données A, échelle log-log



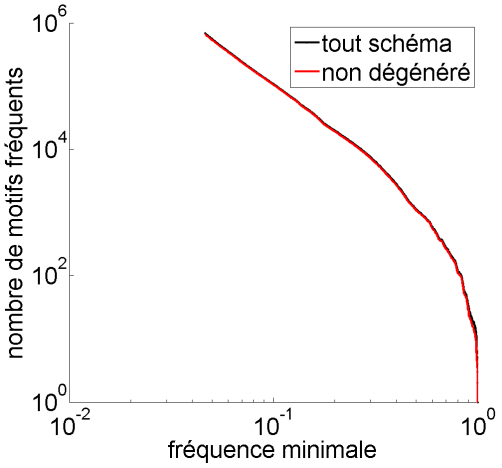
(c) Données B, échelle linéaire



(d) Données B, échelle log-log



(e) Données C, échelle linéaire



(f) Données C, échelle log-log

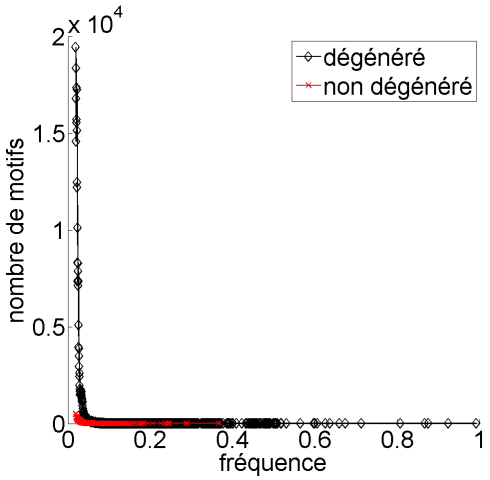
FIG. 4.23: Nombre de motifs fréquents fonction du seuil minimal de fréquence

médiocre pour des données variées (3 % pour le jeu A).

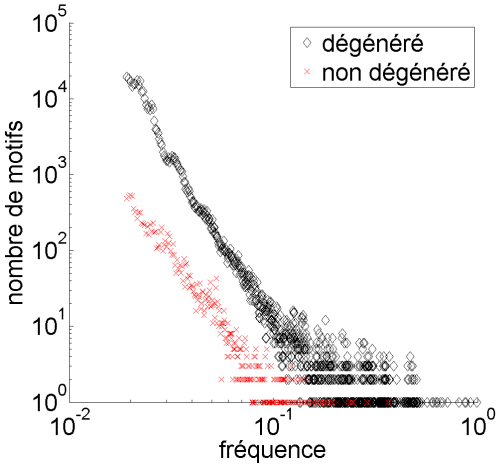
Pour mieux comprendre ce phénomène, il est intéressant d'extraire des courbes précédentes la distribution des schémas réactionnels dégénérés et non dégénérés selon leur fréquence. Ces distributions sont représentées sur la figure 4.24. En particulier les figures 4.24(b), 4.24(d) et 4.24(f) montrent que les trois nuages de points associés aux schémas dégénérés (en noir) sont superposables. Cette observation est intéressante puisqu'elle atteste que la distribution des schémas dégénérés, c'est à dire des fragments de graphes moléculaires est globalement identique entre les 3 jeux de données, et comme on peut le penser, pour la plupart des jeux de données de réactions. Le corollaire immédiat de cette observation est que ce sont les schémas non dégénérés (en rouge) qui sont à l'origine de la différence du nombre de motifs fréquents observée entre les trois jeux de données (cf figure 4.23). Tout comme les schémas dégénérés, les nuages de points associés aux schémas non dégénérés sont d'aspect similaire et se forment autour de droites médianes à peu près parallèles. Leur pente correspond à une loi de type $F(f) \sim \frac{1}{f^4}$, ce qui est cohérent avec les courbes de la figure 4.23 compte tenu du fait de la relation qui lie le nombre $N(f_{min})$ de schémas fréquents à la distribution $F(f)$ des motifs (i.e. $N(f_{min}) = \sum_{f_{min} \leq f \leq 1} F(f)$). Si les pentes sont comparables, l'ordonnée à l'origine des droites médianes varie sensiblement d'un jeu de données à l'autre : pour un jeu de réactions variées (jeu de données A, figure 4.24(b)), les schémas non dégénérés sont minoritaires (i.e. la courbe rouge est sensiblement en dessous de la courbe noire). Au fur et à mesure que le jeu de données se « densifie », la courbe rouge se décale sur la droite et se rapproche de la courbe noire (jeu B, figure 4.24(d)), puis la dépasse au point que les schémas dégénérés deviennent à leur tour minoritaires (jeu C, figure 4.24(f)). Ce résultat s'explique par le fait que les graphes de réactions non dégénérés construits à partir et autour des quelques liaisons modifiées, créées ou brisées présentent une combinatoire élevée dans un jeu varié comme le jeu A. Au contraire les graphes de réactions dégénérés se construisent à partir des liaisons stables beaucoup plus nombreuses et permettent ainsi de faire ressortir les agencements relativement peu variés mais très fréquents des fragments moléculaires présents dans les réactants et les produits des équations chimiques (par exemple le cycle aromatique). Les schémas non dégénérés présentent donc en moyenne une fréquence beaucoup plus faible que les schémas dégénérés. C'est ce que l'on observe sur la figure 4.24(b), où l'essentiel des schémas de fréquence supérieure à 0,2 sont dégénérés mais restent relativement peu nombreux. Si toutefois les données considérées contiennent des réactions similaires ayant en commun de nombreux schémas réactionnels (comme le jeu C, cf figure 4.24(f)), les nombreux schémas de réactions partagés obtiennent une fréquence élevée. Leur combinatoire étant bien supérieure à celle des fragments moléculaires les plus fréquents (car la plupart des grands motifs très fréquents intersectent nécessairement les liaisons modifiées par la réaction qui sont au centre du graphe de réaction et sont donc non dégénérés), la proportion des schémas non dégénérés s'inverse et devient proche de 1. En conclusion, les expériences ont permis de valider la possibilité d'extraire en pratique des millions de schémas réactionnels fréquents à partir de milliers d'équations de réactions. En outre elles donnent une idée de la richesse combinatoire des motifs contenus dans les BdR et suggèrent que les schémas de réactions fréquents sont difficilement exploitables en l'état, sans autre forme de sélection.

4.8 Conclusions

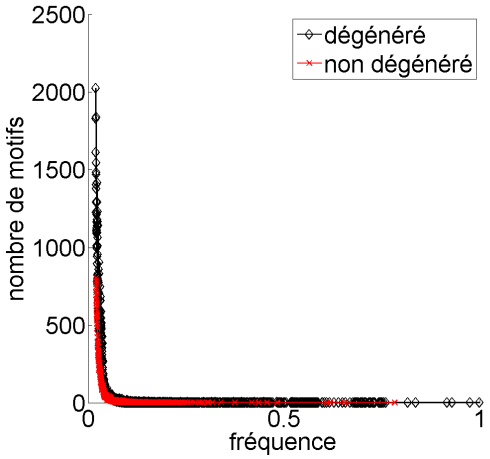
Sans un prétraitement adapté des bases de données de réactions, la fouille des schémas de réactions aurait été impossible. Le développement d'un outil de prétraitement, s'il fut



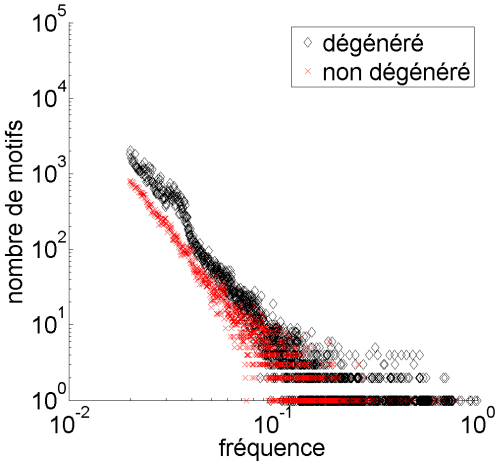
(a) Données A, échelle linéaire



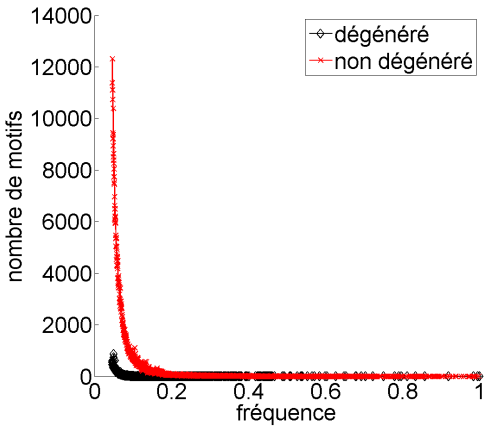
(b) Données A, échelle log-log



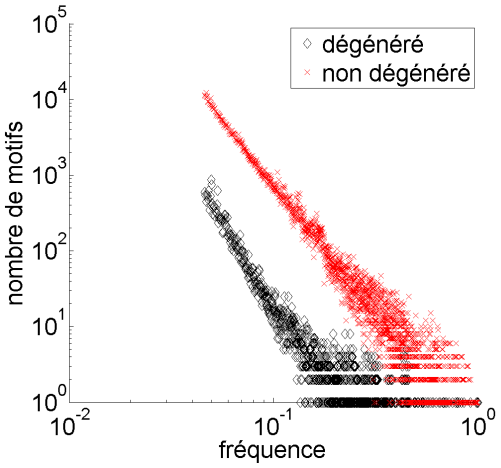
(c) Données B, échelle linéaire



(d) Données B, échelle log-log



(e) Données C, échelle linéaire



(f) Données C, échelle log-log

FIG. 4.24: Distribution des schémas de réactions selon leur fréquence

nécessaire, a exigé le plus grand soin tant les détails techniques posés étaient nombreux. Ces efforts ont permis en retour d'avoir un processus opérationnel d'extraction de connaissance, qui prend en entrée et produit en sortie des informations directement interprétables par les chimistes. Ce prétraitement a non seulement permis d'extraire les schémas de réactions fréquents dans une BdR mais aussi a permis d'évaluer accessoirement la qualité des données dans les BdR et d'en suivre l'évolution. Au delà des difficultés de mise en œuvre, le prétraitement mis au point illustre deux principes intéressants :

- La nécessité de s'intéresser de près à la connaissance du domaine – ici certaines propriétés des molécules et des réactions – pour obtenir des données fiables et pouvoir ainsi réaliser une fouille de données de qualité.
- La possibilité – ici à travers le recours original au modèle des graphes condensés de réactions – de résoudre un problème nouveau à l'aide de méthodes existantes, en recherchant une transposition adéquate du problème dans un modèle qui se prête mieux à sa résolution.

L'extraction des schémas de réactions fréquents a enfin et surtout mis en évidence l'inadéquation des schémas de réactions fréquents (et plus généralement des motifs fréquents) avec les besoins concrets d'une application telle que l'extraction des schémas représentatifs de méthodes de synthèse. Ce constat a motivé le développement du modèle des motifs les plus informatifs présenté au chapitre suivant.

Chapitre 5

Le modèle des motifs les plus informatifs. Application à l'extraction de connaissances à partir des bases de données de réactions

Sommaire

Le chapitre précédent a mis en évidence le fait que les schémas de réactions fréquents sont trop nombreux pour permettre une analyse directe par un expert. Le présent chapitre propose le modèle dit des *motifs les plus informatifs* afin de sélectionner un ensemble réduit de motifs fréquents qui soient représentatifs des données et non redondants comparativement à d'autres familles de motifs comme les motifs fermés fréquents.

5.1 Introduction

La recherche des motifs fréquents a initialement été proposée pour faciliter l'extraction des règles d'association fréquentes (Agrawal *et al.*, 1993b, 1996). Le tri de ces règles d'association selon différentes mesures statistiques (confiance, lift, conviction ... (Piatetsky-Shapiro, 1991; Agrawal *et al.*, 1996; Brin *et al.*, 1997; Omiecinski, 2003)) permet d'aboutir à des règles intéressantes et descriptives des données. Pour comprendre le succès remporté par les règles d'association en fouille de données, il faut se rappeler l'objectif de l'extraction de connaissances qui est de fournir à un expert des éléments d'analyse susceptibles de lui apporter de nouvelles connaissances relatives à sa spécialité. Pour que cette analyse conduite de visu par l'expert soit efficace, ces éléments d'analyse, qu'il s'agisse de règles d'association, de motifs ou de tout autre support imaginable d'information, doivent satisfaire certains critères de qualité :

Expressivité. Les éléments à analyser (règles, motifs ...) doivent s'exprimer dans un langage compréhensible et évocateur pour l'expert. Ainsi une règle d'association $H \rightarrow C$ de confiance 1 est plus expressive que la juxtaposition, pourtant équivalente, de deux motifs H et $H \cup C$ de même fréquence. De même un schéma de réaction est plus parlant pour un chimiste que le graphe de réaction équivalent.

Intérêt. Les éléments pris séparément doivent être *intéressants* ou *informatifs*, c'est-à-dire être les vecteurs d'une véritable information qui aident l'expert à établir, par le biais d'un raisonnement inductif, de nouvelles connaissances. Ainsi les règles de confiance élevée ou au contraire très faible sont, de par leur pouvoir de prédiction, susceptibles d'apporter plus d'information à l'expert que celles de confiance proche de 0,5.

Concision. C'est bien connu, « trop d'information tue l'information ». Les éléments livrés à l'analyse de l'expert ne doivent pas être exhaustifs. Une quantité surabondante d'information dilue l'attention de l'expert qui aura toutes les chances de négliger, suite à sa fatigue ou son désintérêt, l'information réellement importante.

Non-redondance. La production d'un nombre raisonnable d'éléments d'analyse tous très intéressants ne suffit pas toujours à en faire une solution irréprochable. Il se peut en effet que les informations que véhiculent les différents motifs soient certes intéressantes mais se recoupent, voire soient identiques. Les informations sont alors qualifiées de *redondantes*. Ainsi deux schémas réactionnels qui ne diffèrent que par la présence ou l'absence d'un atome d'hydrogène et dont les fréquences et scores sont très proches peuvent être considérés comme redondants. La redondance d'information est contraire au critère de concision et doit être réduite autant que possible.

En conclusion, l'information idéale que doit fournir un système d'extraction de connaissances est un ensemble concis d'éléments expressifs, qui fournisse à l'expert une information intéressante et non redondante. Aux vues de ces critères, les règles d'association présentent cet avantage sur les motifs fréquents d'être plus expressives : alors qu'un motif fréquent flanqué d'une fréquence élevée n'évoquera bien souvent rien à l'expert, une règle d'association de confiance élevée peut mettre en lumière une relation de cause à effet riche d'enseignements.

Malgré cet avantage certain, le problème de l'extraction des règles d'association construites à partir de motifs plus complexes comme les motifs de graphes n'a jamais été traité en pratique, quand bien même des algorithmes de recherche des motifs fréquents sont disponibles pour ces mêmes catégories de motifs (cf section 2.4.2). La notion de règle d'association est pourtant généralisable d'un point de vue théorique à tout ensemble ordonné de motifs : par exemple, une règle d'association entre graphes se définit comme une règle $H \rightarrow C$ où H et C sont des graphes connexes tels que H est isomorphe à un sous-graphe de C. Cette lacune

peut résulter d'une crainte : le nombre déjà très élevé de motifs fréquents de graphes peut faire craindre un nombre encore plus élevé de règles d'association fréquentes, construites à partir des paires de motifs fréquents comparables. Cette crainte est d'autant plus fondée que l'extraction de règles d'association entre graphes nécessite de mettre en œuvre des calculs plus lourds que dans le cas des motifs d'attributs, afin de tenir compte du problème de l'isomorphisme entre motifs.

Quelle qu'en soit la raison, dès lors que la recherche des motifs fréquents ne sert plus l'extraction de règles d'association – comme c'est le cas des graphes – la première préoccupation que devraient susciter de tels motifs fréquents, est d'en trouver l'utilité, avant même de savoir comment les extraire. Certes les motifs de graphes fréquents ont pu trouver des applications dans des problèmes de classification ou d'indexation des données (cf section 2.4.3), à travers l'extraction, parmi les motifs fréquents, de motifs discriminants (i.e. de fréquence élevée dans un ensemble d'exemples positifs et faible dans un ensemble d'exemples négatifs). Mais en dehors de ces applications spécifiques, les motifs de graphes fréquents n'ont pas jusqu'à présent servi à l'extraction de connaissances comme ont pu le faire les motifs d'attributs fréquents, à travers les règles d'association. Faute de pouvoir filtrer les motifs intéressants, l'expert peut toutefois vouloir analyser les motifs les plus fréquents à condition qu'il règle le seuil minimal de fréquence à une valeur élevée, pour limiter son analyse à un nombre acceptable de motifs. Mais cette sélection se fait au détriment de l'intérêt des motifs, puisque la plupart des motifs les plus fréquents, à l'instar du motif vide, sont aussi les moins intéressants. Enfin parmi la faible proportion de motifs qui retiennent *in fine* l'attention de l'expert, de nombreux motifs présentent une *redondance structurelle* d'information, c'est-à-dire à la fois des fréquences et des descriptions très proches (par exemple deux motifs de 10 attributs qui ne diffèrent que d'un attribut ou deux graphes de 10 sommets dont l'un se déduit de l'autre par l'ajout d'une simple arête). Il est alors nécessaire pour l'expert de regrouper l'ensemble des motifs similaires pour ne retenir qu'un seul représentant – ou prototype – par groupe – ou cluster – de motifs.

Le *modèle des motifs les plus informatifs*, introduit dans Pennerath et Napoli (2007) et Pennerath et Napoli (2008b) puis développé dans Pennerath et Napoli (2008a) et Pennerath et Napoli (2009) propose une méthode de sélection des motifs qui satisfait au plus près les critères évoqués précédemment. L'objectif de ce modèle est en effet de produire un ensemble réduit de motifs qui soient simultanément informatifs et peu redondants et qui permette à l'expert de les analyser directement un par un. Intuitivement, un motif est *informatif* s'il est à la fois *descriptif* (i.e. sa structure interne ou description est longue et riche d'information) et *représentatif* des données (i.e. sa fréquence est élevée). Dans le cadre de l'extraction de connaissances à partir de BdR, l'objectif est d'extraire d'une BdR, un nombre limité de schémas de réactions qui soient représentatifs de grandes familles de réactions.

La suite de ce chapitre développe ce modèle des motifs les plus informatifs d'abord dans toute sa généralité puis dans le cadre plus spécifique de l'application aux réactions chimiques. La section 5.2 commence par présenter certaines méthodes utilisées pour réduire le nombre de motifs fréquents à analyser et les répartit selon deux grandes catégories qualifiées de inter et intra motif. Le modèle des motifs les plus informatifs est ensuite introduit comme une approche combinant les principes et les avantages respectifs des deux familles de méthodes. La section 5.3 définit le modèle formel des motifs les plus informatifs et démontre certaines propriétés afférentes à ce modèle. La section 5.4 présente un algorithme pour extraire directement les motifs les plus informatifs puis un algorithme plus efficace de filtrage des motifs fréquents. Cette section compare ensuite les performances de ces algorithmes dans le cas des motifs d'attributs. Enfin la section 5.5 présente l'application du modèle des motifs les plus informatifs à l'extraction de connaissances à partir de bases de données de réactions chimiques

ainsi que les résultats des tests réalisés dans le cadre de cette application.

5.2 Une analyse des méthodes de sélection de motifs fréquents

De nombreux travaux ont eu pour objectif de réduire le nombre de motifs fréquents et donc de règles d'association fréquentes à analyser en proposant diverses méthodes de sélection de motifs. Ces méthodes peuvent se répartir en deux grandes catégories qualifiées de sélection *inter-motifs* et *intra-motif* et présentées respectivement dans les sections 5.2.1 et 5.2.2. Si ces méthodes furent appliquées en premier lieu aux motifs d'attributs, elles sont présentées ici dans leur formalisme le plus général afin de pouvoir être appliquées à toute familles de motifs ordonnés comme les graphes. Cette généralité n'est toutefois que théorique et ne tient pas compte des difficultés spécifiques de mise en œuvre que pose chaque famille de motifs et en particulier celle des graphes.

Ce formalisme général se fonde sur un ensemble arbitraire $\mathcal{M}$ de motifs muni d'une relation d'ordre $\leq_{\mathcal{M}}$ et d'un ensemble $\mathcal{D}$ d'objets tel que chaque objet $o \in \mathcal{D}$ dispose d'une description $d(o) \in \mathcal{M}$. La relation $\leq_{\mathcal{M}}$ est une relation de spécialisation ordonnant les descriptions d'objets des plus générales vers les plus spécifiques : $M_1 <_{\mathcal{M}} M_2$ signifie que le motif M_1 *généralise* ou *subsume* le motif M_2, ou de manière équivalente, que le motif M_2 est une *spécialisation* du motif M_1. Dans le cas des motifs d'attributs, la description d'un objet o est l'ensemble des attributs en relation avec o et la relation $\leq_{\mathcal{M}}$ est l'inclusion ensembliste $\subseteq$. Dans le cas des graphes étiquetés, les motifs sont des graphes non isomorphes deux à deux et la relation $\leq_{\mathcal{M}}$ est la relation de sous-graphe isomorphe. La fréquence d'un motif $M \in \mathcal{M}$ est alors définie comme le nombre d'objets dont M généralise la description : $\text{freq}(M) = |\{o \in \mathcal{D} | M \leq_{\mathcal{M}} d(o)\}|$. En accord avec ces définitions, le terme de *motif* se réfère dans ce qui suit non pas spécifiquement à des motifs d'attributs mais à tout type de motifs ordonnés, qu'il s'agisse de graphes, de séquences ou de schémas de réactions. De même l'expression *ordre des motifs* fait référence à l'ordre $(\mathcal{M}, \leq_{\mathcal{M}})$.

Par ailleurs les différentes méthodes de sélection de motifs présentées dans cette section sont illustrées à partir d'un même exemple fondé, pour des raisons de clarté, sur les motifs d'attributs. Cet exemple jouet est représenté sur la figure 5.1 et constitué de 5 objets décrits par 4 attributs a, b, c et d selon une relation binaire $\mathcal{R}$. L'ordre des motifs associé peut se

$\mathcal{R}$	a	b	c	d
1	×	×	×	
2		×		
3	×	×	×	×
4			×	×
5	×	×	×	×

FIG. 5.1: Exemple de relation binaire objets-attributs

représenter graphiquement par le *diagramme d'ordre* de la figure 5.2. Un diagramme d'ordre est un graphe orienté où les sommets représentent les motifs et où un arc relie le motif M_1 au motif M_2 si M_2 est un successeur immédiat de M_1 (cf définition 5.3.1). La convention veut que les motifs les plus petits et donc les plus généraux soient représentés en haut du diagramme et que les arcs soient représentés par des arêtes implicitement orientées de haut en bas. Sur la figure 5.2 sont également représentées les courbes de niveaux de fréquence départageant

les motifs dont la fréquence est supérieure ou inférieure à un certain niveau, afin de mettre en évidence la décroissance des fréquences dans l'ordre des motifs. Sur cet exemple, l'analyse

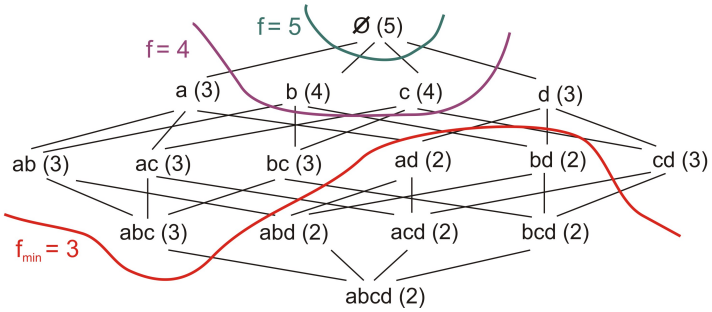


FIG. 5.2: Diagramme de l'ordre des motifs associés à leur fréquence (entre parenthèses) et courbes de niveau de fréquence

brute des motifs de fréquence supérieure ou égale à $f_{min} = 2$ revient à fournir à l'expert la liste $L_f = (5 : \emptyset\ ; 4 : b, c\ ; 3 : a, d, ab, ac, bc, cd, abc\ ; 2 : ad, bd, abd, acd, bcd, abcd)$ des motifs fréquents précédés de leurs fréquences triées par ordre décroissant. Si on omet les exceptions que constituent les motifs *ad* et *bd*, cette liste est celle que l'on obtiendrait en triant les motifs par ordre croissant de longueur. Cette corrélation forte qui existe entre fréquence et longueur des motifs montre l'insuffisance de la fréquence en tant que critère de sélection des motifs.

5.2.1 Sélection inter-motifs

La sélection inter-motifs regroupe l'ensemble des méthodes qui sélectionnent une famille restreinte de motifs fréquents en raison de leur position particulière au sein de l'ordre des motifs. Les *motifs fermés* (Pasquier *et al.*, 1999b,a) sont un exemple d'une telle famille : un motif M est *fermé* s'il n'existe pas de motif supérieur à M dans l'ordre des motifs qui soit de fréquence égale à celle de M. Les motifs fermés peuvent également se définir comme les motifs maximaux dans leur *classe d'équivalence* où deux motifs M_1 et M_2 sont *équivalents* s'ils décrivent le même ensemble d'objets de $\mathcal{D}$: $\forall o \in \mathcal{D}, M_1 \leq_{\mathcal{M}} d(o) \Leftrightarrow M_2 \leq_{\mathcal{M}} d(o)$. Les motifs fermés sont indiqués en gras sur la figure 5.3 au sein de leurs classes d'équivalence.

Parmi les autres familles de motifs appartenant à la même catégorie, on peut citer les motifs *fréquents maximaux* (Bayardo, 1998; Gouda et Zaki, 2005) (i.e. dont tous les successeurs immédiats sont non fréquents), les *motifs générateurs* ou *les motifs libres* (Bastide *et al.*, 2000b; Boulicaut *et al.*, 2003) (i.e. les motifs minimaux des classes d'équivalence, dont la fréquence est strictement supérieure à celles de tous ses prédécesseurs immédiats). La plupart de ces familles de motifs présentent l'avantage de prélever de manière régulière des motifs dans l'ordre des motifs et ainsi de ne pas perdre trop d'information par rapport à la donnée de l'ensemble des motifs fréquents. La donnée de certaines familles de motifs ainsi que de leurs fréquences permettent même d'en déduire, à l'aide d'un algorithme, les fréquences de l'ensemble des motifs fréquents, la perte d'information étant alors nulle. Cette compression réversible de l'ensemble des motifs fréquents par une famille plus réduite de motifs est qualifiée de *représentation condensée* des motifs fréquents. Ainsi les motifs fermés fréquents sont une représentation condensée des motifs fréquents. Il en va de même des motifs générateurs fréquents si on leur adjoint les motifs non fréquents minimaux (i.e. les motifs non fréquents dont tous les prédécesseurs sont fréquents). Lorsque les données à fouiller ne sont pas aléatoires mais que leurs constructions sont contraintes par des règles intrinsèques, les motifs

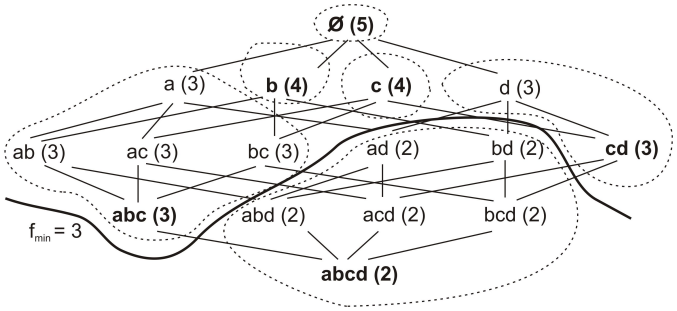


FIG. 5.3: Diagramme de l'ordre des motifs associés à leur fréquence (entre parenthèses), motifs fermés (en gras), classes d'équivalence (courbes pointillées) et la frontière des motifs fréquents (trait en gras)

fermés ou générateurs fréquents tendent à être beaucoup plus rares que les motifs fréquents.

Le principal inconvénient de l'approche inter-motifs reste que les représentations condensées ne sont pas plus informatives en moyenne que les motifs fréquents (i.e. comportent peu d'information par motif utile pour l'expert) : les motifs fermés de grande fréquence auront tendance à être des motifs de faible longueur comportant peu d'information structurelle. Inversement les motifs longs auront tendance à ne pas être représentatifs car de faible fréquence. Ainsi sur l'exemple de la figure 5.1, la liste $L_c = (5 : \emptyset\ ; 4 : b, c\ ; 3 : cd, abc\ ; 2 : abcd)$ des motifs fermés triés par ordre décroissant de fréquence est certes plus courte que la liste L_f mais place toujours en tête des motifs aussi inexpressifs que le motif vide. Les sélections inter-motifs ont également l'inconvénient de ne pas pouvoir intégrer la connaissance du domaine et de produire un nombre de motifs qui reste souvent important en pratique (voir par exemple les courbes de la figure 5.16 à la fin de ce chapitre). Ce dernier point a d'ailleurs motivé des travaux récents (Xin *et al.*, 2005; Hasan *et al.*, 2007; Bringmann et Zimmermann, 2007) dont le but est d'éliminer la redondance d'information entre motifs et par la même de réduire leur nombre.

5.2.2 Sélection intra-motif

À l'opposé de la sélection inter-motifs, la sélection intra-motif consiste à évaluer l'intérêt d'un motif à partir de sa fréquence mais aussi de sa structure intrinsèque. Cette prise en compte de la structure du motif peut être rudimentaire, par exemple en considérant seulement la longueur du motif. Elle peut au contraire être sophistiquée, par exemple en intégrant la sémantique associée aux différents attributs présents dans un motif ou encore certaines caractéristiques topologiques d'un motif de graphe comme la présence de cycles. La sélection intra-motif ne tient donc pas compte des relations d'un motif avec ses motifs voisins dans l'ordre des motifs. La sélection des motifs permet de trier les motifs fréquents par ordre décroissant de leur intérêt a priori afin que seule la tête de liste soit fournie à l'expert pour analyse. Cette approche suppose de pouvoir quantifier l'intérêt a priori d'un motif selon une fonction de score calculable à partir de la structure et de la fréquence du motif. Pour que la sélection soit efficace, l'intérêt a priori calculé par la fonction de score doit réaliser une approximation aussi fidèle que possible de l'intérêt réel exprimé a posteriori par l'expert. L'extraction efficace des top-k motifs (Wang *et al.*, 2005b; Xin *et al.*, 2006; Soulet et Crémilleux, 2007) s'inscrit dans cette approche : les top-k motifs sont les k motifs d'attributs obtenant le meilleur score pour une certaine fonction de score. En particulier Soulet et Crémilleux (2007)

traite le cas particulier des top-k motifs pour la fonction d'aire $\mathbf{s}_a : (M, f) \mapsto f \cdot |M|$ où $|M|$ est la longueur du motif M (i.e. le nombre d'attributs) et f la fréquence de M. Une approche

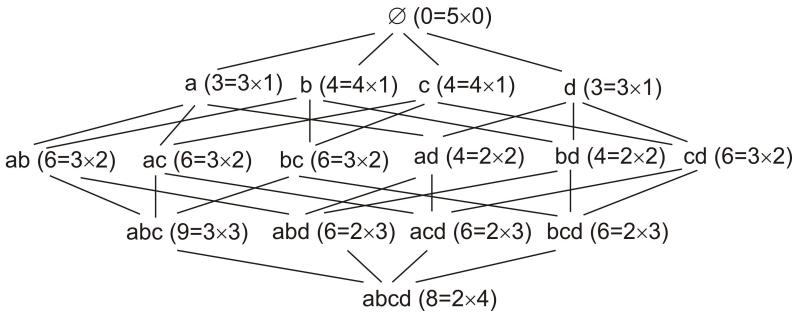


FIG. 5.4: Ordre des motifs et leur aire ($s_a(M) = \text{freq}(M) \times |M|$)

similaire est utilisée par `Subdue` (Cook et Holder, 1994) dans le cas des graphes. Les scores associés à cette fonction d'aire et à l'exemple de la figure 5.1 sont précisés sur la figure 5.4. La liste des motifs fréquents (pour $f_{min} = 2$) précédés de leurs scores triés par ordre décroissant est $L_a = (9 : abc\ ; 8 : abcd\ ; 6 : abd, acd, bcd, ab, ac, bc, cd\ ; 4 : ad, bd, b, c\ ; 3 : a, d\ ; 0 : \emptyset)$.

Contrairement à la sélection inter-motifs, la sélection intra-motif présente l'avantage de permettre facilement l'injection de la connaissance du domaine d'application dans la fonction de score afin que le score reflète l'intérêt du motif relativement à l'application considérée. Mais la sélection intra-motif comporte aussi l'inconvénient d'une forte redondance structurelle entre motifs. Ainsi la tête de la liste L_a présente des motifs similaires *abc*, *ab*, *ac*, *bc* pour lesquels l'expert ne retiendra probablement que le meilleur d'entre eux, soit *abc*. Le motif suivant *cd* dans la liste est différent des motifs précédents par la présence de l'attribut *d*. Les motifs restants *b*, *c*, *a* et *d* seront rattachés tantôt à *abc*, tantôt à *cd*. Au final seuls les deux motifs *abc* et *cd* seront retenus. Cette redondance est une conséquence de la méthode de sélection : les motifs voisins dans l'ordre des motifs (i.e. proches selon la distance du plus court chemin dans le diagramme d'ordre) présentent en effet à la fois une structure et une fréquence similaires donc des scores rapprochés. Ces motifs se retrouvent donc à des rangs également voisins dans la liste des motifs fréquents triés par ordre décroissant de score. La tête de liste se trouve finalement saturée par des motifs fortement redondants situés à proximité l'un de l'autre dans l'ordre des motifs.

5.3 La famille des motifs les plus informatifs

5.3.1 Motivations

La section précédente fait apparaître les deux sélections, inter-motifs et intra-motif comme complémentaires. L'objectif du modèle des motifs les plus informatifs développé ci-après est de concilier les avantages respectifs des deux types d'approches pour produire un nombre restreint de motifs, peu redondants mais informatifs. Intuitivement les motifs les plus informatifs, abrégés par MPI, sont à la fois des motifs descriptifs (i.e. présentant une description longue riche en information) et représentatifs (i.e. d'une fréquence élevée). Dans la mesure où ces deux dimensions sont antinomiques (i.e. la fréquence est une fonction décroissante de la description des motifs), l'extraction des motifs les plus informatifs est nécessairement un problème d'optimisation qui sélectionne les motifs exprimant le meilleur compromis entre leur fréquence et l'information contenue dans leur description (intégrant éventuellement la

connaissance a priori du domaine). Elle retient donc l'idée d'une fonction de score adoptée par la sélection intra-motif pour exprimer ce compromis. La principale différence est de ne sélectionner que les motifs qui rendent la fonction de score localement maximale dans l'ordre des motifs, réduisant ainsi leur nombre de manière drastique et éliminant une grande partie de la redondance entre motifs.

5.3.2 Le modèle des motifs les plus informatifs

L'ordre des motifs $(\mathcal{M}, \leq_{\mathcal{M}})$ est supposé disposer d'un motif minimal $\emptyset_{\mathcal{M}}$, appelé motif vide. La définition des motifs les plus informatifs repose sur les notions de fonction de score et de motifs voisins dans l'ordre des motifs :

Définition 5.3.1. Étant donné un ensemble ordonné $(\mathcal{M}, \leq_{\mathcal{M}})$, l'élément $M' \in \mathcal{M}$ est un *prédécesseur immédiat* de l'élément $M \in \mathcal{M}$ si l'intervalle $]M'; M[$ est vide (i.e. $M' <_{\mathcal{M}} M$ et $M' \leq_{\mathcal{M}} M'' <_{\mathcal{M}} M \Rightarrow M'' = M'$). M' est un *successeur immédiat* de M si M est un prédécesseur immédiat de M'. Deux éléments M et M' sont *voisins* dans l'ordre $(\mathcal{M}, \leq_{\mathcal{M}})$ si M' est un prédécesseur ou successeur immédiat de M.

Définition 5.3.2. Étant donné un ensemble $\mathcal{D}$ de données, une *fonction de score* est une fonction $\mathbf{s} : \mathcal{M} \times [0; 1] \to \mathbb{S}$ où l'ensemble $\mathbb{S}$, appelé *ordre des scores*, est muni d'une relation d'ordre $\leq_{\mathbb{S}}$ qui peut être partielle ou totale. Le *score* $s(M)$ d'un motif M relativement à $\mathcal{D}$ et s est $s(M) = \mathbf{s}(M, \text{freq}_r(M))$, où $\text{freq}_r(M)$ est la fréquence relative de M dans $\mathcal{D}$.

Seules les fonctions de score *informatives* réalisent un compromis acceptable entre description et représentativité des motifs et sont considérées dans ce qui suit.

Définition 5.3.3. Une fonction de score s est *informative* si les propriétés suivantes sont vraies :

1. Pour tout motif non vide M, la fonction partielle $\mathbf{s}^M : f \mapsto \mathbf{s}(M, f)$ est une fonction strictement croissante de f :

$$\forall M \in \mathcal{M} \setminus \{\emptyset_{\mathcal{M}}\}, \forall (f_1, f_2) \in [0; 1]^2, f_1 < f_2 \Rightarrow \mathbf{s}^M(f_1) <_{\mathbb{S}} \mathbf{s}^M(f_2)$$

2. Pour tout nombre réel $f \in]0; 1]$, la fonction partielle $\mathbf{s}^f : M \mapsto \mathbf{s}(M, f)$ est une fonction strictement croissante de $M \in \mathcal{M}$:

$$\forall f \in]0; 1], \forall (M_1, M_2) \in \mathcal{M}^2, M_1 <_{\mathcal{M}} M_2 \Rightarrow \mathbf{s}^f(M_1) <_{\mathbb{S}} \mathbf{s}^f(M_2)$$

3. Conditions aux limites : un motif de fréquence nulle ou vide ne peut avoir un score supérieur à celui d'un motif de fréquence non nulle ou non vide.

Toute fonction $\mathbf{s} : (M, f) \mapsto s'(M) \cdot f$ où l'ordre des scores est l'ordre $(\mathbb{R}^+, \leq)$ des nombres réels positifs et où $s' : \mathcal{M} \to \mathbb{R}^+$ est une fonction strictement croissante définie dans l'ordre des motifs est un exemple de fonction de score informative puisque les fonctions associées $\mathbf{s}^M$ pour $M \neq \emptyset_{\mathcal{M}}$ et $\mathbf{s}^f$ pour $f > 0$ sont des fonctions strictement croissantes respectivement de f et de M. La fonction d'aire s_a introduite précédemment est informative puisqu'elle correspond à la fonction s où $s'(M) = |M|$ est la longueur du motif d'attributs M.

Définition 5.3.4. Soient un ordre des motifs $(\mathcal{M}, \leq_{\mathcal{M}})$ et une fonction de score informative s associée à un ordre des scores $(\mathbb{S}, \leq_{\mathbb{S}})$ et à un ensemble de données $\mathcal{D}$.

– Le motif $M \in \mathcal{M}$ *domine* le motif $M' \in \mathcal{M}$ si et seulement si M et M' sont voisins dans $(\mathcal{M}, \leq_{\mathcal{M}})$ et si $s(M) >_{\$} s(M')$.
– Un motif M est *des plus informatifs* si et seulement si $\text{freq}(M) \neq 0$ et si aucun motif ne domine M.

La figure 5.5 reprend le diagramme de la figure 5.4 dans lequel les arêtes ont été orientées selon la relation de dominance pour la fonction de score s_a : un arc $M_1 \rightarrow M_2$ indique que M_1 domine M_2. Ainsi le motif *abcd* est dominé par le motif *abc* et domine les motifs *abd*, *acd* et *bcd*. Certaines arêtes ne sont pas orientées lorsque les deux motifs reliés sont de scores non comparables ou égaux, comme c'est le cas de l'arête qui relie les motifs voisins *cd* et *bcd*. Les motifs les plus informatifs figurant en gras sont ceux qui ne sont pointés par aucun arc : ils sont comme escompté *abc* de score 9 puis *cd* de score 6.

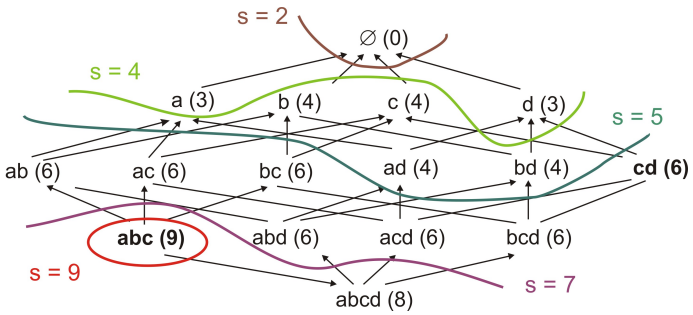


FIG. 5.5: Scores (entre parenthèses), motifs les plus informatifs (en gras), relations de dominance (arcs orientés) et courbes de niveau du score (en couleur) pour la fonction de score s_a

Le problème que posent les motifs les plus informatifs est leur extraction :

Définition 5.3.5. Le problème de l'*extraction des motifs les plus informatifs fréquents* consiste à extraire des données $\mathcal{D}$ l'ensemble des motifs les plus informatifs de fréquence supérieure ou égale à un seuil f_{min} ainsi que le score et la fréquence qui leur sont associés.

Il est important de noter que la détermination des motifs les plus informatifs fréquents nécessite de connaître non seulement les fréquences de tous les motifs fréquents mais aussi celles des motifs non fréquents dont au moins un prédécesseur immédiat est fréquent. En effet en supposant qu'un motif M fréquent ne soit dominé par aucun de ses prédécesseurs immédiats, les successeurs immédiats fréquents de M peuvent très bien ne pas dominer M alors qu'il existe un successeur immédiat M' de M qui le domine mais qui n'est pas fréquent. L'absence de prise en compte du motif M' aboutirait à la conclusion que M est des plus informatifs alors qu'il ne l'est pas. Ainsi sur l'exemple précédent, si on extrait les motifs de fréquence supérieure ou égale à $f_{min} = 4$, les motifs b et c ne sont dominés par aucun autre motif fréquent, alors qu'ils sont dominés tous deux par *bc*. Cet ensemble des motifs non fréquents dont **au moins un** prédécesseur immédiat est fréquent peut faire penser à la frontière négative telle qu'on l'entend habituellement, et qui correspond aux motifs non fréquents dont **tous** les prédécesseurs immédiats sont fréquents. La définition suivante vise à éviter toute confusion tout en soulignant l'analogie partielle avec la frontière négative.

Définition 5.3.6. La *frontière négative extensive*, en abrégé FNE, est l'ensemble des motifs non fréquents ayant au moins un prédécesseur immédiat fréquent. Cet ensemble inclut la frontière négative.

5.3.3 Propriétés des motifs les plus informatifs

La première propriété démontrée concernant le modèle des motifs les plus informatifs suppose que l'ordre des motifs $(\mathcal{M}, \leq_{\mathcal{M}})$ vérifie la propriété suivante :

Propriété 5.3.7. *Pour tout ensemble de données fini et non vide $\mathcal{D}$, le nombre de motifs de fréquence non nulle relativement à $\mathcal{D}$ est fini et non nul.*

Cette hypothèse se vérifie en pratique pour l'essentiel des bases de données fouillées, mais peut être fausse si on considère des objets contenant une infinité de motifs (par exemple des séquences ou des graphes de taille infinie, mais qui peuvent être codés à l'aide de structures de taille finie, comme par exemple des automates finis dans le cas des séquences). Il découle de cette hypothèse la propriété suivante :

Propriété 5.3.8. *Les motifs les plus informatifs forment un ensemble non vide.*

Démonstration. Il existe au moins un motif M_1 de fréquence non nulle. Par l'absurde si l'ensemble des motifs des plus informatifs est vide, M_1 doit être dominé par un autre motif M_2 qui est nécessairement de fréquence non nulle d'après le troisième axiome de la définition 5.3.3. Par itération on obtient ainsi une séquence $(M_i)_{i\geq 1}$ de motifs de fréquence non nulle telle que $s(M_i) <_{\$} s(M_{i+1})$ pour tout $i \geq 1$. Les motifs de cette séquence forment un sous-ensemble de l'ensemble des motifs de fréquence non nulle, qui est un ensemble fini d'après la propriété 5.3.7. La séquence infinie (M_i) prend donc ses éléments dans un ensemble fini de motifs et il existe nécessairement deux indices j et $k > j$ tels que $M_j = M_k$. La relation $<_{\$}$ étant transitive, on obtient la contradiction $s(M_j) <_{\$} s(M_j)$. □

La recherche des motifs les plus informatifs fréquents peut cependant ne produire aucun résultat si le seuil f_{min} est trop élevé. Les motifs les plus informatifs présentent des propriétés intéressantes vis-à-vis des motifs fermés :

Propriété 5.3.9. *Tout motif des plus informatifs relativement à une fonction de score informative est un motif fermé.*

Démonstration. Soit un motif M' des plus informatifs relativement à une fonction de score informative s d'ordre de score $(\$, \leq_{\$})$. Si M' n'est pas fermé, il existe un successeur immédiat M'' de M' tel que $\text{freq}(M'') = \text{freq}(M')$. Puisque s est informative et que $f = \text{freq}(M') \neq 0$, le deuxième axiome de la définition 5.3.3 s'applique : la fonction $\mathbf{s}^f : M \mapsto \mathbf{s}(M, f)$ est strictement croissante. Par définition de M'', $M' <_{\mathcal{M}} M''$ ce qui entraîne $\mathbf{s}^f(M') <_{\$} \mathbf{s}^f(M'')$, soit encore $s(M') <_{\$} \mathbf{s}^f(M'')$. Comme $\text{freq}(M'') = \text{freq}(M')$, $\mathbf{s}^f(M'') = s(M'')$ et donc $s(M') <_{\$} s(M'')$. La dominance du motif M'' sur M' contredirait finalement l'hypothèse voulant que M' soit des plus informatifs. Le motif M' est donc fermé. □

Dans l'exemple de la figure 5.5, les motifs les plus informatifs *abc* et *cd* apparaissent bien comme des motifs fermés sur la figure 5.3. Si les motifs les plus informatifs sont des motifs fermés, à l'inverse, l'extraction des motifs fermés fréquents peut être vue comme un cas particulier de l'extraction des motifs les plus informatifs fréquents. Cette équivalence repose sur la notion d'*ordre produit* noté $(E_1, \leq_1) \times (E_2, \leq_{12})$ de deux ordres $(E_1, \leq_1)$ et $(E_2, \leq_2)$ égal à l'ordre $(E_1 \times E_2, \leq_{12})$ où $E_1 \times E_2$ est le produit cartésien de E_1 et E_2 et où la relation $\leq_{12}$ est définie par : $(x_1, x_2) \leq_{12} (y_1, y_2)$ si et seulement si $x_1 \leq_1 y_1$ et $x_2 \leq_2 y_2$.

Propriété 5.3.10. *Les* motifs fermés *sont les motifs les plus informatifs relativement à la fonction de score informative s égale à la fonction identité $Id : (M, f) \mapsto (M, f)$ et à l'ordre des scores $(\mathcal{M}, \leq_{\mathcal{M}}) \times ([0; 1], \leq)$.*

Démonstration. La fonction identité Id est de façon évidente une fonction de score informative dans l'ordre des scores $(\mathcal{M}, \leq_{\mathcal{M}}) \times ([0;1], \leq)$. La propriété 5.3.9 établit que tout motif des plus informatifs relativement à Id est fermé. Inversement soit M un motif et M' un successeur immédiat de M dominant M, i.e. $(M, \text{freq}(M)) <_{\$} (M', \text{freq}(M'))$. L'ordre produit implique $\text{freq}(M) \leq \text{freq}(M')$, ce qui prouve par définition que M ne peut être fermé. Un motif fermé ne peut donc être dominé par un successeur immédiat. Puisque un prédécesseur immédiat de M est par définition plus petit que M selon $\leq_{\mathcal{M}}$ et ne peut davantage dominer M selon Id, un motif fermé est des plus informatifs relativement à Id. □

Les deux propriétés 5.3.9 et 5.3.10 montrent ensemble que les motifs fermés correspondent à la famille de motifs les plus informatifs la moins restrictive parmi tous les choix possibles de fonctions de scores informatives.

5.3.4 Exemples de fonctions de score

Afin d'illustrer la notion de fonction de score informative, cette section introduit différents exemples de telles fonctions, résumées par la figure 5.6.

Fonction de score	Ordre des scores
$Id : (M, f) \mapsto (M, f)$	$(\mathcal{M}, \leq_{\mathcal{M}}) \times ([0;1], \leq)$
$\mathbf{s}_c : (M, f) \mapsto (\lvert M \rvert, f)$	$(\mathbb{R}^+, \leq) \times ([0;1], \leq)$
$\mathbf{s}_a : (M, f) \mapsto \lvert M \rvert \cdot f$	$(\mathbb{R}^+, \leq)$
$\mathbf{s}_i : (M, f) \mapsto I(M) \cdot f$	$(\mathbb{R}^+, \leq)$

FIG. 5.6: Exemples de fonctions de score informatives

Fonctions Id et s_c

Soit l'opérateur $|\cdot| : \mathcal{M} \mapsto \mathbb{R}^+$ mesurant la taille d'un motif (i.e. la longueur d'un motif d'attributs ou la somme du nombre de sommets et d'arêtes dans le cas d'un graphe). Cette fonction est une fonction strictement croissante (i.e. $M_1 <_{\mathcal{M}} M_2 \Rightarrow |M_1| < |M_2|$). Par conséquent la fonction $s_c : (M, f) \mapsto (|M|, f)$ est une fonction informative dont les motifs les plus informatifs sont ceux de la fonction identité Id, c'est-à-dire les motifs fermés : en effet étant donnés deux motifs M_1 et M_2 de $\mathcal{M}$, la proposition « $M_1 \leq_{\mathcal{M}} M_2$ » est logiquement équivalente à la proposition « M_1 et M_2 sont comparables selon $\leq_{\mathcal{M}}$ et $|M_1| \leq |M_2|$ ». Puisque les motifs voisins d'un motif M sont par définition comparables à M, les motifs les plus informatifs relativement aux fonctions de score Id et s_c sont identiques. La fonction s_c correspond à la fonction véritablement utilisée en pratique pour extraire les motifs fermés en place de Id : la comparaison de deux scores issus de s_c ne nécessite en effet qu'une simple comparaison entre deux nombres (i.e. les tailles des deux motifs), là où la comparaison de deux scores issus de Id nécessite un test beaucoup plus coûteux d'inclusion entre ensembles dans le cas des motifs d'attributs ou pire de détection de sous-graphe isomorphe dans le cas de graphes.

Fonction s_a

Selon les propriétés 5.3.9 et 5.3.10, les motifs fermés forment la plus grande famille de motifs les plus informatifs. La fonction s_a est un exemple de fonction de score plus sélective

qui correspond à la fonction d'aire dans le cas particulier des motifs d'attributs. Cette fonction de score se fonde sur le principe de *longueur minimale de description* ou MDL introduit par Rissanen (1978). Ce paradigme considère l'apprentissage comme un problème de compression de la réalité en modèles qui la décrivent. Les quantités de données et la taille du modèle sont supposées être mesurables selon une même quantité $I(\cdot)$ d'information égale au nombre de bits nécessaires pour les stocker en mémoire. Étant donné un modèle M décrivant dans une certaine mesure les données $\mathcal{D}$, il est possible de comprimer partiellement $\mathcal{D}$ de manière réversible, en tenant compte de l'information apportée par le modèle M. La substitution des données initiales $\mathcal{D}$ par les données compressées $c(\mathcal{D}, M)$ associées au modèle M (ce dernier étant nécessaire pour décompresser $c(\mathcal{D}, M)$ en M) permet une économie de quantité d'information égale à $\Delta I = I(\mathcal{D}) - I(M) - I(c(\mathcal{D}, M))$. Le meilleur modèle décrivant $\mathcal{D}$ est donc celui qui maximise ΔI ou plus simplement qui minimise le terme $I(M) + I(c(\mathcal{D}, M))$. `Subdue` (Cook et Holder, 1994) applique déjà ce principe MDL à la fouille de graphes : `Subdue` considère la compression d'un ensemble $\mathcal{D}$ de graphes consistant, étant donné un motif de graphe connexe g, à contracter toutes les occurrences de g (i.e. tous les sous-graphes isomorphes à g) dans $\mathcal{D}$ en des sommets particuliers. Le motif g le plus efficace est celui qui maximise l'espace sauvé égal en première approximation au produit $|g| \cdot \text{occ}(g)$ de la taille $|g|$ du graphe par le nombre $\text{occ}(g)$ d'occurrences de g dans $\mathcal{D}$. Ce produit s'apparente à la fonction s_a dans le cas particulier où les motifs sont des graphes.

Fonction s_i

La fonction d'information s_i reprend le principe de s_a et l'améliore. Le facteur $|M|$ égal à la taille du motif est remplacé par la quantité d'information $I(M)$ du motif M définie ci-après. Dans le cas des motifs d'attributs, cette quantité est égale à la somme des quantités d'information rattachées à chacune des étiquettes composant le motif M. La définition suivante correspond au cas plus général des graphes, le cas des motifs d'attributs pouvant être traité comme un cas particulier de motifs de graphes sans arêtes :

Définition 5.3.11. La quantité $I(g)$ d'information d'un graphe g relativement à $\mathcal{D}$ est définie ici comme la somme des quantités d'information portées par chacun des sommets $v \in V(g)$ d'étiquette $l_v(v)$ et chacune de ses arêtes $e \in E(g)$ d'étiquette $l_e(e)$:

$$I(g) = \sum_{v \in V(g)} i(l_v(v)) + \sum_{e \in E(g)} i(l_e(e))$$

La quantité d'information associée à une étiquette de sommet ou d'arête est :

$$i(l) = -\log_2 \left(\frac{n(l)}{\sum\limits_{l' \in \mathcal{L}} n(l')} \right)$$

où $n(l)$ désigne le nombre de sommets ou d'arêtes de $\mathcal{D}$ ayant pour étiquette l choisie parmi l'ensemble $\mathcal{L}$ des étiquettes de sommets ou d'arêtes.

Cette fonction de score permet de privilégier les motifs constitués d'éléments (i.e. d'étiquettes ou d'attributs) naturellement rares dans $\mathcal{D}$. Ainsi dans l'exemple jouet de la figure 5.1, l'information associée à a et d est de $-log_2(3/14) = 2,2$ bits, celle associée à b et c, qui sont plus fréquents, est seulement de $-log_2(4/14) = 1,8$ bits. La figure 5.7 représente le diagramme d'ordre de l'exemple associé aux nouveaux scores obtenus selon s_i. Des deux motifs les plus informatifs *abc* et *cd* associés à la fonction s_a (cf diagramme de la figure 5.5), ne subsiste que le motif le plus informatif *abc*.

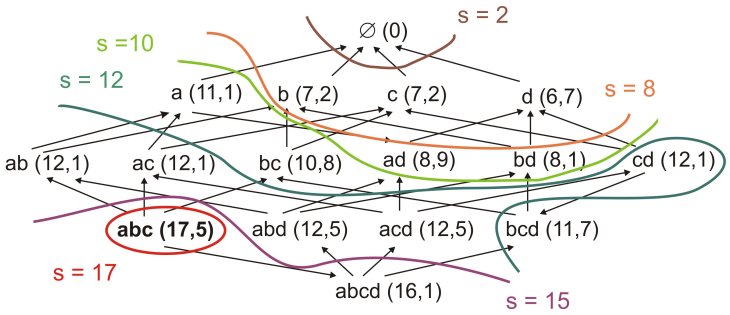


FIG. 5.7: Scores (entre parenthèses), motifs les plus informatifs (en gras), relations de dominance (arcs orientés) et courbes de niveau du score (en couleur) pour la fonction de score s_i

Vers des fonctions de scores spécifiques

La disparition du motif *cd* parmi les motifs les plus informatifs dans l'exemple de la figure 5.7 illustre l'importance du choix de la fonction de score puisque cette dernière conditionne le résultat qui sera ensuite analysé par l'expert. Ce choix, qui peut paraître subjectif, doit être pensé en fonction de la question que se pose l'expert. La fonction de score utilisée doit idéalement être construite « sur mesure » à partir des caractéristiques du motif que l'on veut faire ressortir ou au contraire que l'on veut masquer, puis éventuellement être ajustée au cours des cycles consécutifs du processus d'extraction de connaissance. Si cette volonté de guider les résultats par des connaissances a priori peut être critiquée à tord ou à raison, le choix de la fonction de score n'en demeure pas moins un degré de liberté intéressant.

Une des façons de procéder est d'ajouter dans la fonction de score des facteurs additionnels tenant compte de propriétés structurelles du motif qui sont recherchées. Le seul point à respecter, si on veut conserver les propriétés des motifs les plus informatifs, est que la fonction de score qui en résulte soit informative. Considérons l'exemple des fonctions de score du type $s : (M, f) \mapsto s'(M) \cdot f$ où l'ordre des scores est $(\mathbb{R}^+, \leq)$. La fonction $s' : M \mapsto I(M) + \sum c_i \cdot t_i(M)$ peut inclure de nouveaux termes $t_i(M)$ pondérés, en plus de la quantité d'information $I(M)$ introduite à la section 5.3.4. Pour que la fonction s reste informative, il suffit que les différents termes $t_i : \mathcal{M} \to \mathbb{R}^+$ introduits soient des fonctions croissantes dans l'ordre des motifs. Nombreuses sont les fonctions répondant à tel critère. Dans le cas des graphes, on peut citer en exemple les caractéristiques suivantes : nombre de sommets, nombre d'arêtes, nombre de cycles, nombre de sommets ou arêtes d'un certain type, nombre de sous-graphes partiels isomorphes à un graphe donné, le degré maximal, la longueur maximal du cycle ou du chemin inclus dans le motif ...

L'intérêt de ces termes additionnels est de pouvoir privilégier certains aspects du motif en fonction des spécificités de l'application traitée. Dans le cas de la synthèse organique, la formation de cycles ou de stéréo-centres (d'atomes au centre d'une configuration géométrique particulière) constituent des atouts renforçant l'intérêt d'un schéma de réaction. Une fonction de score adaptée pourrait donc inclure deux termes additionnels que sont le nombre de cycles dans le motif qui contiennent au moins une liaison formée et le nombre de stéréo-centres créés, nombres qui croissent bien avec le motif. De même les groupes fonctionnels sont des groupes d'atomes qui sont connus des chimistes pour influer fortement sur la réactivité des molécules. Leur décompte et leur intégration dans la fonction de score est une possibilité. À l'inverse, le squelette carboné et les atomes d'hydrogène sont en synthèse organique moins significatifs

dans les schémas caractéristiques des méthodes de synthèse, car relativement stables. On peut donc vouloir ramener leur quantité d'information à un niveau plus bas que celui calculé à partir des données pour faire ressortir les groupes fonctionnels principalement constitués d'hétéroatomes.

5.4 L'extraction des motifs les plus informatifs fréquents

Trois algorithmes ont été successivement développés pour extraire les motifs les plus informatifs. Alors que les deux premiers sont des algorithmes d'*extraction directe*, qui extraient les motifs les plus informatifs fréquents en fouillant directement les données, le troisième est un *algorithme de filtrage* qui sélectionne les motifs les plus informatifs parmi les motifs fréquents. Le premier des deux algorithmes d'extraction directe, qui est décrit dans l'article Pennerath et Napoli (2008a) a été depuis modifié pour donner le second plus efficace, décrit dans la section 5.4.1. L'algorithme de filtrage est une alternative à l'extraction directe et est présenté dans la section 5.4.2. La section 5.4.3 réalise ensuite une étude comparative des deux approches. Il est également utile de préciser que les algorithmes présentés extraient non seulement les motifs les plus informatifs fréquents mais aussi les motifs fermés fréquents dans la mesure où le calcul additionnel nécessaire à cette seconde extraction est négligeable.

5.4.1 Algorithme d'extraction directe

L'algorithme n° 2 présenté dans cette section et implémenté en C++ dans le logiciel de fouille de graphes baptisé `Forage` réalise l'extraction directe des motifs les plus informatifs fréquents à partir des données $\mathcal{D}$. L'algorithme consiste à parcourir les arcs du diagramme de l'ordre des motifs $(\mathcal{M}, \leq_{\mathcal{M}})$ selon un parcours en profondeur partant d'un ensemble $\mathcal{M}_{min}$ de motifs minimaux généralement réduit au motif vide $\emptyset_{\mathcal{M}}$. Les motifs sont générés en appliquant à un motif courant M une extension aboutissant à un successeur immédiat de M. Dans le cas des motifs d'attributs, une extension consiste à ajouter un nouvel attribut non encore présent dans M. Dans le cas des graphes connexes simples, une extension d'un motif non vide consiste à ajouter soit un nouveau sommet connecté par une arête à un sommet de M, soit une arête entre deux sommets de M non adjacents. Un motif n'est développé que si celui-ci est fréquent. Par ailleurs, chaque extension du motif courant correspond à un et seul arc du diagramme de l'ordre des motifs. L'algorithme parcourt donc la totalité du diagramme de l'ordre des motifs fréquents (i.e. la restriction du diagramme aux motifs fréquents) plus tous les arcs reliant un motif fréquent à un motif non fréquent (i.e. de la FNE). À chaque arc $M_1 \rightarrow M_2$ est associée la comparaison du score du motif courant M_1 avec celui d'un de ses successeurs immédiats M_2. Si les deux scores sont comparables et différents, le motif du score inférieur est éliminé de l'ensemble des motifs candidats à être des plus informatifs.

La figure 5.8 précise les détails de l'algorithme n° 2. Le parcours en profondeur de l'ordre des motifs est réalisé à l'aide d'une fonction `developpe` appelée récursivement à la ligne 11. Cette fonction prend pour arguments le motif courant M ainsi que l'entrée e qui lui est associée dans un dictionnaire $\mathcal{T}$ de motifs. Ce dictionnaire implémenté selon une structure d'arbre de préfixes (ou trie), permet de déterminer l'ensemble des motifs qui ont déjà été fouillés. Chaque entrée e associée à un tel motif M contient quatre champs pour stocker sa fréquence $\text{freq}_r(M)$, le score $s(M, \text{freq}_r(M))$ de M et deux drapeaux booléens *mpi* et *closed* qui sont vrais si M figure toujours parmi les candidats valables pour faire partie respectivement de l'ensemble $\mathcal{I}$ des motifs les plus informatifs fréquents et de l'ensemble $\mathcal{C}$ des motifs fermés fréquents. L'algorithme garantit qu'à l'issue de la récursion, le champ *mip* (resp. *closed*) n'a pu rester

Données : Les données $\mathcal{D}$, l'ensemble $\mathcal{M}_{min}$ des motifs minimaux, la fréquence minimale f_{min}, la fonction de score s et son ordre des scores $(\mathbb{S}, \leq_{\mathbb{S}})$
Résultat : La liste $\mathcal{I}$ des motifs fréquents les plus informatifs et la liste $\mathcal{C}$ des motifs fréquents fermés

début
  Créer un dictionnaire de motifs $\mathcal{T}$ et les listes vides $\mathcal{I}$ et $\mathcal{C}$;
  pour chaque *motif* $M \in \mathcal{M}_{min}$ **faire**
    Calculer $f \leftarrow \text{freq}_r(M, \mathcal{D})$;
    Créer l'entrée e telle que $e.score \leftarrow s(M, f)$, $e.freq \leftarrow f$, $e.mpi \leftarrow$ vrai et $e.closed \leftarrow$ vrai;
    Associer M à e dans $\mathcal{T}$;
    Appeler `developpe` (M, e)
1 **pour chaque** *motif* $M \in \mathcal{T}$ *associé à l'entrée* e **faire**
    si $e.closed$ *est vrai* **alors**
      Ajouter $(M, e.score, e.freq)$ à $\mathcal{C}$;
      si $e.mpi$ *est vrai* **alors**
        Ajouter $(M, e.score, e.freq)$ à $\mathcal{I}$
2 Trier les motifs de $\mathcal{I}$ et $\mathcal{C}$ par ordre décroissant de score
fin
3 **fonction** `developpe`(*motif* M, *entrée* e) **début**
4 Calculer les paires $(M', \text{freq}_r(M'))$ de $\text{succs}(M, \mathcal{D})$;
5 **pour chaque** $(M', f') \in \text{succs}(M, \mathcal{D})$ **faire**
6 **si** $f' \geq f_{min}$ **alors**
7 Chercher l'entrée e' associée à M' dans $\mathcal{T}$;
8 **si** e' *n'existe pas* **alors**
9 Créer l'entrée e' telle que $e'.score \leftarrow s(M', f')$, $e'.freq \leftarrow f'$, $e'.mpi \leftarrow$ vrai et $e'.closed \leftarrow$ vrai;
10 Associer M' à e' dans $\mathcal{T}$;
11 Appeler `developpe` (M', e')
12 **si** $e.score <_{\mathbb{S}} e'.score$ **alors**
13 $e.mpi \leftarrow$ faux ;
  si $e.freq = e'.freq$ **alors**
14 $e.closed \leftarrow$ faux
15
  sinon si $e.score >_{\mathbb{S}} e'.score$ **alors**
16 $e'.mpi \leftarrow$ faux
17
  sinon si $e.mpi$ *est vrai* **alors**
18 **si** $e.score <_{\mathbb{S}} s(M', f')$ **alors**
19 $e.mpi \leftarrow$ faux
fin

FIG. 5.8: Algorithme n° 2 d'extraction directe des motifs les plus informatifs

vrai que si M est un motif fréquent des plus informatifs (resp. fermé). Cette garantie explique la construction de $\mathcal{I}$ et $\mathcal{C}$ dans la boucle finale à la ligne 1. Les motifs de $\mathcal{I}$ et $\mathcal{C}$ sont enfin triés par ordre décroissant de score et les listes résultant du tri sont éventuellement tronquées (cf ligne 2).

La fonction `developpe`, qui est au cœur de l'algorithme, commence à la ligne 4 par calculer en une seule passe dans les données $\mathcal{D}$ toutes les extensions possibles du motif courant M qui conduisent à des successeurs immédiats de M de fréquence non nulle. L'ensemble de ces successeurs ainsi que leur fréquence sont stockées dans la liste $\text{succs}(M, \mathcal{D})$. Chacun de ces successeurs (cf ligne 5) qui s'avère fréquent (cf ligne 6), est ensuite recherché dans le dictionnaire $\mathcal{T}$. Si ce motif M' ne s'y trouve pas, cela signifie que le motif est généré pour la première fois. Il est alors ajouté au dictionnaire puis développé à son tour (cf ligne 11). Que M' ait déjà été généré ou non, les scores de M et M' sont ensuite comparés dans l'ordre des scores et leurs drapeaux *mpi* respectifs sont mis à jour en fonction du résultat de la comparaison (lignes de 12 à 16 et de 17 à 19).

Si l'algorithme est dans son principe assez simple, sa mise en œuvre se complique lorsque les motifs sont définis à un isomorphisme près, comme c'est le cas des graphes. Ces complications portent principalement sur deux points :

- D'une part, puisque deux graphes isomorphes correspondent à un et un seul motif, les motifs ne peuvent pas être directement utilisés comme les clefs d'accès au dictionnaire $\mathcal{T}$ sans quoi deux motifs isomorphes auraient des entrées distinctes. Pour résoudre ce problème, un représentant canonique est choisi dans chaque classe d'équivalence de motifs isomorphes pour représenter tout élément de cette classe. Chaque fois que le dictionnaire est consulté pour un motif, le motif canonique de ce dernier est d'abord calculé avant d'être utilisé comme clef pour accéder au dictionnaire. À ces fins, `Forage` intègre un algorithme dont le principe est semblable à celui de `Nauty` (McKay, 1981), qui est connu pour être un des algorithmes les plus rapides dans le calcul de la représentation canonique de graphes étiquetés.
- D'autre part le calcul des fréquences de motifs et de l'ensemble $\text{succs}(M, \mathcal{D})$ des successeurs de fréquence non nulle nécessite de recourir à des listes d'appariement, stockant la liste des occurrences du motif dans les données (voir les détails à la section 2.4.2).

Par rapport à l'algorithme n° 1 décrit dans Pennerath et Napoli (2008a), cet algorithme n° 2 apporte essentiellement deux améliorations :

- Afin d'éviter de calculer plusieurs fois la fréquence d'un même motif de graphe à travers deux représentants isomorphes, l'algorithme n° 1 utilisait un mécanisme de cache des fréquences. Cette optimisation consistait à stocker dans le dictionnaire $\mathcal{T}$ les fréquences de tous les motifs générés, c'est-à-dire non seulement de tous les motifs fréquents mais aussi de la FNE (cf 5.3.2). Ce n'est que dans le cas où le motif généré n'était pas présent dans le dictionnaire que sa fréquence était calculée en une passe dans les données puis ajoutée au dictionnaire. En pratique cette optimisation n'était pas judicieuse dans la mesure où les motifs de la FNE étaient très nombreux, surchargeaient le dictionnaire et ralentissaient sensiblement les temps d'accès (voir la courbe de la FNE de la figure 5.13 pour s'en convaincre). Dans l'algorithme n° 2, le mécanisme de cache n'est plus utilisé. Seuls les motifs fréquents sont stockés dans le dictionnaire, ce qui allège sensiblement les besoins de mémoire vive et réduit les temps d'accès aux entrées du dictionnaire.
- L'algorithme n° 1 générait toutes les extensions du motifs, y compris les extensions qui conduisaient à des motifs de fréquence nulle. Ces derniers n'ont aucune chance de dominer un motif fréquent selon le troisième axiome de la définition d'une fonction informative (cf définition 5.3.3) et leur génération devrait être évitée autant que pos-

sible. Pour éviter ce gaspillage en temps de calcul, le calcul de l'ensemble des extensions possibles du motif courant dans les données (cf ligne 5) évite de générer ces motifs de fréquence nulle. Dans les faits, les motifs de fréquence nulle sont très nombreux (du moins en ce qui concerne les graphes de molécules ou de réactions) de sorte que cette optimisation diminue sensiblement le nombre de calculs de fréquence.

Enfin,avant de décrire l'algorithme de filtrage, il est utile de préciser cette propriété :

Proposition 5.4.1. *L'algorithme n^o 2 est valide et complet.*

Démonstration. L'algorithme est complet car si un motif M fréquent des plus informatifs venait à manquer dans le résultat, c'est que soit il n'a pas été traité par la fonction `developpe`, soit son drapeau *mip*, qui est initialement vrai, a été passé à faux. Comme l'algorithme garantit d'appliquer la fonction `developpe` à tout motif fréquent du fait du caractère anti-monotone de la fréquence, c'est donc que le drapeau *mip* a été passé à faux. Cela ne peut se produire qu'aux lignes 13, 16 ou 19. Or dans les 3 cas, le motif M est explicitement dominé tantôt par un prédécesseur (ligne 16), tantôt par un successeur (lignes 13 et 19) immédiat. Ceci entraîne une contradiction avec le fait que le motif M soit des plus informatifs.

L'algorithme est valide parce que un motif ne peut avoir, à l'issue de la récursion à la ligne 1, un drapeau *mip* vrai que s'il est dans le dictionnaire $\mathcal{T}$, c'est-à-dire s'il est fréquent. Or tout motif fréquent voit son score comparé à ceux de tous ses successeurs immédiats, à la ligne 12 si son successeur immédiat est fréquent ou à la ligne 19 sinon. Par ailleurs tout motif fréquent M a des prédécesseurs immédiats qui sont nécessairement fréquents. Tout prédécesseur immédiat de M étant fréquent, voit donc son score comparé à ceux de tous ses successeurs immédiats, et dont M fait partie. Finalement tout motif fréquent voit son score comparé à ceux de tous ses prédécesseurs et successeurs immédiats et ne peut avoir, à l'issue du traitement, un drapeau *mip* vrai que s'il est un motif des plus informatifs fréquent. □

Le principe de l'extraction directe des motifs les plus informatifs peut paraître assez coûteuse, en particulier parce qu'elle nécessite de générer tous les motifs de la FNE. Un algorithme de filtrage des motifs fréquents permet d'éviter la génération de tous les motifs de la FNE, comme expliqué dans la section suivante, et peut à ce titre, être une alternative plus efficace.

5.4.2 Algorithme de filtrage des motifs fréquents

Une autre solution pour extraire les motifs les plus informatifs consiste à filtrer les motifs fréquents extraits préalablement à l'aide d'un outil existant de recherche des motifs fréquents. L'idée est donc de procéder à une fouille de données en deux temps, comme c'est déjà le cas pour l'extraction des règles d'association fréquentes. Tout comme les algorithmes de recherche des motifs fréquents par niveau du type `Apriori`, le principe général de l'algorithme n^o 3 consiste à traiter les motifs fréquents niveau par niveau, un niveau regroupant l'ensemble des motifs de même taille dans l'ordre des motifs (la taille d'un motif d'attributs étant sa longueur et celle d'un motif de graphes étant son nombre d'arêtes). Plus exactement les scores des motifs fréquents du niveau n sont comparés avec ceux de leurs prédécesseurs immédiats qui font partie des motifs fréquents du niveau $n-1$. Ce faisant, la comparaison des scores permet d'éliminer les candidats à être des plus informatifs parmi i) ceux du niveau n qui auraient des prédécesseurs immédiats de score plus élevé et ii) ceux du niveau $n-1$ qui auraient des successeurs immédiats de score plus élevé. Le processus est réitéré pour toutes les longueurs n possibles des motifs fréquents. À l'issue de ce filtrage dit *primaire*, les motifs

candidats qui n'ont pas été éliminés ne peuvent pas encore être proclamés des plus informatifs et doivent subir un *filtrage secondaire* plus sélectif. En effet, il se peut que certains successeurs immédiats d'un tel motif M candidat, ne soient pas fréquents – et qui donc n'ont pas pu être considérés – mais dominent malgré tout M, auquel cas M doit être éliminé. Il est donc nécessaire de calculer les scores et donc les fréquences de tous les successeurs immédiats de M afin de vérifier que M n'est dominé par aucun d'entre eux.

La figure 5.9 donne les détails de l'algorithme n° 3, qui prend en entrée l'ensemble des motifs fréquents $\mathcal{F}$ relativement à un seuil minimal f_{min} de fréquence et fournit en sortie les listes $\mathcal{I}$ et $\mathcal{C}$ des motifs les plus informatifs fréquents et fermés fréquents. L'algorithme n° 3 se décompose en 3 grandes étapes :

Initialisation des lignes 1 à 4. L'initialisation consiste d'abord à subdiviser la liste $\mathcal{F}$ des motifs fréquents en $k+1$ listes $\mathcal{F}_l$ pour $0 \leq l \leq k$ (k étant la longueur maximale atteinte par un motif fréquent), qui chacune regroupe les motifs de même longueur l (boucle de la ligne 2). Dans un deuxième temps, les motifs fréquents de longueur nulle sont stockées dans une liste temporaire $\mathcal{L}$ (boucle de la ligne 3). Chaque motif stocké dans cette liste se voit associé une entrée e identique à celle utilisée dans l'algorithme n° 2.

Filtrage primaire Une fois l'initialisation réalisée, la boucle principale (ligne 5) considère successivement chaque niveau l allant de 1 à k. Le filtrage primaire s'étend de la ligne 9 à la ligne 16. La $l^{\text{ème}}$ itération compare les scores de l'ensemble des motifs fréquents de longueur l, appelé dans ce contexte *niveau supérieur*, aux scores de l'ensemble des motifs fréquents de longueur $l-1$, appelé *niveau inférieur*. Le niveau supérieur devient le niveau inférieur à l'itération suivante. De cette façon, un motif de niveau l est d'abord comparé à ses prédécesseurs immédiats lors de la $l^{\text{ème}}$ itération puis comparé aux successeurs immédiats fréquents lors de la $l+1^{\text{ème}}$ itération. C'est pourquoi chaque itération prend en entrée la liste $\mathcal{L}$ calculée lors de l'itération précédente des motifs de longueur $l-1$ associés à leur score et leur drapeau *mip* et produit une liste $\mathcal{L}'$ similaire pour les motifs de longueur l (ligne 16). Cette liste $\mathcal{L}'$ devient la liste $\mathcal{L}$ au passage à l'itération suivante (ligne 23). Afin de permettre la comparaison des scores entre motifs, le niveau inférieur stocké dans la liste $\mathcal{L}$ est d'abord chargé dans un dictionnaire $\mathcal{T}$ (boucle de la ligne 6). Ensuite le score de chaque motif M du niveau supérieur présent dans $\mathcal{F}_l$ (boucle de la ligne 8) est calculé à partir de la fréquence f de M et stocké dans une entrée e associée à M. Pour chaque prédécesseur immédiat M' de M (boucle de la ligne 9), son entrée e' est récupérée dans $\mathcal{T}$, les scores de M et M' sont comparés et les drapeaux *mpi* mis à jour. Le résultat du filtrage primaire se trouve ainsi dans $\mathcal{T}$ pour le niveau inférieur et dans $\mathcal{L}'$ pour le niveau supérieur.

Filtrage secondaire À l'issue du filtrage primaire de la $l^{\text{ème}}$ itération, les motifs les plus informatifs fréquents de longueur $l-1$ se trouvent nécessairement parmi les motifs de $\mathcal{T}$ dont le drapeau *mip* est encore vrai. Comme expliqué précédemment, un tel motif M doit toutefois passer un second test avant d'être ajouté, à la ligne 22, parmi la liste $\mathcal{I}$ des motifs les plus informatifs fréquents. Ce test consiste à calculer en une passe dans les données les fréquences de l'ensemble des successeurs immédiats de M de fréquence non nulle (ligne 19). Les scores de ses successeurs sont ensuite calculés un par un, jusqu'à ce que éventuellement un de ces scores soit supérieur à celui de M (ligne 20) et le cas échéant que M soit éliminé des candidats (ligne 21).

Tout comme l'algorithme n° 2 de la section 5.4.1, l'algorithme n° 3, simple dans son principe, se complique dans le cas de motifs tels que les graphes :

Données : Les données $\mathcal{D}$, la fonction de score s, son ordre des scores $(\$, \leq_\$)$, et la liste $\mathcal{F}$ des motifs fréquents relativement à un seuil de fréquence minimale f_{min}

Résultat : La liste $\mathcal{I}$ des motifs fréquents les plus informatifs et la liste $\mathcal{C}$ des motifs fréquents fermés munis de leur scores et fréquences

$k \leftarrow 0$;
pour tous les $(M, \text{freq}(M)) \in \mathcal{F}$ **faire**
  Soit l la longueur du motif M ; $k \leftarrow \max(l, k)$;
  $\mathcal{F}_l \leftarrow \mathcal{F}_l \cup \{(M, \text{freq}(M))\}$
Créer les listes vides L, $\mathcal{I}$ et $\mathcal{C}$ et un dictionnaire de motifs $\mathcal{T}$;
pour tous les $(M, f) \in \mathcal{F}_0$ **faire**
  Créer l'entrée e telle que $e.score \leftarrow s(M, f)$, $e.freq \leftarrow f$, $e.mpi \leftarrow$ vrai et $e.closed \leftarrow$ vrai;
  Ajouter (M, e) dans L
pour *l de 1 à $k+1$* **faire**
  Vider le dictionnaire $\mathcal{T}$ et la liste L';
  pour chaque $(M, e) \in L$ **faire**
    Associer à M l'entrée e dans $\mathcal{T}$
  si $l \leq k$ **alors**
    pour chaque $(M, f) \in \mathcal{F}_l$ **faire**
      Créer l'entrée e telle que $e.score \leftarrow s(M, f)$, $e.freq \leftarrow f$, $e.mpi \leftarrow$ vrai et $e.closed \leftarrow$ vrai;
      pour chaque $M' \in \text{preds}(M)$ **faire**
        Chercher l'entrée e' associée à M' dans $\mathcal{T}$;
        si $e.score <_\$ e'.score$ **alors**
          $e.mpi \leftarrow$ faux ;
        sinon si $e.score >_\$ e'.score$ **alors**
          $e'.mpi \leftarrow$ faux **si** $e.freq = e'.freq$ **alors**
            $e'.closed \leftarrow$ faux
      Ajouter (M, e) dans L'
  pour chaque *motif M dans $\mathcal{T}$ associé à l'entrée e* **faire**
    si *$e.closed$ est vrai* **alors**
      Ajouter $(M, e.score, e.freq)$ dans $\mathcal{C}$;
      si *$e.mip$ est vrai* **alors**
        Calculer les motifs $\text{succs}(M, \mathcal{D})$;
        pour chaque $(M', f') \in \text{succs}(M, \mathcal{D})$ *et tant que $e.mip$ est vrai* **faire**
          si $s(M', f') >_\$ e.score$ **alors**
            $e.mip \leftarrow$ faux
        si *$e.mpi$ est vrai* **alors**
          Ajouter $(M, e.score, e.freq)$ à $\mathcal{I}$
  Échanger L et L' ;

FIG. 5.9: Algorithme n° 3 de filtrage des motifs fréquents

– Le problème de l'isomorphisme se pose de la même manière lors de l'accès au dictionnaire $\mathcal{T}$. Il est donc nécessaire de calculer le représentant canonique d'un motif (aux lignes 7 et 10) avant d'accéder au dictionnaire. Toutefois le dictionnaire ne contient à chaque itération qu'un seul niveau de motif contrairement à l'algorithme n° 2 qui nécessitait de stocker tous les motifs fréquents. Il en résulte un besoin réduit en mémoire vive ainsi que des temps d'accès plus rapides.
– Par ailleurs le calcul des prédécesseurs immédiats du motif M à la ligne 9 n'est pas aussi trivial que dans le cas des motifs d'attributs puisqu'il consiste non seulement à supprimer une arête parmi toutes celles possibles de $E(M)$ mais aussi à vérifier que le graphe résultant reste connexe. En termes de théorie des graphes, de telles arêtes dont la suppression conduit à déconnecter le graphe sont des *ponts*. Un algorithme de détection de ces ponts permet de ne supprimer que les arêtes qui n'en sont pas.

Enfin terminons par cette propriété :

Proposition 5.4.2. *L'algorithme 5.9 est valide et complet.*

Démonstration. L'algorithme est complet. En effet l'algorithme passe en revue tous les motifs fréquents et les associent à une entrée e dont le drapeau *mip* est initialisé à vrai. Par conséquent si un motif M des plus informatifs fréquent venait à manquer dans le résultat $\mathcal{I}$, c'est que son drapeau *mip* a été passé à faux. Cela ne peut se produire qu'aux lignes 12, 14 ou 21. Or dans les 3 cas, le motif M est explicitement dominé tantôt par un prédécesseur (ligne 12), tantôt par un successeur (lignes 15 et 21) immédiat. Le motif ne peut donc être des plus informatifs.

L'algorithme est valide parce que un motif est ajouté à l'ensemble résultat $\mathcal{I}$ à la ligne 22 que si son drapeau *mip* est vrai, c'est-à-dire, n'a pas été passé à faux au cours du traitement qui précède. Or tout motif fréquent voit son score comparé à ceux de tous ses prédécesseurs immédiats aux lignes 11 et 13. Tout motif fréquent dont le drapeau *mip* est encore vrai à la ligne 18, voit également son score comparé à la ligne 21 à tous ceux de ses successeurs immédiats qu'ils soient fréquents ou non. Par conséquent un motif ne peut se retrouver dans $\mathcal{I}$ que s'il n'est dominé par aucun de ses prédécesseurs ou successeurs immédiats, c'est-à-dire s'il est des plus informatifs. □

Si les algorithmes n° 2 et n° 3 sont strictement équivalents en terme de résultats, puisqu'ils sont tous deux valides et complets, ces algorithmes se différencient par leurs performances, que ce soit du point de vue du temps de calcul ou de l'occupation mémoire. La section suivante évalue les performances respectives des deux algorithmes.

5.4.3 Analyse comparative des performances

La suite de cette section présente les tests qui ont été réalisés afin de comparer les performances respectives des deux algorithmes déjà présentés ainsi que du plus ancien présenté dans Pennerath et Napoli (2008a) (algorithme n° 1). Afin de faciliter cette comparaison, la famille des motifs fouillés qui a été choisie pour réaliser les tests est celle des motifs d'attributs. Cette famille de motifs présente en effet l'avantage d'être associée à des primitives de calcul (comme le test d'inclusion, le calcul du représentant canonique, le calcul des prédécesseurs ou successeurs immédiats ...) dont la mise au point est plus simple que dans le cas de graphes, ce qui minimise le risque de biaiser la mesure de performance par des différences d'implémentations plus ou moins efficaces. En outre certaines bornes théoriques sont faciles à établir dans le cas des motifs d'attributs alors qu'elles sont très difficiles, voire impossibles à établir dans le cas

des graphes (comme le nombre de prédécesseurs d'un motif ou la complexité du calcul du motif canonique ...)

Par ailleurs les algorithmes développés ont été instrumentés afin de fournir des indicateurs utiles comme le nombre de calcul de fréquences ou le nombre maximal de nœuds internes de l'arbre préfixé servant de dictionnaire de motifs. Ainsi chacun des trois algorithmes permet de déterminer le nombre de motifs fréquents, fermés fréquents et des plus informatifs fréquents qu'il extrait. Comme ces nombres coincident parfaitement d'un algorithme à l'autre, il y a de bonnes raisons pour penser que les trois implémentations ne comportent pas d'erreurs. L'algorithme n° 1 permet par ailleurs de déterminer le nombre de motifs appartenant à la frontière négative extensive (FNE). Enfin l'instrumentation de l'algorithme n° 3 permet de connaître le nombre de motifs candidats à être des plus informatifs à l'issue du filtrage primaire et avant le filtrage secondaire.

Les tests ont tous été réalisés sur des jeux de données issus de l'entrepôt FIMI[40] consacré à la recherche des motifs fréquents. Les fonctions de score testées sont les fonction d'aire et d'information (cf fonctions s_a et s_i du tableau 5.6). La figure 5.10 résume les résultats pour l'ensemble des jeux de données testés. La rapidité de l'algorithme n° x est évalué par

données	MUSHROOM	CHESS	CONNECT	PUMSB	ACCIDENTS
Nombre de transactions	8124	3196	67557	49046	340184
Nombre d'attributs	119	75	129	7116	468
Seuil $f^1_{min}(300s)$ pour s_a	0,27	0,82	0,988	0,99	
Seuil $f^2_{min}(300s)$ pour s_a	0,085	0,63	0,945	0,885	0,63
Seuil $f^3_{min}(300s)$ pour s_a	0,0535	0,49	0,81	0,84	0,19
$N^1(f^1_{min}(300s))$ (s^{-1})	15	18	0,3	0,02	
$N^2(f^2_{min}(300s))$ (s^{-1})	2317	563	16	17	5
$N^3(f^3_{min}(300s))$ (s^{-1})	5666	5350	1458	108	3744
N. de motifs fréquents rel. à $f^3_{min}(300s)$	1626262	1492592	415392	30514	1093194
N. de motifs fermés fréquents	11796	420274	13557	11454	1088580
N. de MPI fréquents pour s_a	19	1	0	0	1
N. de MPI candidats pour s_a	19	191	599	1357	2

FIG. 5.10: Performances des algorithmes n° 1, 2 et 3 pour les différents jeux de données

le nombre $N^x(f)$ de motifs fréquents traités par l'algorithme n° x en une seconde lorsque $f = f_{min}$. Ce seuil f est choisi comme étant la valeur minimale $f^x_{min}(t)$ que le seuil de fréquence f_{min} peut atteindre sans que l'exécution de l'algorithme n° x prenne plus de t secondes[41]. L'algorithme de filtrage (algorithme n° 3) est toutefois avantagé par rapport aux autres algorithmes, puisqu'il dispose dès le départ des fréquences de l'ensemble des motifs

[40] Cf http ://fimi.cs.helsinki.fi/

[41] La mesure de performance est habituellement mesurée par le temps d'exécution que met un algorithme pour réaliser sa tâche pour des entrées données, c'est-à-dire ici, pour un seuil minimal de fréquence f_{min} et des données fixées. Toutefois les différences de rapidité entre les algorithmes sont telles qu'il est difficile de trouver une valeur f_{min} commune aux trois algorithmes, qui concilie à la fois un calcul réalisable pour l'algorithme le plus lent et à la fois une mesure de temps en « régime asymptotique » pour l'algorithme le plus rapide (i.e. hors effets « d'overhead » tel que celui mis en évidence pour l'algorithme 3 sur la figure 5.12). Afin d'éviter ce problème, on utilise en place de cette mesure, cette mesure $f^x_{min}(t)$ qui évite le phénomène d'overhead pour t suffisamment grand, tout en garantissant un temps de calcul qui n'excède pas t secondes.

fréquents. Afin de réaliser une comparaison équitable, le temps de calcul de l'algorithme n° 3 doit inclure non seulement le temps d'exécution propre mais aussi le temps nécessaire pour rechercher l'ensemble des motifs fréquents à l'aide d'un algorithme conçu à cet effet. Dans le cas présent, l'implémentation de l'algorithme `FP-growth` écrite en C++ par Bart Goethals[42] a été utilisée.

Le tableau de la figure 5.10 fait apparaître de grandes variations de densité des motifs fréquents et fermés fréquents. Ainsi un calcul de 5 minutes avec l'algorithme n° 3 permet d'extraire les motifs dont la fréquence relative est supérieure à 5 % dans le cas des données MUSHROOM et à 84 % dans le cas de PUMSB. Le fait que le nombre de MPIs extraits est presque nul dans le cas des données autres que MUSHROOM s'explique par la nature de ces jeux de données qui décrivent des objets ne formant pas de familles caractéristiques. Ces objets ne partageant pas de motifs communs significatifs ont alors peu de chance de faire émerger des MPIs à des niveaux élevés de score et de fréquence. Ainsi le jeu CHESS décrit des successions de coups gagnants ou perdants au jeu d'échec, qui ont peu de chance de présenter des « sous-séquences » communes. Au contraire le jeu MUSHROOM décrit des espèces de champignons, qui forment des familles caractéristiques en accord avec la taxinomie réalisée par les botanistes. À titre d'exemple, l'annexe B énumère les MPIs fréquents extraits du jeu MUSHROOM triés par ordre décroissant de score et précise leur score et leur fréquence. La fréquence et la longueur apparaissent fluctuer au fil de la liste, en accord avec le principe selon lequel la fonction de score exprime un compromis entre fréquence et information structurelle contenue dans le motif. Les MPIs permettent d'identifier de grandes familles de champignons. Par ailleurs, certains MPIs apparaissent contenir d'autres MPIs, formant ainsi des « sous-familles » de champignons.

Toutefois, quel que soit le jeu de données considéré, l'algorithme n° 1 apparaît toujours beaucoup plus lent que les deux autres. L'algorithme n° 3 de filtrage est au contraire le plus rapide même si le temps passé à rechercher les motifs fréquents est inclus dans son temps de calcul. Enfin les performances intermédiaires de l'algorithme n° 2 sont généralement plus proches de celles de l'algorithme n° 3 le plus rapide que de celles de l'algorithme n° 1 le plus lent. La suite de cette section développe à titre d'exemple les résultats associés à la fonction d'aire et au jeu de données MUSHROOM. La figure 5.11 présente les temps de calcul de chacun des trois algorithmes en fonction du seuil minimal de fréquence f_{min}. La figure montre que l'algorithme n° 2 est certes moins rapide que l'algorithme n° 3 mais lorsque f_{min} diminue, l'écart entre les deux algorithmes tend asymptotiquement à se stabiliser vers un facteur situé entre 3 et 4. Par ailleurs, alors que pour un seuil de fréquent élevé, l'essentiel du temps de calcul total de l'algorithme n° 3 est consacré à la recherche des motifs fréquents (cf courbe `FP-growth`), le rapport s'inverse lorsque le seuil f_{min} devient inférieur à $0,3$ (cf courbe algorithme n° 3).

La figure 5.12 précise la répartition du temps de calcul entre les différentes étapes de l'algorithme n° 3 que sont la recherche par `FP-growth` des motifs fréquents, puis la partition des motifs fréquents en niveau, le filtrage primaire et enfin le filtrage secondaire. En régime asymptotique, l'algorithme n° 3 passe l'essentiel de son temps au filtrage primaire des motifs. Même la partition des motifs fréquents en niveau prend plus de temps que la recherche des motifs fréquents en raison des nombreux accès au système de fichier que nécessite cette étape. Le filtrage secondaire consomme un temps négligeable au regard des autres étapes de calcul, même pour les faibles valeurs de f_{min}.

Les courbes des figures 5.11 et 5.12 s'expliquent en partie si on observe sur la figure

[42]Cette implémentation peut être téléchargée à l'adresse http ://www.adrem.ua.ac.be/~goethals/software/

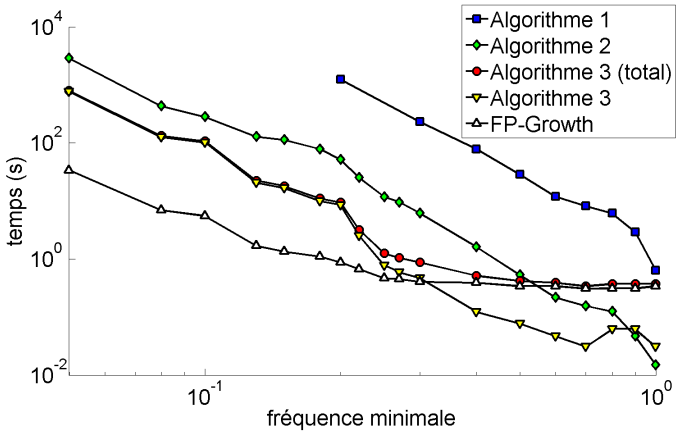


FIG. 5.11: Temps de calcul des trois algorithmes (échelle log-log) pour le jeu de données MUSHROOM

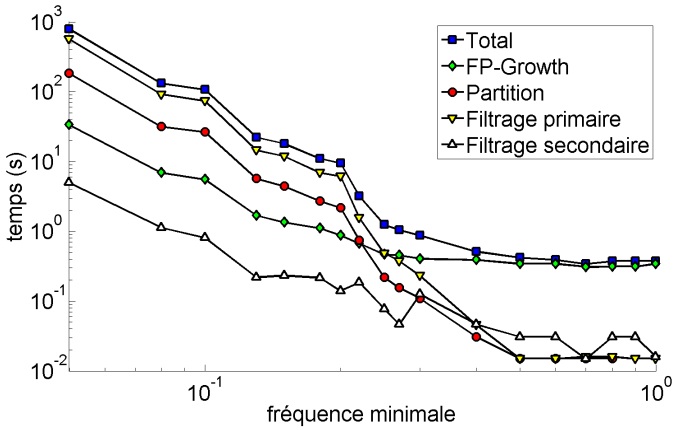


FIG. 5.12: Détail du temps de calcul de l'algorithme 3 (échelle log-log) pour le jeu de données MUSHROOM

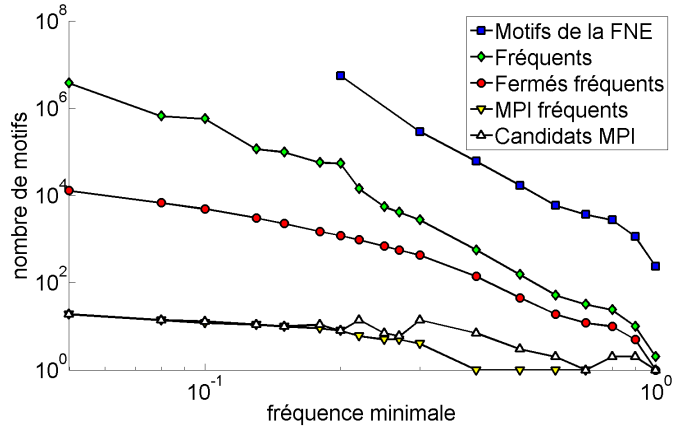


FIG. 5.13: Nombre de motifs par catégorie (échelle log-log) pour le jeu de données MUSHROOM

5.13 le nombre de motifs par catégorie en fonction du seuil minimal de fréquence f_{min}. Les catégories représentées sont les motifs fréquents, les fermés fréquents, les plus informatifs fréquents, les motifs de la FNE générés par l'algorithme n° 1 et les candidats à être des MPIs générés par l'algorithme n° 3. La première constatation est que le nombre de motifs des plus informatifs fréquents est très inférieur au nombre de motifs fréquents et même fermés fréquents. Ainsi pour un seuil de fréquence de 5 %, il existe 19 MPIs fréquents, 12854 motifs fermés fréquents et 3755706 motifs fréquents. Le rapport entre le nombre de MPIs fréquents et de fermés fréquents dépend essentiellement du jeu de données : le jeu MUSHROOMS est connu pour présenter de fortes similarités entre transactions de sorte que la proportion de motifs fermés parmi les motifs fréquents est très faible, de l'ordre de 0,3 %. À l'inverse un jeu de données comme ACCIDENTS a une très forte dispersion. La proportion de fermés parmi les motifs fréquents est alors de 99,7 %, signifiant que la très grande majorité des classes d'équivalence sont réduites à leur motif fermé. Mais même dans le cas de Mushroom, les MPI ne représentent qu'une part infime (0,1 %) des motifs fermés. La figure 5.13 montre également que le nombre de motifs de la FNE est supérieur de 100 fois au nombre de motifs fréquents. Ce phénomène explique le temps de calcul prohibitif de l'algorithme n° 1 qui fouille et stocke dans son dictionnaire les motifs de la FNE. Ce rapport 100 se retrouve ainsi dans le rapport des temps de calcul des algorithmes n° 1 et n° 2 sur la figure 5.11.

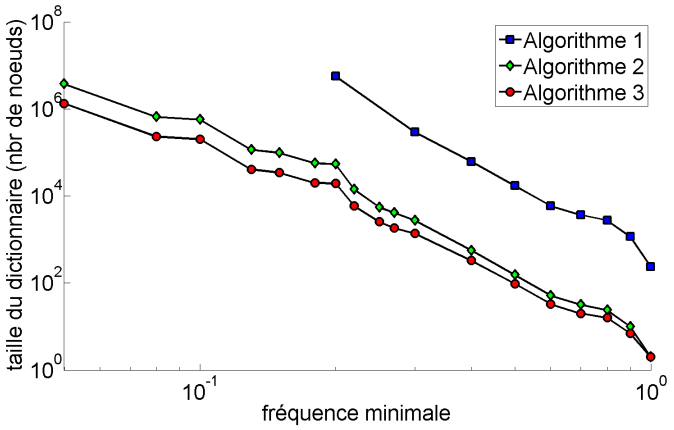


FIG. 5.14: Taille mémoire du dictionnaire exprimé en nombre de nœuds (échelle log-log) pour le jeu de données MUSHROOM

On constate par ailleurs que le nombre de candidats MPIs issus de l'étape de filtrage primaire de l'algorithme n° 3 n'est pas une fonction croissante de f_{min}, contrairement à toutes les autres courbes représentées sur la figure 5.13. À supposer en effet qu'un motif M candidat à être des plus informatifs pour un certain seuil $f_{min} = f$ soit dominé par un successeur immédiat M' de fréquence $\text{freq}(M') < f$ et est donc éliminé lors du filtrage secondaire. Lorsque le seuil f_{min} diminue au point de devenir inférieur à $\text{freq}(M')$, le motif M est directement éliminé par le filtrage primaire et ne figure plus parmi les motifs candidats. Ce phénomène explique de la phase de croissance suivie de celle de décroissance des nombres de motifs candidats lorsque f_{min} décroît. Dans tous les cas le temps de calcul nécessaire pour réaliser le filtrage secondaire (cf figure 5.12) est en toute logique proportionnel au nombre de motifs candidats, qui est par ailleurs faible, de l'ordre du double du nombre des motifs des plus informatifs.

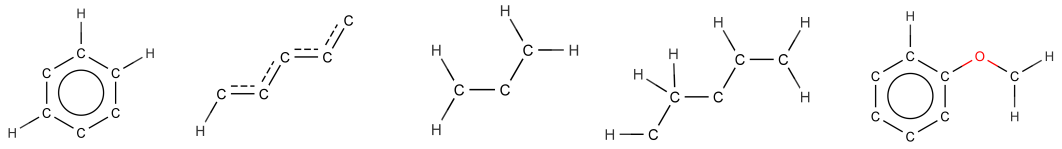
Enfin concernant le passage à l'échelle (i.e. scalability) des algorithmes, la figure 5.14 donne la consommation mémoire maximale atteinte par chacun des algorithmes, exprimée en nombre de nœuds utilisés par le dictionnaire de motifs. Comme prévu, l'algorithme n° 1 qui stocke les motifs fréquents mais aussi ceux très nombreux de la FNE, s'avère très consommateur de mémoire. Mais contrairement aux attentes, la différence de consommation mémoire entre les algorithmes n° 2 et n° 3 apparaît minime. Ce phénomène s'explique par le fait que tout sous-motif inclus dans un motif d'attribut fréquent forme lui-même motif fréquent. Dans le cas de l'algorithme n° 2, l'arbre préfixé $\mathcal{T}$ sert à stocker tous les motifs fréquents. Par conséquent, tous les nœuds de l'arbre préfixé $\mathcal{T}$ qui servent d'embranchements intermédiaires servent aussi à stocker un motif fréquent, de sorte que tous les nœuds de $\mathcal{T}$ sont utiles. Au contraire, dans le cas de l'algorithme n° 3, l'arbre $\mathcal{T}$ stocke un seul niveau de motifs fréquents à la fois. Mais lorsque la longueur l des motifs stockés augmente, il devient nécessaire de créer de nombreux nœuds intermédiaires supplémentaires, ce qui a pour effet de rendre l'algorithme n° 3 presque autant consommateur de mémoire que l'algorithme n° 2.

En résumé, il apparaît que l'algorithme de filtrage des motifs plus fréquents est plus efficace et « passant mieux à l'échelle » que l'algorithme d'extraction directe, du moins dans le cas de motifs d'attributs, même si le gain apparaît relativement faible et fluctuant d'un jeu de données à un autre.

5.5 Application à la fouille de schémas de réactions

5.5.1 Introduction

L'idée des motifs les plus informatifs est née du besoin de décrire le contenu des bases de données de réactions par un nombre réduit de schémas de réactions caractéristiques. Or la donnée des motifs fréquents ne suffisait pas à satisfaire ce besoin : un tri des motifs selon leur fréquence, ni même selon un score comme l'information, ne permettait de réduire significativement ni le nombre de motifs à analyser, ni la redondance d'information entre motifs. À titre d'exemple, la figure 5.15 présente un résumé des 1000 premiers schémas dégénérés extraits du jeu B (introduit à la section 4.7.2) et triés par ordre décroissant d'information (pour un seuil f_{min} de 2 %). Le premier motif (cf figure 5.15(a)), qui présente la plus grande information (84



(a) Rang 1 (b) Rang 174 (c) Rang 372 (d) Rang 652 (e) Rang 909

Fig. 5.15: Résumé des 1000 premiers schémas dégénérés triés par ordre décroissant d'information

bits), est pertinent puisqu'il s'agit du cycle benzénique omniprésent en synthèse organique. La présence d'atomes d'hydrogène sur le cycle indique ainsi que dans la majorité des cas, les atomes de carbone d'un tel cycle ne sont pas tous connectés au squelette[43] de la molécule. La

[43]Le *squelette* d'une molécule est son « ossature » constituée de tous ces cycles et de tous ces atomes de carbone (complétés par les éventuels hétéroatomes nécessaires à la connexion des fragments de carbone) et sur laquelle viennent se greffer les groupes fonctionnels en périphérie.

configuration la plus représentative semble être celle où seulement trois atomes de carbone non adjacents du cycle sont connectés au reste du squelette. La satisfaction procurée par l'intérêt du premier motif s'estompe cependant rapidement. En effet le second motif s'avère être une pâle copie du premier motif, dont la seule différence est de présenter deux atomes d'hydrogène en périphérie du cycle au lieu de trois. Il en va de même des schémas de rang 3, 4, 5 et 6 qui varient uniquement par le nombre d'atomes d'hydrogène et leur position autour du cycle. Le $7^{\text{ème}}$ motif est moins intéressant encore puisqu'il correspond au premier motif dans lequel on aurait ouvert le cycle en supprimant une des liaisons aromatiques. Les motifs qui suivent, à l'image du motif de la figure 5.15(b), sont tous des combinaisons variées de liaisons aromatiques et parfois simples entre atomes de carbone et d'hydrogène et qui n'apportent rien de plus que le premier motif. Il faut attendre le $372^{\text{ème}}$ motif avec une information de 31 bits pour percevoir un motif de nature différente (cf motif de la figure 5.15(c)), qui ne soit plus un fragment de cycle aromatique. Puis il faut attendre le $909^{\text{ème}}$ motif avec une information de 19 bits (cf motif de la figure 5.15(e)) pour voir enfin apparaître le premier hétéroatome (ici un atome d'oxygène). Le choix de la fonction de score peut certes être remis en question à la faveur d'une fonction qui traduirait mieux l'intérêt que peuvent avoir les chimistes pour tel ou tel fragment de graphe moléculaire ou tel ou tel schéma de réaction (par exemple en valorisant davantage les cycles et les groupes fonctionnels). Mais ce changement ne résout pas le problème essentiel de la dispersion des motifs pertinents au sein d'une masse de motifs similaires. L'élimination des motifs non fermés de la liste des motifs triés ne supprime pas plus le problème de la redondance, même si elle l'atténue très légèrement : ainsi le motif de la figure 5.15(e) est $858^{\text{ème}}$ au lieu d'être $909^{\text{ème}}$.

Fort de ce constat, le modèle des motifs les plus informatifs a été développé pour permettre une sélection des motifs efficace et qui élimine la redondance observée entre les motifs. Il est important toutefois de préciser que les schémas réactionnels les plus informatifs ne peuvent pas se confondre rigoureusement avec les schémas génériques de méthodes de synthèses. Pour qu'il en soit ainsi, de nombreuses conditions doivent être réunies. D'abord cela supposerait que l'on puisse disposer d'ensembles de réactions qui soient une image fidèle de la pratique et de la connaissance qu'ont les chimistes en synthèse organique. Ensuite et surtout, cela supposerait qu'on puisse disposer d'une fonction de score spécifique qui qualifie précisément ce qu'est une méthode de synthèse. Enfin, puisqu'une méthode de synthèse se caractérise aussi par ses conditions réactionnelles (catalyseurs, solvants ...), cela supposerait d'intégrer ces conditions réactionnelles d'une manière ou d'une autre dans le processus de fouille. Si on peut envisager sur le long terme de surmonter ces obstacles un à un, l'objectif à plus court terme qui est abordé ici est d'identifier un nombre réduit de schémas de réactions significatifs qui soient partagés par un nombre important de réactions. À moyen terme, le modèle des motifs les plus informatifs peut permettre une classification automatique et hiérarchique des réactions. Il suffit pour cela de regrouper les réactions incluant un même schéma des plus informatifs en une famille de réactions et de réitérer l'extraction des motifs les plus informatifs sur ce sous-ensemble. En ce sens l'extraction des motifs les plus informatifs est une méthode de classification conceptuelle automatique, l'intension et l'extension des concepts étant respectivement le motif le plus informatif et l'ensemble des données décrites par le motif.

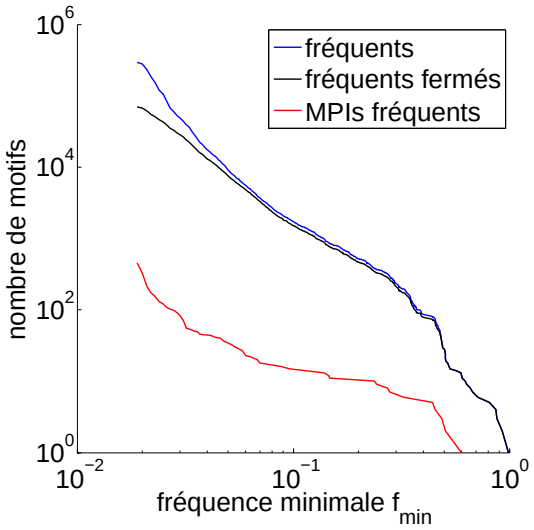
Le problème posé étant très général et sans a priori particulier, la fonction de score adoptée ici est la fonction d'information s_i introduite à la section 5.3.4. Par ailleurs, le prétraitement exposé au chapitre précédent (cf section 4.5) est réutilisé de sorte qu'en pratique, le problème traité est celui équivalent de l'extraction des graphes de réactions les plus informatifs contenus dans un ensemble de graphes de réactions. La suite de cette section présente les résultats obtenus, d'abord d'un point de vue macroscopique dans la section 5.5.2 puis d'un point de

vue plus qualitatif dans la section 5.5.3.

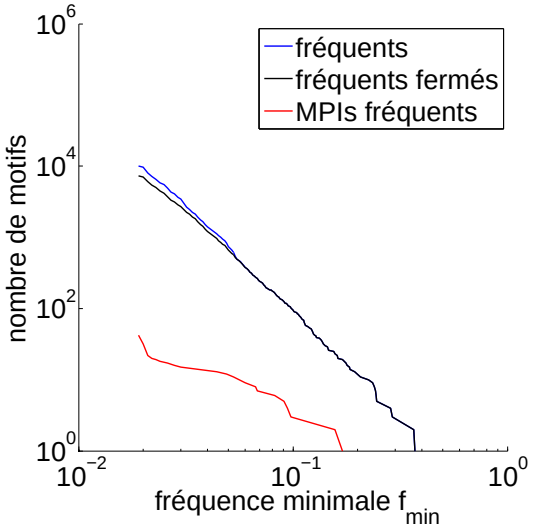
5.5.2 Analyse statistique

Les tests ont été réalisés sur les trois jeux de réactions déjà présentés au chapitre 4 en appliquant le même prétraitement des données que celui exposé en section 4.6.2. La méthode utilisée est celle de l'extraction par filtrage des motifs fréquents qui est la plus rapide et qui a été implémentée en C++ au sein du logiciel `Forage`. La méthode prend en entrée les graphes de réactions fréquents produits par `Gaston` et produit en sortie l'ensemble des graphes de réactions les plus informatifs fréquents et fermés fréquents, qui sont ensuite reconvertis en leurs schémas réactionnels équivalents. Les schémas dégénérés et non dégénérés sont ensuite séparés dans chacune des trois listes des schémas fréquents, fermés fréquents et des plus informatifs fréquents puis décomptés. Il est important de n'effectuer cette séparation qu'après filtrage et non pas avant. En effet un schéma non dégénéré peut très bien être dominé par un schéma dégénéré ou vice versa. Une séparation avant filtrage conduirait donc à ne pas détecter ce cas de figure et à augmenter artificiellement la liste des schémas des plus informatifs. Les résultats du décompte sont présentés sur la figure 5.16. Contrairement aux données Mushroom (cf section 5.4.3), les jeux de réactions présentent tous une grande diversité de motifs de sorte que les motifs fermés sont presque aussi nombreux que les motifs fréquents. Seuls les schémas non dégénérés du jeu C (cf courbe 5.16(f)) présentent un nombre de motifs fermés fréquents sensiblement inférieur à celui du nombre de motifs fréquents. Ce phénomène s'explique par le fait que contrairement aux autres jeux de données, le jeu C regroupe des réactions d'une même méthode de synthèse ce qui a pour effet de réduire la diversité des schémas non dégénérés qui y sont présents. Mais le point le plus important à observer est que les schémas les plus informatifs fréquents sont typiquement 1000 à 10000 fois moins nombreux que les schémas fréquents. L'allure de la courbe des motifs des plus informatifs a toutefois une allure différente de celles des motifs fréquents ou fermés fréquents. Comparativement à ces deux dernières courbes, la courbe des MPIs semble être décalée vers la gauche, c'est-à-dire vers les fréquences plus basses. Autrement dit, alors que le motif le plus fréquent parmi les motifs fréquents ou fermés fréquents a forcément une fréquence relative de 1, le MPI le plus fréquent apparaît à une fréquence plus basse, ici de 0,7. La courbe des MPIs présente généralement une croissance régulière et très lente jusqu'à ce qu'on observe une brusque remontée du nombre de MPIs, la courbe 5.16(c) faisant figure d'exception à cette règle. Ce rebond pour le moins étrange peut provenir d'une erreur d'implémentation commise lors du filtrage secondaire. Toutefois aucune anomalie n'a pu être détectée et ce phénomène reste pour l'heure non élucidé.

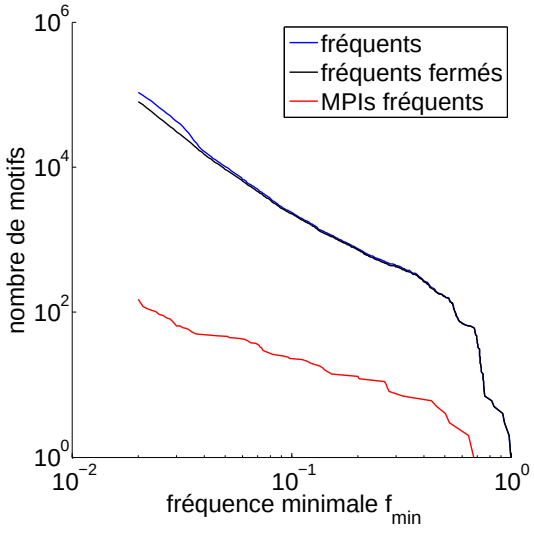
Si on s'intéresse à la distribution de ces motifs en fonction non seulement de leur fréquence mais aussi de leur information, on obtient les nuages de points représentés sur la figure 5.17. La rareté des motifs non fermés fréquents (losanges bleus) indique que la plupart des motifs fréquents sont fermés (carrés noirs). Les motifs les plus informatifs fréquents (disques rouges) sont dispersés au milieu du nuage des motifs fréquents, prouvant ainsi que les motifs les plus informatifs ne se concentrent pas dans les zones de scores les plus élevés mais qu'ils se répartissent de façon homogène parmi les motifs fréquents. Les motifs les plus informatifs semblent certes se concentrer dans la zone de faible fréquence et de faible information mais la concentration des MPIs n'y est toutefois pas plus élevée dans cette zone que ne le sont les motifs fréquents et fermés fréquents (Cette impression est due au fait que les points associés aux motifs fermés fréquents sont si denses qu'on ne peut plus les discerner à l'œil nu et paraissent donc relativement moins nombreux qu'ils ne sont en réalité).



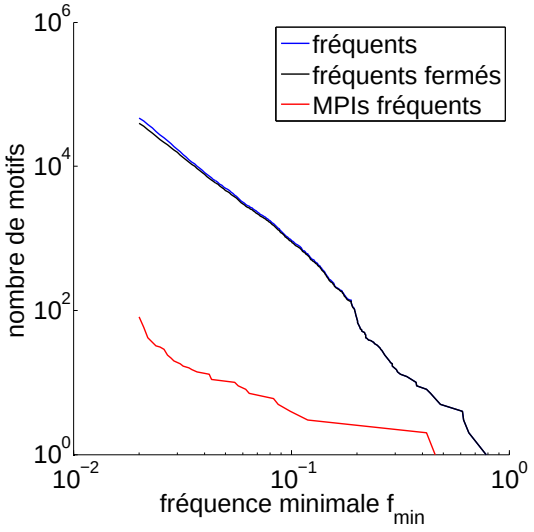
(a) Données A, schémas dégénérés



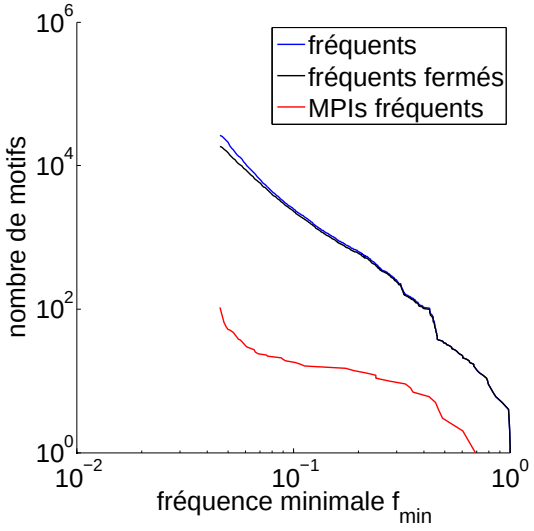
(b) Données A, schémas non dégénérés



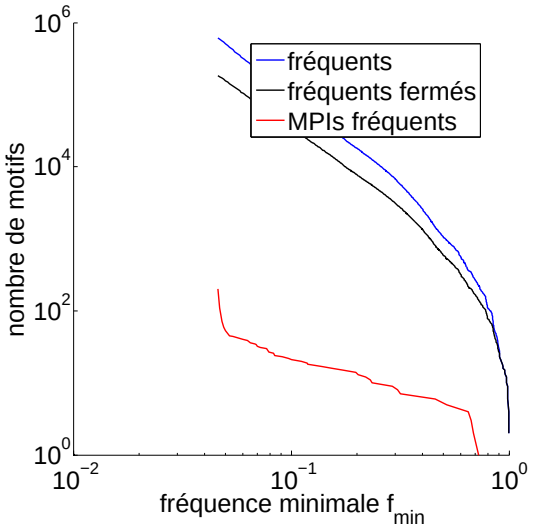
(c) Données B, schémas dégénérés



(d) Données B, schémas non dégénérés

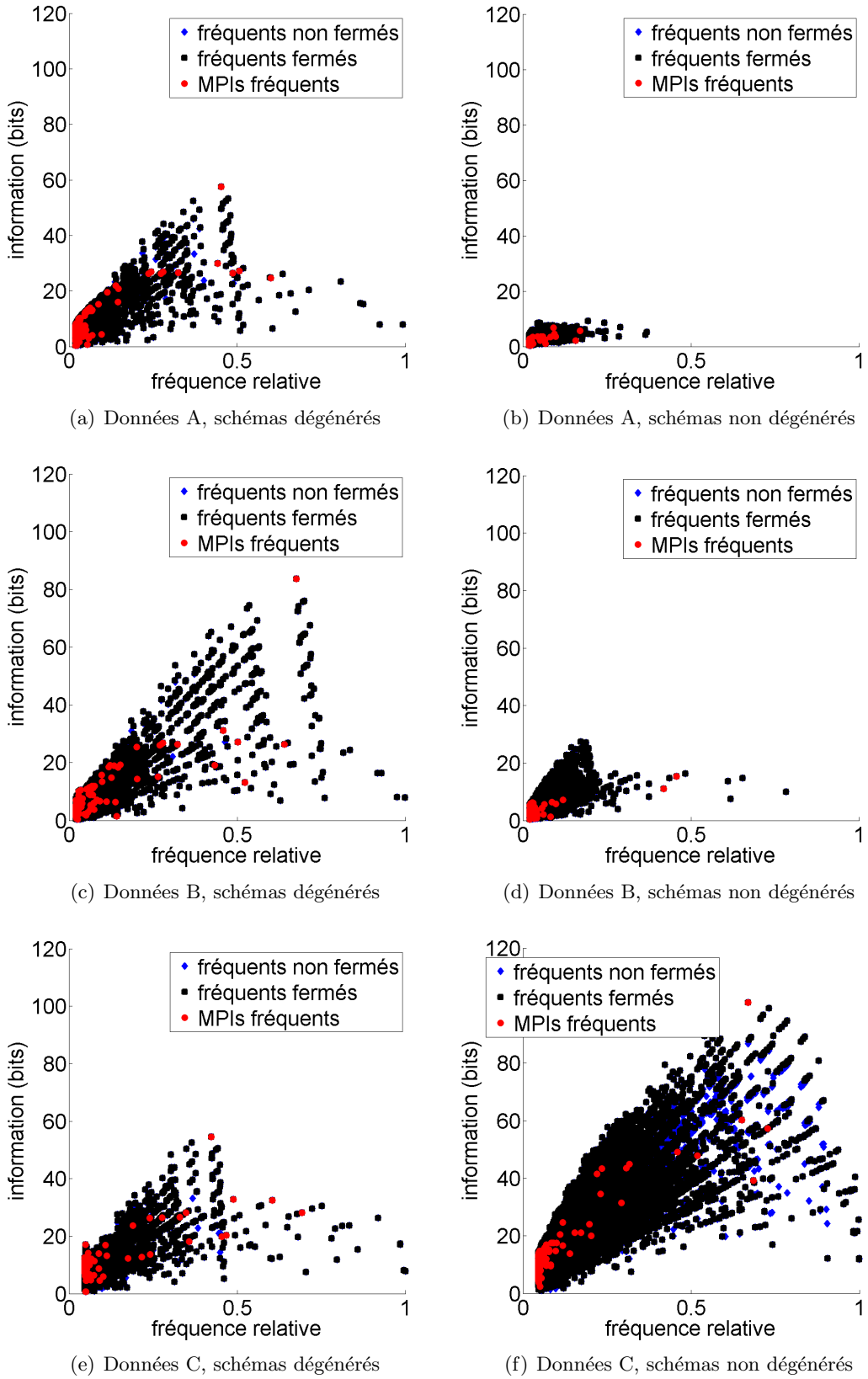


(e) Données C, schémas dégénérés



(f) Données C, schémas non dégénérés

FIG. 5.16: Nombres des motifs fréquents, fermés fréquents et des plus informatifs fréquents (échelle logarithmique)



(a) Données A, schémas dégénérés

(b) Données A, schémas non dégénérés

(c) Données B, schémas dégénérés

(d) Données B, schémas non dégénérés

(e) Données C, schémas dégénérés

(f) Données C, schémas non dégénérés

FIG. 5.17: Répartition des motifs fréquents, fermés fréquents et des plus informatifs fréquents dans le plan fréquence × information

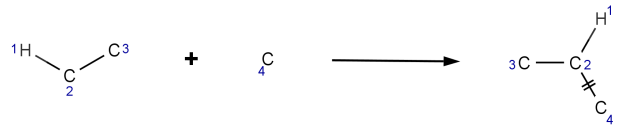
Ces nuages de points permettent aussi de vérifier que le schéma qui détient l'information la plus élevée est toujours un motif des plus informatifs. Ce schéma peut être tantôt dégénéré, tantôt non dégénéré selon la nature des données. Ainsi la prédominance des motifs dégénérés dans les jeux A et B, déjà constatée au chapitre 4, conduit à des niveaux d'information en moyenne plus élevés pour les motifs dégénérés (i.e. les nuages 5.17(a) et 5.17(c) des motifs dégénérés atteignent des scores de 60 et 80 bits quand les nuages 5.17(b) et 5.17(d) des motifs non-dégénérés ne sont que de 10 et 30 bits). Dans le cas du jeu C constitué d'exemples de la méthode de Diels-Alder, la tendance est inverse et les scores les plus élevés sont aux environs d'une centaine de bits (cf nuage 5.17(f)) pour des schémas similaires au schéma général de la méthode de Diels-Alder. Si l'amplitude de l'information varie d'un jeu de données à l'autre, les nuages de points des schémas dégénérés semblent superposables à un facteur d'échelle près : certains points peuvent ainsi être appariés d'un nuage de points à un autre en ne tenant compte que de leur position relative dans leur nuage de point. Après vérification, les couples de points appariés correspondent à des couples de motifs identiques. En particulier on observe dans les trois nuages de points associés aux motifs dégénérés, un alignement similaire presque vertical de motifs pour des fréquences relativement élevées, de l'ordre de 0,7 sur la figure 5.17(c). Après examen des motifs associés, ces alignements de points correspondent aux fragments du cycle benzénique qui saturaient déjà la tête de la liste des motifs fréquents triés étudiée à la section 5.5.1. Cet alignement vertical s'explique par le fait que leur fréquence reste approximativement la même que celle du cycle benzénique alors que leur structure plus pauvre que celle du cycle fait baisser leur information. Cette superposition des nuages de points va dans le sens de l'hypothèse déjà faite à la fin du chapitre 4 selon laquelle les schémas dégénérés présentent la même distribution statistique dans les trois jeux de données pourtant indépendants.

La figure 5.17 fait également apparaître d'autres alignements surprenants de points, qui se regroupent cette fois-ci sur des faisceaux de droite passant par l'origine. Ces regroupements correspondent à l'ensemble de schémas fréquents qui ont le même ensemble d'atomes – en chimie, qui ont même formule brute – et le même ensemble de liaisons mais agencées différemment. L'information $I = I(M)$ portée par la structure du motif est donc la même pour chacun des motifs. La fonction de score $s_i(M) = I(M) \times \text{freq}_r(M)$ étant le produit du terme constant I par la fréquence variable du motif, les points associés au groupe de motifs se trouvent alignés selon une droite $y = I \times x$ passant par l'origine. La figure 5.18 illustre ce phénomène. Les trois schémas représentés sont issus des schémas de réactions fréquents dans le jeu B. Chacun de ces schémas est constitué de 3 atomes de carbone, d'un atome d'hydrogène, de deux liaisons simples stables et d'une liaison simple créée. Leur information de 33,7 bits correspond à la pente de la droite sur laquelle ces points se trouvent.

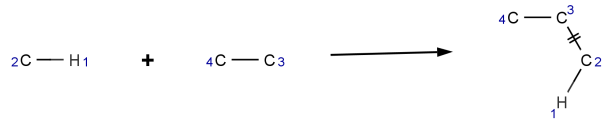
L'analyse statistique des résultats permet de tirer les premières conclusions quant aux propriétés des schémas les plus informatifs fréquents : ces derniers, qu'ils soient dégénérés ou non, apparaissent bien comme peu nombreux au regard du nombre de schémas fermés fréquents. Ces motifs sont par ailleurs « prélevés » dans l'ordre des schémas de réactions à des niveaux variés de score, de fréquence ou de taille. Ce dernier point suggère une redondance réduite entre structures de motifs, aspect qui est davantage développé dans la section suivante ; à travers une analyse visuelle qualitative des résultats.

5.5.3 Analyse qualitative

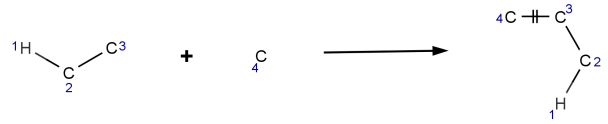
Une analyse qualitative a également été réalisée pour apprécier la pertinence des schémas de réactions les plus informatifs ainsi que la redondance d'information que ces motifs



(a) $s_i(M_1)/\,\mathrm{freq}(M_1) = 16,3/0,48 = 33,7$



(b) $s_i(M_2)/\,\mathrm{freq}(M_2) = 15,4/0,46 = 33,7$



(c) $s_i(M_3)/\,\mathrm{freq}(M_3) = 14,7/0,44 = 33,7$

FIG. 5.18: Exemples de schémas alignés dans le plan fréquence × information.

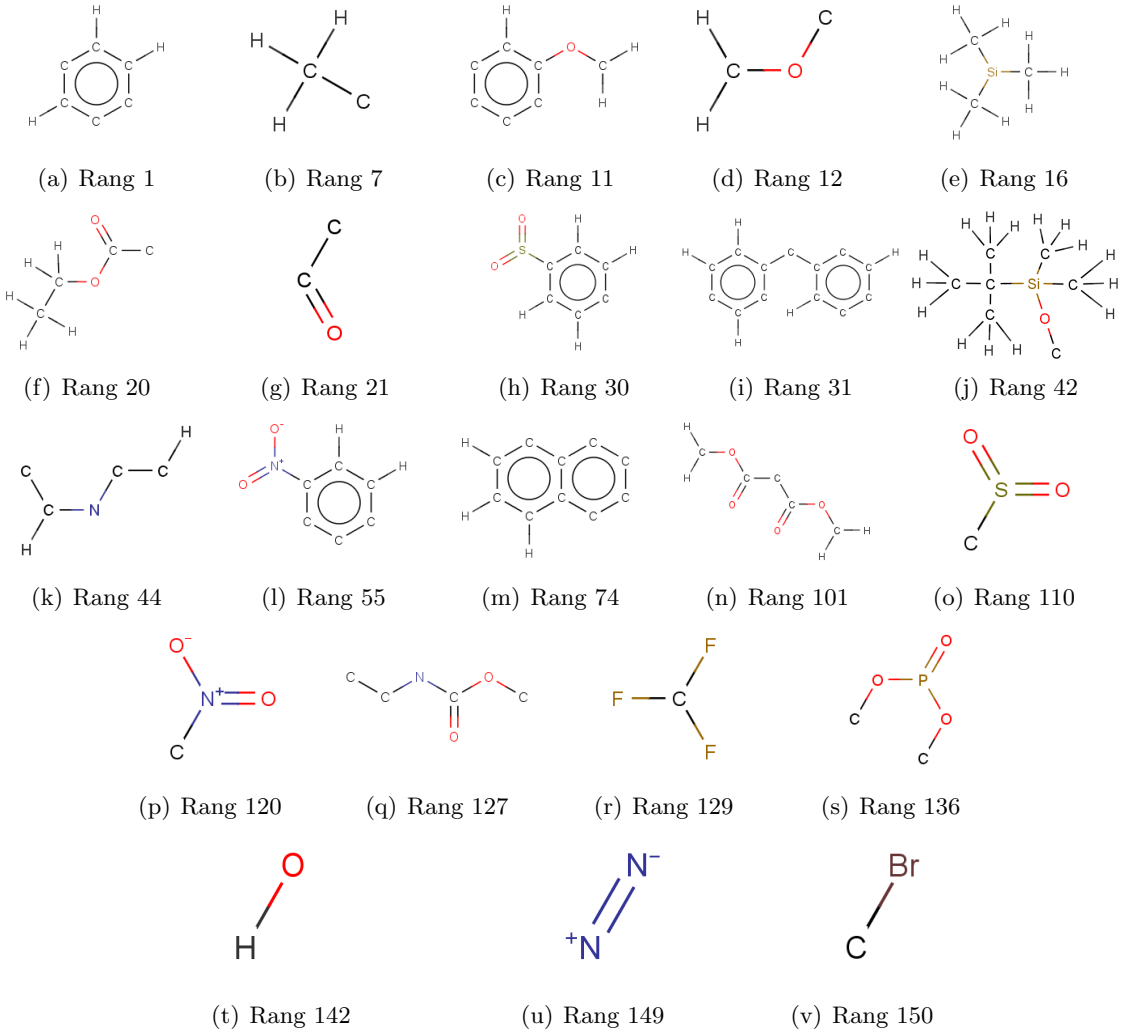
présentent vis-à-vis des schémas fréquents ou fermés fréquents. Le principe général de cette analyse consiste à comparer l'intérêt et la diversité (i.e. la non-redondance des motifs) des listes L_f, L_c et L_i des « top-k » schémas fréquents, fermés fréquents et des plus informatifs fréquents triés par ordre décroissant de score.

Analyse des schémas dégénérés

La première expérience reprend l'exemple de la section 5.5.1 des schémas dégénérés du jeu de données B extrait de REFLIB. Cette expérience considère la liste L_i des schémas dégénérés des plus informatifs fréquents. Parmi les 150 schémas que comprend cette liste, 22 schémas ont été sélectionnés pour l'intérêt qu'ils présentent et en particulier lorsque ces schémas comportent un groupe fonctionnel identifié. Lorsque plusieurs motifs sont apparentés, un seul motif, celui de score le plus élevé, est retenu. Ce choix de 22 schémas est arbitraire et pourrait en inclure davantage jusqu'à 35 schémas mais difficilement au delà. Le restant des 115 motifs sont en effet soit des répliques très proches de motifs déjà sélectionnés, soit des fragments de squelette carboné sans grand intérêt. Les schémas sélectionnés sont présentés sur la figure 5.19 ainsi que leur rang r_i au sein de la liste L_i.

Il apparaît d'emblée que la complexité des schémas (i.e. leurs tailles, leurs particularités en terme de groupes fonctionnels ...) varie au fil de cette liste, sans croître ni décroître régulièrement. Cette observation est conforme aux attentes qui veulent que l'information exprime un compromis entre la taille et la fréquence d'un motif. Un chimiste reconnaîtra immédiatement dans cette liste un certain nombre de groupes fonctionnels ou de structures caractéristiques. Le tableau 5.20 précise pour chacun des motifs sélectionnés son score, sa fréquence et une brève description des groupes fonctionnels qui y sont reconnaissables.

Le premier schéma est nécessairement le schéma fréquent de plus grand score et s'apparente donc sans surprise au groupe phényl, en accord avec ce qui avait déjà été constaté à la section 5.5.1. Le tableau précise par ailleurs les rangs r_f et r_c des schémas les plus infor-



(a) Rang 1 (b) Rang 7 (c) Rang 11 (d) Rang 12 (e) Rang 16

(f) Rang 20 (g) Rang 21 (h) Rang 30 (i) Rang 31 (j) Rang 42

(k) Rang 44 (l) Rang 55 (m) Rang 74 (n) Rang 101 (o) Rang 110

(p) Rang 120 (q) Rang 127 (r) Rang 129 (s) Rang 136

(t) Rang 142 (u) Rang 149 (v) Rang 150

FIG. 5.19: Sélection de 22 schémas parmi les 150 schémas dégénérés des plus informatifs fréquents, triés par ordre décroissant d'information

Rang r_i des MPIs fréq.	Information (bits)	Fréquence abs.	rel.	Rang r_c des fermés fréq.	Rang r_f des motifs fréq.	Description du motif
1	83,6	4756	67 %	1	1	Groupe phényle (+3 H)
7	26,3	4499	64 %	406	438	Groupement éthyle
11	19,2	856	12 %	858	909	Groupe éther (méthoxyphényle)
12	19,1	3045	43 %	859	910	Groupe éther (+ évent. ester)
16	15,8	679	10 %	1285	1353	Groupe triméthylsilyle
20	13,4	692	10 %	1942	2068	Groupe ester (éthylique)
21	13,2	3681	52 %	2001	2132	Groupement carbonyle
30	10,8	368	5 %	3205	3398	Groupe phénylsulfonyle
31	10,5	245	3 %	3465	3672	Groupe diphénylméthyle
42	7,05	141	2 %	12257	20272	Éther silylé
44	6,72	649	10 %	14264	24039	Amine second. ou tertiaire
55	5,56	220	3 %	25221	40472	Groupe nitrophényle
74	5,26	173	2 %	30506	46780	Groupe naphtalényle
101	4,34	172	2 %	49933	69042	Di-ester 1,1
110	3,73	527	8 %	63119	84904	Groupe sulfonyle
120	2,91	391	6 %	73470	98512	Groupe nitro
127	2,31	196	3 %	77069	103496	Uréthane
129	2,17	321	5 %	77559	104238	Trifluorométhyle ou tétrafluoro
136	1,56	144	2 %	78824	105917	Ester phosphonique
142	1,5	1001	14 %	78873	105976	Groupe alcool
149	0,64	151	2 %	79114	106303	Groupe diazo
150	0,38	178	3 %	79126	106317	Groupe bromure
150		7029	100 %	79126	106317	Total

FIG. 5.20: Sélection de 22 schémas parmi les 150 schémas dégénérés des plus informatifs fréquents, triés par ordre décroissant d'information.

matifs respectivement dans la liste L_f des motifs fréquents – déjà étudiée en introduction à la section 5.5.1 – et dans celle L_c des motifs fermés fréquents[44] triés par ordre décroissant d'information. On observe une augmentation très rapide des rangs r_f et r_c, qui traduit la dispersion des motifs les plus informatifs au sein des motifs fermés fréquents du fait de la redondance des structures de motifs. Réciproquement l'étude des mille premiers motifs de L_c conduit à sélectionner 4 motifs qui sont ou s'apparentent aux quatre premiers motifs des plus informatifs. La densité de motifs intéressants et non redondants est donc beaucoup plus importante dans la liste des MPIs que dans celle des motifs fermés (i.e. $22/150 = 15\%$ au lieu de $4/1000 = 0,4\%$) sans que cela implique une perte apparente d'information.

Du point de vue purement applicatif, les motifs les plus informatifs ne correspondent pas toujours exactement à une liste de groupes fonctionnels comme pourraient le souhaiter les chimistes. Ainsi le groupement carbonyle $C = O$, représenté par le motif de rang 21, n'est pas à proprement parler un groupe fonctionnel. Ce groupement peut en effet participer à différents groupes fonctionnels, en tant que fragment des groupes cétone $(C-(C=O)-C)$, ester $(C-(C=O)-O-C)$, aldéhyde $(C-(C=O)-H)$, carboxyle $(C-(C=O)-OH)$ ou amide $(C-(C=O)-N)$. Comme ces groupes sont toutefois structurellement très proches, ils contribuent à augmenter le score du groupement carbonyle qui les domine. L'émergence du carbonyle en tant que motif des plus informatifs se fait au détriment des groupes cétones, aldéhyde ou amide, qui ne figurent pas parmi les motifs les plus informatifs. On observe

[44]Le rang r_c est forcément défini puisqu'un MPI est aussi un motif fermé.

la même ambiguïté pour le groupe éther $C-O-C$ représenté par le motif de rang 12. Ainsi sur les 9963 occurrences de ce 12$^{\text{ème}}$ motif, 3975 sont dues à la présence d'un groupe ester. Puisque ces groupes sont dissociés du point de vue de leurs propriétés chimiques, la contribution du groupe ester à la fréquence du groupe éther ne devrait pas être comptabilisée. La décomposition des graphes moléculaires en groupes fonctionnels permettrait d'éviter ce phénomène.

Analyse des schémas non dégénérés

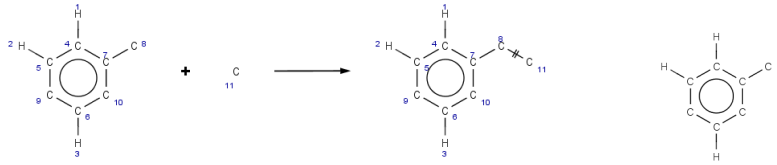
Le même exercice a été effectué concernant les schémas non dégénérés. Les caractéristiques (information, fréquence, rangs) des 24 schémas sélectionnés sont précisées dans le tableau de la figure 5.21. Les schémas associés sont représentés sur les figures 5.23 et 5.24.

Conformément à l'analyse de la distribution statistique des MPIs (cf section 5.5.2), les scores des schémas non dégénérés apparaissent globalement plus faibles que ceux des schémas dégénérés. De ce fait et contrairement aux schémas dégénérés, le premier MPI n'est pas le premier mais seulement le 251$^{\text{ème}}$ schéma non dégénéré dans la liste L_c des schémas fermés fréquents, avec une information de 15 bits. En effet, les 250$^{\text{ers}}$ schémas qui le précèdent dans L_c s'avèrent être dominés par des schémas dégénérés. Par exemple, le premier schéma M_1 en tête de L_c, représenté sur la figure 5.22(a), obtient une information de plus de 27 bits, soit bien plus que le MPI de rang 1. Mais cette information lui vient en grande partie d'un motif dégénéré M_2 voisin dans l'ordre des motifs, représenté sur la figure 5.22(b). Le score de M_2 atteignant plus de 67 bits d'information, M_2 domine largement M_1, ce qui a pour conséquence d'éliminer ce dernier de la liste des motifs les plus informatifs. Cette élimination est souhaitable dans la mesure où l'information du schéma M_1 ne vient pas de la transformation chimique qu'exprime M_1 mais d'un fragment stable représenté par le motif M_2. Ce phénomène souligne la nécessité d'extraire les motifs non dégénérés les plus informatifs à partir de la liste de tous les schémas fréquents, y compris dégénérés, même si ces derniers peuvent ne pas présenter d'intérêt in fine dans le cadre de la fouille des BdR. On ne peut donc réaliser l'économie qui consisterait à extraire les motifs les plus informatifs de la liste réduite des schémas non dégénérés fréquents. Ce faisant, la dominance de M_2 sur M_1 ne serait plus détectée et le schéma M_1 apparaîtrait en tête de la liste des motifs des plus informatifs en tant que motif non dégénéré de plus grand score.

La plupart des schémas non dégénérés obtenus sont très généraux et décrivent de grandes familles de réactions regroupant de quelques centaines à quelques milliers d'exemples. Ainsi les deux premiers schémas (cf figures 5.23(a) et 5.23(b)) représentent chacun plus de 40 % des données. Certains schémas font apparaître des groupes fonctionnels déjà identifiés lors de l'analyse des schémas dégénérés comme le triméthylsilyle (cf figure 5.23(f)) ou le groupe ester (cf figure 5.23(h)). De nouveaux groupes fonctionnels apparaissent comme le groupe de l'acide boronique (cf figure 5.23(d)). Leur absence de la liste des schémas dégénérés suggèrent une forte réactivité ou du moins une implication presque systématique de ce groupe dans les réactions étudiées. Ainsi sur les 241 occurrences du groupe de l'acide boronique, toutes, sans exception, réagissent en se détachant d'un atome de carbone. La présence de groupes fonctionnels dans les schémas non dégénérés indique l'importance qu'ils revêtent en synthèse organique. Par ailleurs les transformations exprimées sont variées faisant apparaître majoritairement des créations de liaisons, qu'elles soient simples (cf figure 5.23(a)), doubles (cf figure 5.24(d)) ou aromatiques (cf figure 5.24(a)). D'autres schémas font également apparaître des modifications de certains types de liaisons comme par exemple le passage d'une liaison double à une liaison simple par addition (cf figure 5.23(c)). Mais aucun schéma ne comporte

Rang r_i des MPIs fréq.	Inform. (bits)	Fréquence abs.	rel.	Rang r_c des fermés fréq.	Rang r_f des motifs fréq.	Description du motif
1	15,4	3214	46 %	251	288	Création d'une liaison $C-C$
2	11,0	2941	42 %	965	1089	Substitution de $C-H$ en $C-C$
3	7,1	836	12 %	2962	3297	Addition sur une liaison $C=C$
5	6,2	195	3 %	4492	4999	Créat. de liaison $C-C$ (acide boronique partant)
7	6,0	221	3 %	4956	5505	Créat. de liaison $C=C$ (oléfination)
8	6,0	166	2 %	5111	5683	Créat. de liaison $C-C$ (via un énoxysilane)
9	5,8	694	10 %	5565	6185	Add. sur une liais. $C=O$ et créat. d'un alcool
10	5,8	186	3 %	5724	6364	Créat. de liaison $C-C$ (alkylation d'ester)
29	4,2	180	3 %	15505	18142	Créat. de liaison C-C entre deux cycles aromatiques
55	2,7	294	4 %	32373	38682	Add. d'hydrogène sur une double liais.
56	2,5	151	2 %	33652	40163	Créat. de liaison C-C (addition 1,4 sur une double liaison)
59	2,3	146	2 %	35255	41955	Créat. de 2 liaisons C-C suivant un processus en cascade
64	2,0	223	3 %	36718	43578	Créat. d'un cycle aromatique
67	1,9	141	2 %	37228	44133	Idem rang 59
69	1,8	303	4 %	37633	44570	Subst. de $N-H$ en $N-C$
70	1,7	154	2 %	37939	44900	Créat. d'une liais. $C=C$ en prés. d'un éther
72	1,6	140	2 %	38269	45265	Créat. d'une liaison C-C et N-C (alkylation d'imine)
75	1,5	177	3 %	38549	45567	Créat. d'une liaison C-C connexe à une fonction alcyne
76	1,4	583	8 %	38668	45690	Créat. d'une liaison $N-H$
77	1,2	153	2 %	38942	45983	Add. sur une l. $C=C$ en prés. d'un oxyg. en α
79	0,9	156	2 %	39141	46191	Subst. de $O-H$ en $O-C$
80	0,8	296	4 %	39166	46218	Créat. d'un cycle arom. azoté à partir d'une liais. simple
81	0,7	239	3 %	39197	46251	Créat. d'un cycle aromatique azoté
82	0,4	141	2 %	39214	46268	Créat. d'une double liaison $C=O$
82		7029	100 %	39214	46268	Total

FIG. 5.21: Sélection de 24 schémas parmi les 82 schémas non dégénérés des plus informatifs fréquents, triés par ordre décroissant d'information.



(a) $I(M_1) = 27$ (b) $I(M_2) = 67$

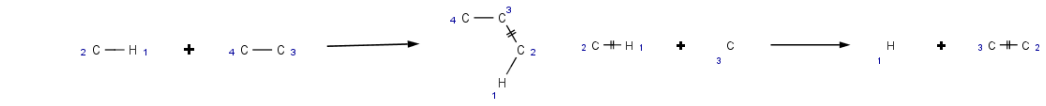
FIG. 5.22: Exemple de schéma M_1 non dégénéré dominé par un schéma M_2 dégénéré

de destruction de liaisons sans qu'elle ne soit associée à la création d'une autre liaison. Enfin l'ambiguïté de certains schémas constatée dans le cas des schémas dégénérés subsiste : ainsi le groupement carbonyle de la figure 5.23(e) peut participer à un groupe cétone ou carboxylique. De même l'oxygène du schéma de la figure 5.23(j) qui apparaît comme un groupe éther, est en réalité du pour moitié à l'ester de la figure 5.23(k).

L'analyse des motifs les plus informatifs peut être affinée en la réappliquant sur un sous-ensemble particulier des données. Pour obtenir une analyse plus fine d'une famille particulière de réactions caractérisée par un de ces schémas, il suffit ainsi d'extraire l'ensemble des réactions présentant le schéma considéré puis d'extraire les MPIs sur ce sous-ensemble. Ainsi l'extraction des MPIs non dégénérés fréquents sur le jeu C (qui regroupe des exemples de réactions de la méthode de synthèse de Diels-Alder) permet d'identifier des MPIs spécifiques à la méthode de Diels-Alder. Ces schémas permettent de caractériser des configurations récurrentes favorables à l'emploi de la méthode de Diels-Alder. La figure 5.26 représente 5 des 20 schémas sélectionnés, la liste complète étant précisée en annexe C.1. Les précisions associées à ces 20 schémas sont données au tableau de la figure 5.25. Le premier motif des plus informatifs correspond sans surprise au schéma général de la méthode de Diels-Alder. Les schémas qui suivent font apparaître des groupes fonctionnels qui définissent autant de variétés de la méthode de Diels-Alder, même si certains des schémas fournis en annexe C.1, comme les MPIs de rangs 33, 34, 41, 65, 79, etc, ne comportent parfois qu'un fragment du schéma caractéristique de la méthode de Diels-Alder.

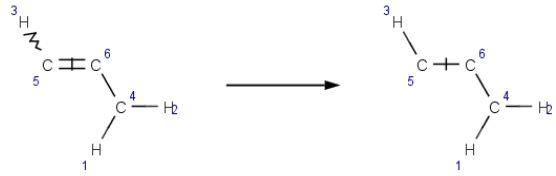
5.6 Conclusion

Le besoin d'extraire des schémas réactionnels représentatifs d'un ensemble de réactions a donné naissance au modèle des motifs les plus informatifs. Dans son formalisme le plus général, le modèle se définit à partir d'une famille quelconque de motifs ordonnés, qu'il s'agisse de motifs d'attributs, de graphes ou de schémas de réactions. Ce modèle concilie les avantages des méthodes de sélection dites inter- et intra-motif qui sont respectivement un échantillonnage régulier dans l'ordre des motifs et la possibilité de spécifier l'intérêt d'un motif grâce à une fonction de score. Le modèle permet ainsi de retenir un nombre réduit de motifs qui sont informatifs (i.e. descriptifs et représentatifs des données) et présentent peu d'information redondante relativement à la liste des motifs fréquents ou fermés fréquents triés par ordre décroissant de score. La possibilité de choisir la fonction de score permet en outre de guider l'extraction par une connaissance a priori. Le modèle bénéficie également de propriétés formelles intéressantes. En particulier tout motif des plus informatifs est un motif fermé et

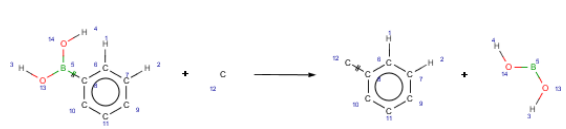


(a) Rang 1

(b) Rang 2



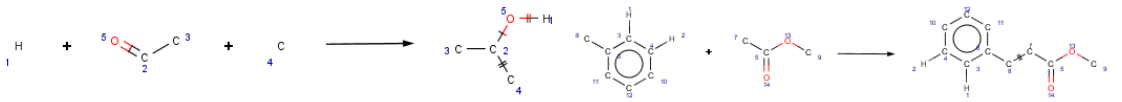
(c) Rang 3



(d) Rang 5

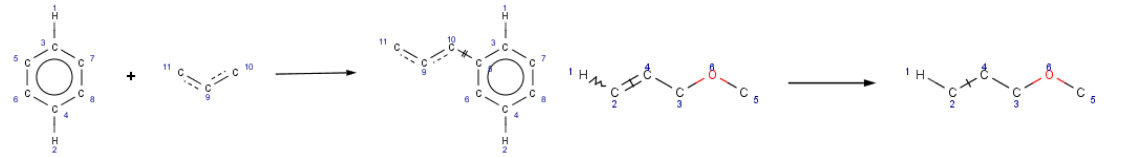
(e) Rang 7

(f) Rang 8



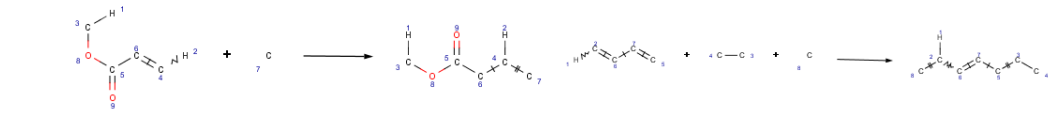
(g) Rang 9

(h) Rang 10



(i) Rang 29

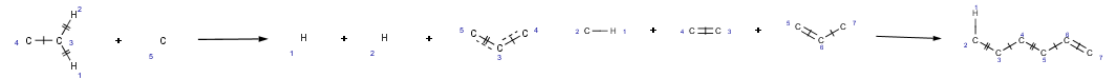
(j) Rang 55



(k) Rang 56

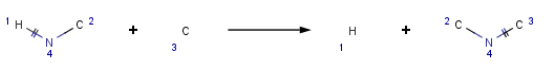
(l) Rang 59

FIG. 5.23: Sélection de 24 schémas parmi les 82 schémas non dégénérés des plus informatifs fréquents, triés par ordre décroissant d'information (du 1er au 12ème)



(a) Rang 64

(b) Rang 67



(c) Rang 69

(d) Rang 70

(e) Rang 72

(f) Rang 75

(g) Rang 76

(h) Rang 77

(i) Rang 79

(j) Rang 80

(k) Rang 81

(l) Rang 82

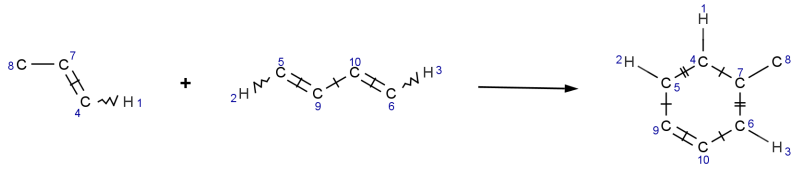
FIG. 5.24: Sélection de 24 schémas parmi les 82 schémas non dégénérés des plus informatifs fréquents, triés par ordre décroissant d'information (du 13$^{\text{ème}}$ au 24$^{\text{ème}}$)

Rang MPIs fréq. r_i	Information (bits)	Fréquence abs.	Description du motif
1	101	2174	Schéma général de la méthode de Diels-Alder
8	43,3	763	Groupe éther à 2 liaisons du centre
13	24,7	383	Groupe amide
31	14,8	162	Groupes méthyl environnants
33	14,4	249	Groupe phényl
34	14,3	156	Groupes amide + phényl
41	13,4	150	Groupe phényl + N
65	11,5	154	N
74	10,8	221	
79	10,7	152	
114	8,9	156	Éther de silyle
137	7,5	152	
143	7,1	155	Groupe éther à une liaison du centre
144	7,0	150	Groupe phényl + thioéther
146	7,0	150	Groupe N-C=O-O
158	6,7	150	
160	6,6	152	Groupe thioéther
170	6,2	151	Groupe sulfoxyde ?
182	5,6	152	Groupe éther à 4 liaisons du centre
198	4,4	151	Groupe sulfone
203		3248	Total

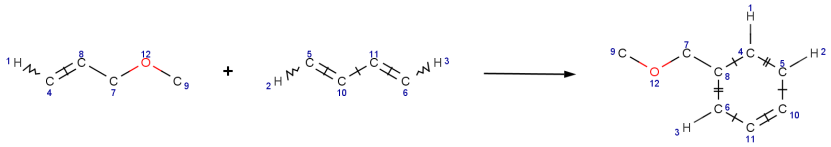
FIG. 5.25: Sélection de 20 schémas parmi les 203 schémas non dégénérés des plus informatifs fréquents, triés par ordre décroissant d'information, extraits du jeu C de réactions de Diels-Alder

les motifs fermés sont la famille la plus générale des motifs les plus informatifs. L'extraction des motifs les plus informatifs fréquents peut se faire directement ou à partir de l'ensemble des motifs fréquents, cette dernière méthode apparaissant plus rapide et légèrement moins coûteuse en ressources. En terme d'applications, l'extraction des schémas réactionnels les plus informatifs contenus dans les BdR a permis d'identifier des schémas caractéristiques de grandes familles de réactions, même si ces schémas ne correspondent pas exactement aux schémas des méthodes de synthèse identifiées par les chimistes. Le modèle des motifs les plus informatifs n'en demeure pas moins un outil d'extraction de connaissances intéressant qui permet d'appréhender le contenu des données.

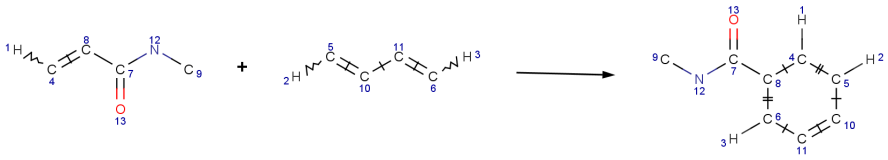
L'étude du modèle des motifs les plus informatifs doit maintenant être affinée. Les développements à venir passent ainsi par une évaluation quantifiée de la redondance entre motifs et l'intégration de connaissances du domaine dans les fonctions de scores et la modélisation des données. Au delà de ces questions d'approfondissement et d'ajustement, les motifs les plus informatifs posent, comme toutes les autres familles de motifs dérivées des motifs fréquents, un problème plus fondamental. En effet, il est impossible d'extraire les motifs des plus informatifs dont la fréquence est très faible. Dans le cas particulier des motifs les plus informatifs, le frein n'est pas tant un nombre trop important de motifs à analyser mais l'explosion combinatoire du nombre de motifs fréquents qu'il est nécessaire d'extraire puis de filtrer en amont, lorsque le seuil minimal de fréquence vient à décroître. Ce problème inhérent à toute fouille de données systématique, empêche l'utilisation des motifs les plus informatifs dans le cadre de certaines applications. Supposons en effet que l'on s'intéresse à une réaction particulière



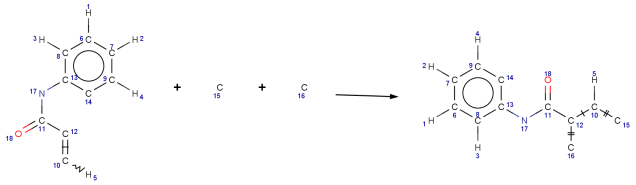
(a) Rang 1



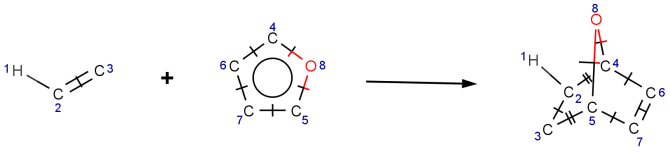
(b) Rang 8



(c) Rang 13



(d) Rang 34



(e) Rang 74

FIG. 5.26: Quelques schémas non dégénérés fréquents des plus informatifs représentatifs de la méthode de Diels-Alder

et que l'on cherche, en vue de sa classification, à identifier le schéma caractéristique de la méthode de synthèse sous-jacente à cette réaction. Supposons en outre que la détermination de ce schéma caractéristique soit un problème approché sous l'angle de l'extraction des schémas les plus informatifs relativement à une fonction de score et à un ensemble d'exemples de réactions. Supposons enfin que le problème soit parfaitement soluble du point de vue théorique (i.e. que l'on soit capable de qualifier formellement ce qu'est une méthode de synthèse à l'aide d'une fonction adaptée). Se pose alors un problème pratique qui malgré toutes les hypothèses précédentes, rend impossible l'identification du schéma caractéristique. Ce dernier schéma peut en effet dans certain cas être partagé par un nombre très faible d'exemples de réactions, typiquement une dizaine d'exemples sur des milliers voire des dizaines de milliers de réactions. À un seuil de fréquence aussi bas, il est difficile, pour ne pas dire impossible, d'extraire les schémas des plus informatifs fréquents à l'aide des algorithmes présentés dans ce chapitre. Ce constat a motivé le développement d'une autre approche pour résoudre ce problème, fondée sur une recherche heuristique et développée au chapitre suivant.

Chapitre 6

Méthode heuristique d'apprentissage transductif à partir de graphes. Application à l'extraction des schémas caractéristiques de méthodes de synthèse

Sommaire

6.1 Introduction

Ce chapitre traite d'une approche originale de fouille de graphes, conçue pour servir l'application déjà évoquée dans les chapitres précédents, de l'extraction des schémas génériques

caractéristiques de méthodes de synthèse et abrégés « schémas CMS ». En comparaison avec le modèle des motifs les plus informatifs du chapitre précédent, l'approche développée dans ce chapitre, apporte la possibilité de chercher certains motifs de très faible fréquence dans de très grandes bases de données et ce, grâce à une approche qualifiée de « transductive », par analogie avec certaines méthodes d'apprentissage dit transductif. Ce principe, lorsqu'il est appliqué au problème de l'extraction de schémas CMS, conduit à introduire une contrainte supplémentaire dite d'intervalle de graphes, dans le processus de recherche des motifs de score maximal.

L'objet de cette introduction est de présenter les raisons qui ont conduit à concevoir cette méthode, que ces raisons soient propres au domaine d'application, à savoir l'extraction de connaissances appliquée à la synthèse organique, ou à des considérations purement informatiques. Ainsi la section 6.1.1 développe le problème de l'extraction des schémas CMS à travers la difficile question de la définition formelle de ces schémas. La section 6.1.2 s'intéresse ensuite au problème informatique que pose l'extraction des schémas CMS, à savoir l'extraction de motifs de très faible fréquence, et envisage pour le résoudre, de recourir aux méthodes de recherche heuristique dans un espace d'état. La section montre toutefois que l'utilisation des recherches heuristiques existantes n'est pas adaptée tant l'espace de recherche associé est grand. La section 6.1.3 propose en conséquence de formuler différemment le problème afin de contraindre l'espace de recherche : au lieu de chercher à extraire tous les schémas CMS présents dans une BdR, le nouveau problème consiste à extraire le schéma caractéristique de la méthode de synthèse sous-jacente à une réaction donnée. Le problème ainsi reformulé, qui présente des similitudes avec l'apprentissage transductif, conduit au problème de la recherche du motif optimal dans un intervalle de graphes, dont la formulation et la résolution sont précisées dans les sections suivantes. Les travaux présentés dans ce chapitre sont par ailleurs résumés dans l'article Pennerath *et al.* (2008b).

6.1.1 La difficile définition des schémas caractéristiques de méthodes de synthèse.

Comme le chapitre 1 a pu l'expliquer, les méthodes de synthèse sont des éléments de connaissances essentiels en synthèse organique. À ce titre, l'extraction automatique des schémas caractéristiques de ces méthodes de synthèse à partir des BdR est un problème important de la synthèse organique assistée par ordinateur. Les chapitres 4 et 5 se sont intéressés à la fouille de schémas réactionnels à partir d'exemples d'équations chimiques. Les motifs les plus informatifs introduits au chapitre 5 ont ainsi permis d'identifier des schémas réactionnels caractéristiques de grandes familles de réactions. Ces schémas ne sont toutefois pas caractéristiques de méthodes de synthèse au sens précis où l'entendent les chimistes. Pour comprendre cette distinction, il convient de développer davantage la notion de méthode de synthèse, à travers l'exemple déjà mentionné de la méthode de synthèse de Diels-Alder. Jusqu'à présent une méthode de synthèse a été présentée comme une transformation chimique commune à un grand nombre de réactions et caractérisée par un schéma de réaction unique. Les exemples de la méthode sont alors les réactions qui contiennent son schéma caractéristique. Le concept de méthode de synthèse, tel qu'il est perçu par les chimistes, est en fait beaucoup plus complexe et par conséquent plus difficile à définir. En réalité, les chimistes définissent une méthode de synthèse comme un « procédé réactionnel » réutilisable dans différents plans de synthèse, et dont le déroulement s'explique non pas par un schéma réactionnel unique mais par la coordination de différents effets chimiques, comme les effets inducteurs ou mésomères... Ainsi la méthode de Diels-Alder peut être décrite dans toute sa généralité, comme l'addition d'un

dième conjugué (i.e. deux doubles liaisons séparées par une liaison simple) à un alcène substitué diénophile (i.e. une molécule présentant une double liaison et une affinité pour le dième) pour former un cycle. Si on voulait réduire cette transformation à un schéma unique, le schéma résultant serait celui de la figure 6.1. La différence entre ce schéma et le schéma de

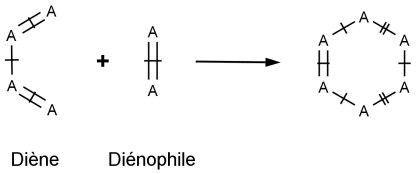
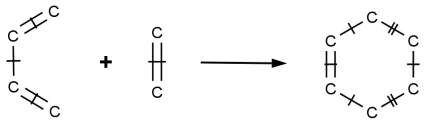
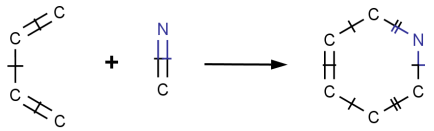


FIG. 6.1: Schéma général de la méthode de Diels-Alder

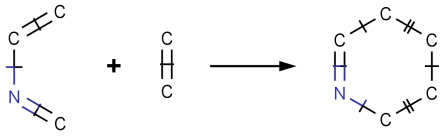
la figure 6.2(a) donné dans les chapitres précédents, est que les atomes de symbole A ne sont plus nécessairement des atomes de carbone (A signifiant ici « any »). Toutefois et contrairement à ce que suggère le schéma, seules quelques combinaisons d'atomes peuvent conduire à des transformations réalisables. La figure 6.2 donne différentes configurations d'atomes qui peuvent conduire à une réaction de type Diels-Alder. Une méthode de synthèse est donc un



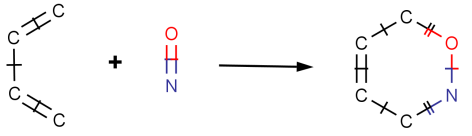
(a) Schéma « classique »



(b) Schéma d'« Aza-Diels-Alder »



(c) Autre schéma d'« Aza-Diels-Alder »



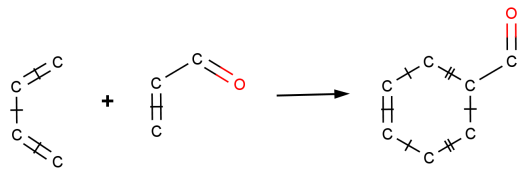
(d) Schéma comportant un groupe nitroso $-N=O$

FIG. 6.2: Schémas généraux de la méthode de Diels-Alder

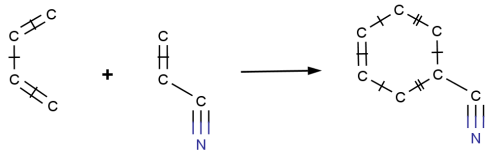
objet polymorphe qui ne peut pas de façon générale, être décrit par un seul schéma de réaction générique mais par plusieurs. Ces schémas n'étant pas inclus les uns dans les autres, correspondent à des sous-familles disjointes de réactions. Les chimistes répertorient toutefois ces sous-familles sous une même méthode de synthèse dans la mesure où ces réactions reposent sur des principes similaires.

Les schémas CMS présentent, outre le problème précédent de polymorphisme, une autre ambiguïté rendant leur caractérisation formelle encore plus difficile. Ainsi même dans le cas d'une méthode de synthèse n'admettant qu'un seul schéma caractéristique, la seule donnée

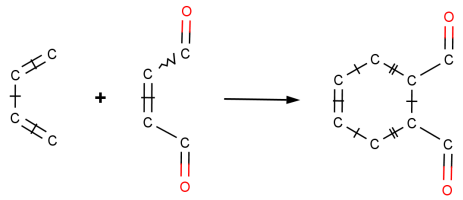
de ce schéma demeure une description très incomplète et en fait, inexploitable de la méthode de synthèse. À ce schéma doivent aussi s'ajouter les conditions réactionnelles, définies notamment en terme de catalyseurs, pour que la méthode puisse être mise en œuvre ; même si ces conditions sont des données extérieures à la description structurelle des réactions (i.e. à leurs équations chimiques) et ne sont donc pas davantage abordées dans ce mémoire dévolu à la fouille de graphes. Il existe toutefois d'autres éléments d'information qui sont contenus dans les équations chimiques des réactions associées à une méthode de synthèse et qui pourtant ne sont pas exprimés par les schémas caractéristiques généraux de la méthode : les schémas généraux peuvent en effet être avantageusement complétés par l'ajout de groupes fonctionnels qui sont nécessaires au bon déroulement de la réaction. De tels schémas complétés sont qualifiés dans ce qui suit de « *schémas caractéristiques étendus* » afin de les distinguer des *schémas caractéristiques généraux* à partir desquels ces schémas étendus sont construits. Par exemple, pour que la méthode de Diels-Alder puisse être appliquée, il faut que le réactant diénophile présente un groupe électroattracteur à proximité de sa double liaison, comme par exemple un groupe aldéhyde $C = O$, nitrile $C{\equiv}N$ ou halogène $C - Cl$, $C - F \ldots$ La figure 6.3 fournit des exemples de tels schémas où le diénophile est doté d'un groupe aldéhyde $C = O$ (figure 6.3(a)) ou nitrile $C{\equiv}N$ (figure 6.3(b)). L'effet favorable du groupe peut encore être augmenté



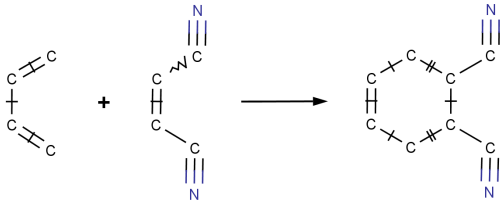
(a) Schéma avec un groupe aldéhyde



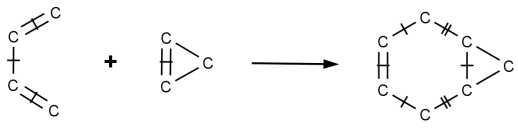
(b) Schéma avec un groupe nitrile



(c) Schéma avec deux groupes aldéhyde



(d) Schéma avec deux groupes nitrile



(e) Schéma avec un cyclopropène

FIG. 6.3: Exemples de schémas « étendus » de la méthode de Diels-Alder

si le groupe est dupliqué de part et d'autre de la double liaison, comme sur les schémas des figures 6.3(c) et 6.3(d). Enfin citons un schéma plus méconnu de la méthode de Diels-Alder

représenté sur la figure 6.3(e), dans lequel le diénophile est un cyclopropène (i.e. le cycle à 3 atomes de carbone), sans présence d'un groupe électro-attracteur. Ces observations montrent combien la représentation d'une méthode de synthèse par un schéma caractéristique unique constitue une approche grossière du problème de la caractérisation des méthodes de synthèse.

Contrairement aux schémas généraux de la figure 6.2 qui représentent des familles disjointes de réactions, les schémas étendus sont tous rattachés à un schéma général de la méthode de Diels-Alder, en l'occurrence le schéma « classique » de la figure 6.2(a) : les réactions que ces schémas étendus représentent sont donc des sous-familles de la famille des réactions associée au schéma « classique ». Certains schémas étendus peuvent même être des spécialisations d'autres schémas étendus : le schéma de la figure 6.3(c) avec ses deux groupes aldéhyde peut être perçu comme une spécialisation du schéma de la figure 6.3(a) muni d'un seul groupe aldéhyde. L'ensemble des schémas CMS forme donc une hiérarchie organisée par la relation d'inclusion entre schémas et représentée schématiquement sur la figure 6.4, au sommet de laquelle se trouvent les schémas généraux et en dessous desquels se trouvent les schémas étendus. Si l'extraction automatique des schémas généraux de méthodes de synthèse à par-

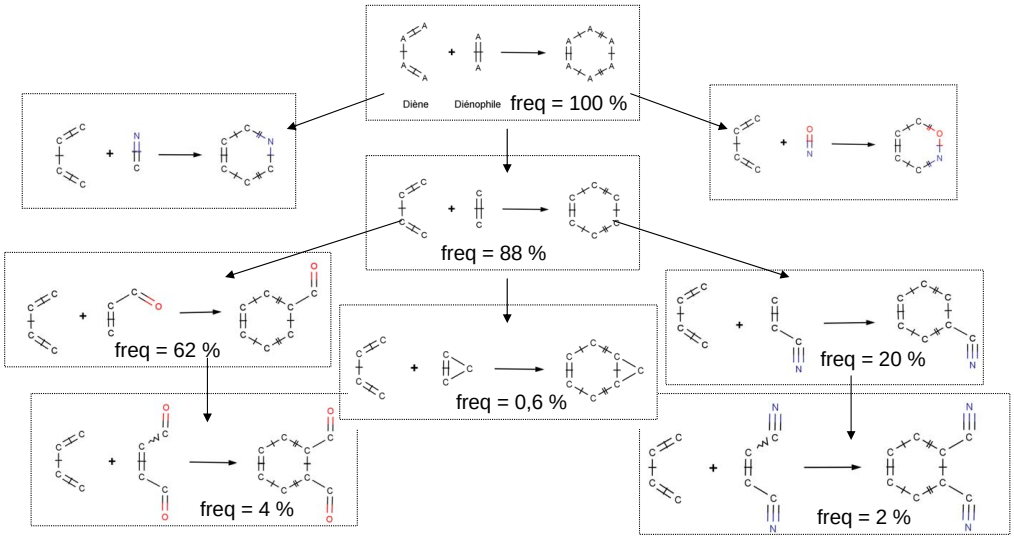


Fig. 6.4: Hiérarchie des schémas caractéristiques de la méthode de Diels-Alder

tir d'un ensemble de réactions ne présente pas un grand intérêt pour les chimistes, puisque ces derniers connaissent déjà ces schémas généraux, l'extraction automatique des hiérarchies des schémas étendus peut renseigner les chimistes sur la répartition des différentes formes que revêt une ou des méthodes de synthèse. Cette application est d'autant plus intéressante qu'il peut être difficile et fastidieux d'énumérer de façon exhaustive tous les schémas caractéristiques étendus contenus dans une grande BdR. La principale difficulté pour établir cette hiérarchie est de disposer des schémas caractéristiques étendus les plus spécifiques (i.e. les plus bas dans la hiérarchie). Les autres schémas composant le reste de la hiérarchie peuvent alors être calculés comme les schémas à l'intersection des schémas étendus (i.e. en calculant les sous-graphes communs maximaux aux graphes de réactions équivalents aux schémas étendus). Pour pouvoir extraire les schémas caractéristiques étendus de méthodes de synthèse, plusieurs obstacles doivent cependant être surmontés.Certains de ces obstacles sont étroitement liés au domaine d'application alors que d'autres sont de nature informatique.

Le principal problème relatif à la chimie est de savoir caractériser formellement et précisément un schéma de méthode de synthèse étendu, quite si nécessaire, à intégrer dans l'algorithme d'identification des informations complémentaires, comme les conditions réactionnelles. En particulier, cette caractérisation doit déterminer le niveau de détail que doivent présenter les schémas étendus. La solution à ce problème passe vraisemblablement par différentes pistes à suivre parallèlement, notamment par la définition d'une fonction de score, d'une méthode de sélection et d'une modélisation des données intégrant les connaissances du domaine. Si ces problèmes n'ont rien d'évident et mobilisent des connaissances pointues en synthèse organique, ces problèmes n'étant pas au cœur de ce mémoire ont été éludés au profit de l'analyse des aspects informatiques directement liés à la fouille de graphes. Àfin de pouvoir tester les méthodes développées (cf section 6.4), il est toutefois nécessaire de disposer d'une fonction de score. Une fonction similaire à la fonction d'information du chapitre 5 a été choisie à cette fin, de sorte que les schémas recherchés en pratique ne prétendent pas être rigoureusement les schémas CMS.

Ce chapitre s'intéresse donc essentiellement au problème informatique que pose de façon sous-jacente l'extraction des schémas CMS et qui consiste à rechercher des motifs de graphes de faible fréquence, typiquement inférieure à 0,1 %, dans de grands volumes de données. L'exemple de la méthode de Diels-Alder permet de mieux appréhender ce problème. Le tableau de la figure 6.5 précise les fréquences absolues et relatives des différents schémas caractéristiques de la méthode de Diels-Alder présentés sur la figure 6.3, dans les jeux de données A, B et C introduits au chapitre 4. La méthode de Diels-Alder est une des méthodes

Données	Jeu A (Orgsyn)			Jeu B (Extrait de Reflib)			Jeu C (Réactions de Diels-Alder)		
Taille	2608			7029			3248		
Schéma	fréq.	fréq. rel.	rapp. fréq.	fréq.	fréq. rel.	rapp. fréq.	fréq.	fréq. rel.	rapp. fréq.
6.2(a)	9	0,3 %	100 %	133	1,9 %	100 %	2855	88 %	100 %
6.3(a)	7	0,3 %	78 %	101	1,4 %	76 %	2009	62 %	70 %
6.3(b)	4	0,2 %	44 %	35	0,5 %	26 %	664	20 %	23 %
6.3(c)	0	0 %	0 %	5	0,07 %	4 %	139	4 %	4,8 %
6.3(d)	0	0 %	0 %	1	0,01 %	0,8 %	70	2 %	2,4 %
6.3(e)	1	0,04 %	11 %	0	0 %	0 %	20	0,6 %	0,7 %

Fig. 6.5: Fréquences des schémas de la méthode de Diels-Alder de la figure 6.3

les plus connues en synthèse organique et ses nombreuses « vertus » en font également une des plus employées. Pourtant le schéma « classique » de la méthode de Diels-Alder (cf figure 6.2(a)) n'est présent que dans 7 des 2608 réactions traitées de la base Orgsyn, soit une fréquence relative de 0,3 %, et dans 101 des 7029 réactions extraites de la base Reflib, soit une fréquence à peine plus importante de 1,9 %[45]. Même dans le jeu C de données qui contient exclusivement des réactions de Diels-Alder, le schéma n'a pas une fréquence de 100 % mais seulement de 88 %, les 22 % de réactions restantes ne répondant pas au schéma classique de la figure 6.2(a) mais à d'autres schémas généraux tels que ceux présentés sur la figure 6.2. Si on s'intéresse aux schémas caractéristiques étendus de la figure 6.3, les fréquences chutent encore davantage. Le schéma étendu le plus commun est celui de la figure 6.3(a), qui comporte un groupe aldéhyde. Pourtant sa fréquence relative n'est plus que de 0,3 % dans le jeu A et de

[45]Le fait que la fréquence soit plus élevée dans l'extrait de Reflib que dans Orgsyn peut s'expliquer par le fait que l'extrait de Reflib présente un contenu relativement plus homogène, en raison d'une méthode ciblée de sélection des réactions (rendement élevé, éléments chimiques présents ...) que celui de la base Orgsyn, qui au contraire illustre une grande diversité de méthodes de synthèse à l'aide de relativement peu de réactions.

1,4 % dans le jeu B. Si on rajoute un second groupe aldéhyde (cf figure 6.3(c)), le support chute respectivement à 0 et 5 (soit 0,07 %). Enfin le schéma de la figure 6.3(e) est absent du jeu B et présent dans une seule réaction du jeu A.

Par ailleurs, le rapport de la fréquence d'un schéma étendu divisée par la fréquence du schéma général de la figure 6.2(a), précisé dans la colonne *rapp. fréq.* du tableau de la figure 6.5, permet d'apprécier la distribution des différents schémas étendus. On observe en effet que ce rapport est comparable d'un jeu de données à un autre, quand toutefois son calcul s'appuie sur un nombre significatif d'exemples[46]. Par exemple, le rapport du schéma de la figure 6.3(a) présentant un groupe aldéhyde fluctue entre 70 % et 78 % et celui du schéma de la figure 6.3(c) présentant un groupe nitrile est proche de 25 %. Cette observation laisse à penser que la distribution des schémas étendus au sein de la méthode de synthèse à laquelle ces schémas sont rattachés, est relativement indépendante du jeu de données considéré. Si on admet cette hypothèse, la fréquence relative qu'aurait le schéma de la figure 6.3(e) dans une BdR diversifiée comme la base Orgsyn peut être extrapolée comme étant égale au produit du rapport des fréquences dans le jeu C pour le schéma étendu considéré, multiplié par la fréquence relative du schéma général dans le jeu A, soit $0,7\% \times 0,3\% = 0,002\%$! Pour qu'un tel schéma soit suffisamment représenté par 10 réactions, il faudrait que les données contiennent pas moins de $10/(2 \cdot 10^{-5}) = 500000$ réactions. Si la confection d'un tel jeu de données est possible, puisqu'il existe déjà des millions de réactions répertoriées dans les BdR, il est évident qu'aux vues des conclusions du chapitre 4, la recherche des schémas fréquents de fréquence supérieure à 0,002 % dans 500000 réactions est une tâche hors de portée des processeurs les plus performants. L'extraction des schémas les plus informatifs fréquents présentée au chapitre 5 ne semble donc pas envisageable pour extraire à grande échelle les schémas représentatifs de méthode de synthèse.

6.1.2 Recherche heuristique dans un espace d'état

En supposant que l'extraction des schémas CMS puisse être traitée à l'aide du modèle des motifs les plus informatifs et d'une modélisation chimique adéquate, les conclusions de la section 6.1.1 montrent qu'il est extrêmement difficile, pour ne pas dire impossible, d'extraire les motifs les plus informatifs à des fréquences aussi basses. En extrapolant les courbes de la figure 4.23, une telle extraction nécessiterait de filtrer des centaines de millions de motifs fréquents. Ce problème de la fouille de motifs de faible fréquence n'est pas nouveau. Ce problème a notamment été abordé à travers l'extraction des motifs d'exception (ou « outliers ») qui sont des motifs qui contredisent (ou du moins apportent une information complémentaire) aux règles ou motifs qui sont représentatifs d'une grande partie des données (et qui sont donc de fréquence élevée). Par exemple Liu *et al.* (1999) cherchent les motifs d'exception couvrant les quelques transactions qui par ailleurs, ne sont pas couvertes par les motifs fréquents. La recherche de motifs d'exception n'est toutefois pas la seule motivation pour chercher des motifs de faible fréquence : ainsi Szathmary *et al.* (2007) cherchent à décrire les « motifs rares » (i.e. non fréquents) par les motifs rares minimaux. Si la problématique de l'extraction des schémas CMS se rapproche davantage de cette dernière approche que de celle de la détection de outliers, elle s'en distingue toutefois dans la mesure où la caractérisation de **tous** les motifs de faible fréquence (i.e. inférieure à un seuil) ne présente pas ici d'intérêt. En effet seul un

[46] Les rapports de fréquence tendent toutefois à s'écarter lorsque le schéma considéré présente une fréquence proche de zéro dans le jeu A ou B. La raison de cet écart est un problème de mesure puisque, les supports étant nécessairement des entiers, les rapports de supports proches de zéro introduisent un bruit de quantification empêchant toute comparaison fiable.

sous-ensemble très restreint de motifs pertinents (i.e. maximisant une fonction de score) sont ici recherchés, à l'image des motifs les plus informatifs du chapitre précédent.

À ce propos, la principale limitation qui rend impossible l'extraction des motifs les plus informatifs de faible fréquence, est que les algorithmes d'extraction associés sont garantis exacts et complets : on est en droit de se demander si la contrainte de complétude du résultat ne pourrait pas être relâchée dans la mesure où cette relaxation permettrait en retour de diminuer substantiellement les temps de calcul et donc d'accéder à des motifs de fréquence plus faible. En particulier l'extraction systématique des motifs les plus informatifs pourrait être remplacée par une recherche heuristique dans un espace d'état telle qu'elle a été introduite en intelligence artificielle (voir par exemple le chapitre 3 de Russell et Norvig (2002)). Dans le cas présent, l'espace d'état n'est rien d'autre que le diagramme de l'ordre des motifs : les états sont les motifs et les transitions entre états correspondent aux arcs du diagramme, c'est-à-dire aux relations qui unissent un motif à ses successeurs et prédécesseurs immédiats. L'algorithme fouillerait cet espace en faisant croître un motif courant à partir du motif vide, en tant qu'état de départ de la recherche. La construction de ce motif serait guidée par la plus forte croissance de la fonction de score utilisée, et ce jusqu'à ce que qu'aucun successeur immédiat du motif courant ne conduise à un plus grand score. Un algorithme de recherche par faisceaux[47] permettrait de scinder la recherche en k trajectoires indépendantes afin de retenir non pas un mais k meilleurs motifs candidats trouvés. Un tel motif qui détient un score supérieur à tous ceux de ses successeurs immédiats n'est toutefois pas un motif des plus informatifs. Pour qu'il en soit ainsi, il faudrait encore vérifier que son score n'est inférieur à celui d'aucun prédécesseur immédiat. Les méthodes heuristiques mentionnées permettent de produire facilement des motifs non dominés par aucun successeur immédiat, à partir desquels peuvent être éventuellement extraits des motifs des plus informatifs grâce à un algorithme de filtrage qui compare le score du motif candidat à celui de ses prédécesseurs immédiats.

Outre le fait de pouvoir espérer une convergence plus rapide vers quelques motifs non dominés par aucun successeur immédiat, cette approche présente l'autre avantage de s'affranchir de tout seuil f_{min} de fréquence minimale. En effet l'algorithme n'ayant pas à être complet, ne doit plus énumérer tous les motifs fréquents. Il peut au contraire converger vers quelques motifs de faible fréquence et exhiber ainsi des motifs des plus informatifs qui sont par ailleurs inaccessibles aux algorithmes complets du chapitre 5. Ce faisant, une telle méthode deviendrait dans son principe, similaire à celui de l'algorithme `Subdue` (Cook et Holder, 1994). Toutefois, l'utilisation de l'algorithme `Subdue` pose essentiellement deux problèmes.

D'abord l'algorithme `Subdue` ne tient pas compte du problème que pose les cas d'isomorphisme entre motifs de graphes. Les auteurs de `Subdue` reconnaissent qu'il s'agit là d'une véritable limitation (cf la fin de la section 7.4.1 et la conclusion du chapitre 7 dans Cook et Holder (2006)). En effet la non-prise en compte de l'isomorphisme fait qu'un motif peut être généré plusieurs fois à une permutation des sommets près, comme illustré par l'exemple de la figure 6.6. Le motif (a) peut ainsi être généré de deux façons différentes selon l'ordre dans lequel les atomes de carbone et d'hydrogène sont insérés. Pire, la duplication de motifs n'étant pas détectée, se propage aux successeurs immédiats de ces motifs, ce qui provoque une génération en chaîne explosive de motifs isomorphes suivie de l'écroulement des performances. Ainsi sur la figure 6.6, la duplication du motif (a) double le nombre de fois où le motif (b) est généré à partir d'extensions successives du motif (a). Finalement sur cet exemple pourtant très simple, le motif (b) peut être généré pas moins de 25 fois selon des chemins de génération différents. Ce phénomène se produit d'autant plus souvent que le guidage de

[47] *Beam search* en anglais. Voir chapitre 3 de Russell et Norvig (2002).

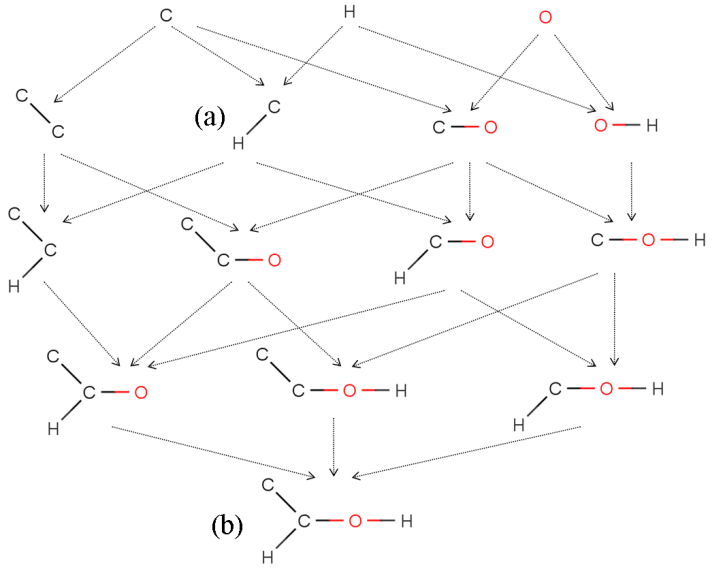


FIG. 6.6: Exemple de génération multiple de motifs

la recherche par une fonction de score augmente les chances pour que les différents faisceaux de recherche convergent vers un même motif selon des chemins différents. À titre d'exemple, lorsque `Subdue`[48] est appliqué dans sa configuration la plus rapide (avec l'option -prune) au plus petit jeu de données introduit au chapitre 4 (i.e. le jeu A extrait de la base Orgsyn, cf p. 95), un seul motif est renvoyé au bout de 40 minutes[49], qui n'est autre que le motif trivial constitué d'un atome de carbone relié à deux atomes d'hydrogène (cf figure 6.7). Sans élagage (i.e. sans l'option -prune), `Subdue` met 58 minutes pour trouver 2 motifs supplémentaires qui sont isomorphes et représentent un atome de carbone lié à un atome d'hydrogène... Cet écroulement de performances pourrait toutefois être évité en partie si l'algorithme `Subdue` mémorisait les motifs déjà fouillés à l'aide d'un dictionnaire de motifs identique à celui utilisé par l'algorithme n° 2 du chapitre 5 (cf p. 116).

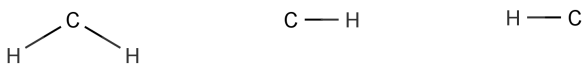


FIG. 6.7: Les trois motifs produits par `Subdue` sur le jeu A (ORGSYN)

Le second problème n'est pas spécifique à `Subdue`, comme l'est le problème précédent, mais est général à tous les algorithmes de recherche heuristique similaires dans leur principe à `Subdue`. En effet, puisque l'objectif est de libérer la recherche de toute contrainte vis-à-vis d'un seuil de fréquence minimale, l'espace d'état explorable en théorie est l'ensemble des motifs de graphes présents dans les données (i.e. de fréquence non nulle). Les données étant formées de réactions variées, cet espace est énorme[50]. Une recherche exhaustive étant

[48] `Subdue` peut être téléchargé à l'adresse http ://ailab.wsu.edu/subdue

[49] Sur un Intel Core 2, 1,8 GHz

[50] Il est difficile de quantifier exactement le nombre de motifs contenus dans un ensemble de quelques milliers d'équations de réactions mais ce nombre est selon toute vraisemblance supérieur à des dizaines de milliards.

impossible, il est nécessaire de restreindre la recherche à un sous-ensemble plus réduit de motifs, sans recourir pour autant à un seuil sur la fréquence. Une des façons de résoudre ce problème est de guider l'algorithme de recherche, par exemple en suivant les chemins de plus forte croissance de la fonction de score considérée. Ce faisant, les faisceaux de recherche auront tendance à se focaliser sur les sous-espaces de score élevé et ainsi à se concentrer sur des « zones » de l'espace de recherche où les motifs présentent un score élevé proche du maximum global. Or l'analyse des schémas les plus informatifs atteste que les schémas les plus représentatifs des données (i.e. de score le plus élevé) ne sont pas les plus caractéristiques de méthodes de synthèse : ainsi les trois premiers schémas les plus informatifs de la figure 5.23 p. 139, dont les scores sont les plus élevés (cf tableau 5.21 p. 137), sont clairement trop généraux pour être qualifiés de schémas CMS. Au contraire, les schémas les plus informatifs qui s'apparentent le plus à des schémas CMS ont des scores variables plutôt faibles (et donc des rangs élevés dans la liste des MPIs), comme par exemple les MPI n° 10, n° 72 ou n° 75 représentés sur la figure 5.24 p. 140. Les algorithmes de recherche guidés par la croissance de la fonction de score ont donc toutes les chances de manquer la plupart des motifs les plus intéressants, et de ne produire que des motifs de plus grande score, souvent par ailleurs triviaux. Les motifs de la figure 6.7 obtenus par `Subdue` en sont un parfait exemple.

La suite de ce chapitre propose une méthode alternative pour réduire l'espace de recherche tout en permettant l'accès à des motifs de faible score et de faible fréquence. Contrairement à l'approche de type « `Subdue` », cette approche consiste à ne pas guider la recherche uniquement selon la croissance de la fonction de score mais à intégrer une contrainte supplémentaire sur la structure des motifs recherchés.

6.1.3 Recherche heuristique dans un espace d'état contraint par un exemple : une approche « transductive »

L'introduction de cette nouvelle contrainte passe par une redéfinition du problème. Le problème initial tel qu'il a été considéré jusqu'à présent consiste à extraire directement les schémas CMS à partir d'un ensemble de réactions, comme indiqué sur la figure 6.8. La nouvelle

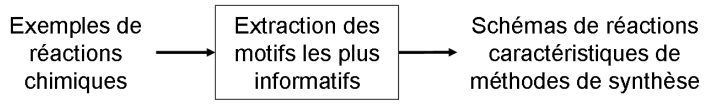


FIG. 6.8: L'extraction des schémas CMS

définition du problème, représentée sur la figure 6.9, repose sur la donnée supplémentaire d'une réaction en entrée. Le problème consiste à extraire le schéma caractéristique de la méthode de synthèse sous-jacente à la réaction fournie en entrée en comparant les schémas inclus dans cette réaction avec ceux présents dans un grand nombre d'exemples de réactions. La recherche est réitérée pour toute nouvelle réaction disponible en entrée du processus. La contrainte structurelle évoquée précédemment vient du fait que l'espace de recherche se limite à l'ensemble des schémas de réactions inclus dans la réaction fournie en entrée et non plus à l'ensemble beaucoup plus grand de tous les schémas de réactions inclus dans les données. Outre la réduction de l'espace de recherche, le problème de *l'extraction du schéma CMS sous-jacent à une réaction* présente cet autre avantage d'associer à la réaction en entrée son schéma caractéristique. Cette association peut servir à la classification conceptuelle des réactions (les schémas caractéristiques jouant le rôle d'intension du concept, cf page 55).

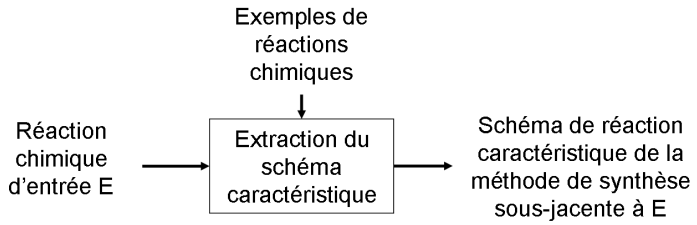


FIG. 6.9: L'extraction du schéma CMS sous-jacent à une réaction

D'un point de vue purement formel, il est possible de faire un parallèle entre cette approche de la classification et l'*apprentissage transductif* tel que l'a introduit Vladimir Vapnik (cf le concept de « transductive inference » au chapitre 8 de Vapnik (1998))[51]. Ce dernier

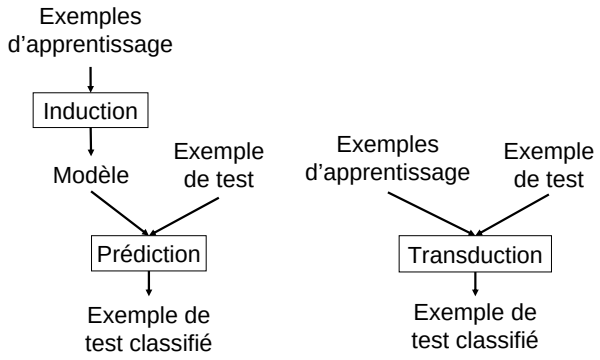


FIG. 6.10: Induction. Prédiction. Transduction

concept est illustré par la figure 6.10. Ainsi, l'apprentissage « classique » à partir d'exemples consiste, si on prend l'exemple d'un problème de classification supervisée, à induire un modèle (par exemple sous forme de fonctions ou de règles de classification) à partir des exemples d'apprentissage dont on connaît la classe d'appartenance, puis à déduire de ce modèle la classe la plus vraisemblable pour chaque exemple de test (par exemple en faisant voter les règles de classification qui couvrent l'exemple). L'avantage d'un tel procédé en deux temps est que la phase de prédiction repose sur un calcul généralement beaucoup plus rapide que celui de l'apprentissage. L'inconvénient est que le modèle induit présente une information inférieure, et en pratique généralement beaucoup plus pauvre, que celle contenue dans les exemples d'apprentissage, au sens où il est impossible de reconstruire l'ensemble des exemples à partir du modèle. Par exemple, dans le cas des techniques de classification fondées sur les motifs (cf la section 2.4.3 dans le cas des motifs de graphes), le modèle est constitué d'un ensemble de motifs associés à leurs fréquences dans les jeux d'exemples positifs et négatifs. Plus le nombre de motifs y est important, plus la prédiction pourra être précise. Il est évident qu'il n'est pas possible de mémoriser les fréquences de tous les motifs et il est alors nécessaire de

[51] Merci à P. Jauffret de m'avoir suggéré cette comparaison.

réduire le nombre de motifs considérés, par exemple grâce à un filtrage selon la fréquence. Mais il est tout aussi évident que la pertinence du choix des motifs dépend des exemples de test que la méthode va devoir classifier mais qu'elle n'est pas en mesure de connaître. Dans le cas de l'apprentissage transductif, ce problème du choix du modèle ne se pose plus : la classification de l'exemple de test se fait à partir d'un modèle établi spécifiquement pour l'exemple de test considéré et ce à partir de l'ensemble des exemples d'apprentissage. Chaque classification d'un exemple de test se fonde donc sur l'intégralité de l'information contenue dans les exemples d'apprentissage. Le seul inconvénient de l'apprentissage transductif est de nécessiter pour classifier chaque exemple de test, un traitement plus complexe et donc plus long que celui utilisé lors de la phase de prédiction.

Ainsi, si on prolonge cette analogie dans le cas de l'extraction des schémas CMS, l'extraction des motifs les plus informatifs présentée au chapitre 5 correspond à l'étape d'induction, qui produit à partir des données un modèle constitué d'un ensemble de schémas représentatifs de grandes familles de réactions. La phase de prédiction consiste étant donné une réaction, à déterminer les schémas les plus informatifs contenus dans l'équation de la réaction et en fonction du résultat, à classer la réaction dans une des familles de réactions associées aux schémas les plus informatifs. Cette approche, on l'a vu, est limitée par le fait que les schémas CMS ont peu de chance de figurer parmi les motifs les plus informatifs (du fait du filtrage des motifs non fréquents). L'approche « transductive » de la figure 6.9 permet, comme on va le voir, de renvoyer le schéma caractéristique de la réaction fournie en entrée, en fouillant les exemples d'apprentissage partageant ce même schéma caractéristique, même si leur nombre est très faible.

La suite du chapitre présente cette approche de recherche heuristique contrainte par un exemple. La section 6.2 définit formellement le problème de l'extraction du schéma CMS d'une réaction et montre que ce problème applicatif propre à la chimie revient d'un point de vue informatique, à rechercher un sous-graphe de score maximal dans un intervalle de graphes. La section 6.3 traite ce problème en proposant deux solutions : la première est fondée sur les méthodes existantes de recherche de sous-graphes fréquents. La seconde est un algorithme original de recherche heuristique baptisé `CrackReac`. Enfin la section 6.4 présente les tests qui ont été réalisés pour évaluer les performances et la qualité des résultats produits dans le cadre de l'extraction des schémas CMS sous-jacents à des réactions.

6.2 Le problème de l'extraction du schéma CMS sous-jacent à une réaction

Intuitivement, le *problème de l'extraction du schéma CMS sous-jacent à une réaction* consiste, étant donnée une réaction d'équation E, à déterminer le schéma réactionnel $S_{car}(E)$ contenu dans E qui exprime non seulement **l'effet** que produit la méthode de synthèse sous-jacente de la réaction sur les graphes moléculaires des réactants, mais aussi et dans la mesure du possible, les éléments structuraux présents dans les réactants qui contribuent à l'« applicabilité » de la méthode, c'est-à-dire qui sont pour partie à **l'origine** de la transformation. Considérons l'exemple de la réaction dont l'équation est représentée sur la figure 6.11. Les chimistes reconnaîtront dans cette équation la méthode de synthèse dite de *couplage de Sonogashira* dont le schéma général est donné sur la figure 6.12 et où l'atome X représente un atome halogène quelconque [52]. Ce schéma est bien inclus dans l'équation de la figure 6.11, au

[52] Les éléments halogènes sont le fluor F, le chlore Cl, le brome Br ou l'iode I.

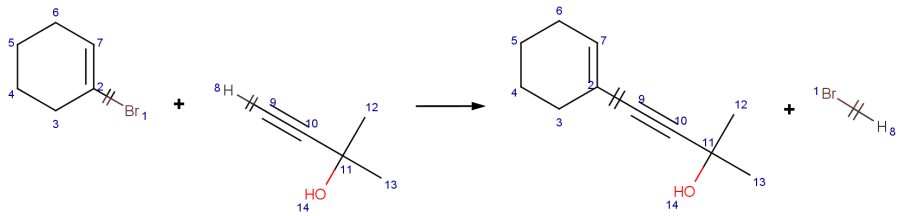


FIG. 6.11: Un exemple de réaction

FIG. 6.12: Schéma général de la méthode de synthèse de couplage de Sonogashira

sens de l'inclusion des schémas de réactions définie à la section 4.2. Toutefois le schéma caractéristique de la réaction de la figure 6.11 ne se résume pas au schéma général de la méthode de la figure 6.12. D'autres éléments, comme des groupes fonctionnels, peuvent indirectement prendre part au « mécanisme » de la réaction. Ainsi le schéma CMS sous-jacent à la réaction de la figure 6.11 est vraisemblablement celui de la figure 6.13 où la liaison double du second réactant est connue pour favoriser la réaction, en plus du groupe alcool du premier réactant.

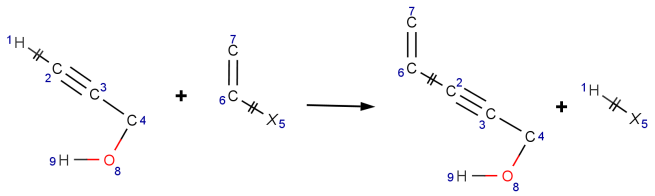


FIG. 6.13: Schéma caractéristique de la réaction de la figure 6.11

Les « schémas CMS sous-jacents aux réactions » correspondent aux schémas CMS « étendus » introduits à la section 6.1.1, qui comportent tous les éléments structurels importants favorisant la réaction. La section 6.1.1 a déjà mis en évidence la difficulté de définir formellement un schéma CMS étendu et il en va de même pour la définition formelle des schémas CMS sous-jacents aux réactions. La résolution de ce problème de modélisation chimique étant loin d'être évidente, on émet ici l'hypothèse forte qu'il existe une fonction de score $s_E : S \mapsto s_E(S)$ capable d'apprécier la qualité du schéma S en tant que schéma caractéristique de l'équation chimique E. Conformément au schéma de la figure 6.9, cette fonction de score peut s'appuyer sur des données $\mathcal{D}$, constituées d'un ensemble d'équations de réactions.

Par ailleurs, le schéma CMS sous-jacent à une réaction exprime nécessairement l'action que produit la dite réaction sur les graphes moléculaires des réactants. Ce schéma contient donc

au moins l'ensemble des liaisons brisées, modifiées et formées par la réaction. Cet ensemble de liaisons et les atomes incidents forment ce que les chimistes appellent parfois le *cœur de la réaction*[53].

Définition 6.2.1. Étant donnée une équation appariée $E = (\mathcal{R}, \mathcal{P}, \lambda_{\mathcal{R}}, \lambda_{\mathcal{P}})$ d'une réaction, le *schéma du cœur de la réaction* est le schéma $S_{coeur}(E)$ obtenu à partir de l'équation E en ne conservant dans $\mathcal{R}$ et $\mathcal{P}$ que les liaisons instables et les atomes qui sont incidents à ces liaisons. Formellement en notant $\nu = \lambda_{\mathcal{P}}^{-1} \circ \lambda_{\mathcal{R}}$ l'appariement des sommets de $\mathcal{R}$ vers ceux de $\mathcal{P}$ (cf section 4.5), une liaison est *instable* si la paire $\{v_1, v_2\}$ de sommets associée dans $\mathcal{R}$ vérifie une des trois conditions suivantes :

$\{v_1, v_2\} \in E(\mathcal{R})$ et $\{\nu(v_1), \nu(v_2)\} \notin E(\mathcal{P})$ (Destruction d'une liaison)
$\{v_1, v_2\} \notin E(\mathcal{R})$ et $\{\nu(v_1), \nu(v_2)\} \in E(\mathcal{P})$ (Formation d'une liaison)
$\{v_1, v_2\} \in E(\mathcal{R})$ et $\{\nu(v_1), \nu(v_2)\} \in E(\mathcal{P})$ et $l^e_{\mathcal{P}}(\{\nu(v_1), \nu(v_2)\}) \neq l^e_{\mathcal{R}}(\{v_1, v_2\})$ (Modification d'une liaison)

Le centre de la réaction de la figure 6.11 est représenté sur la figure 6.14(a). Le schéma

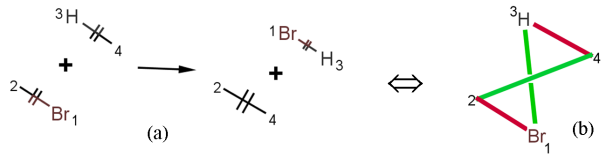


FIG. 6.14: Cœur (a) et graphe de réaction équivalent (b) de la réaction de la figure 6.11

CMS sous-jacent à une réaction d'équation E contient donc nécessairement le cœur $S_{coeur}(E)$ tout en étant contenu dans E. Le schéma caractéristique est donc contraint d'appartenir à un intervalle dont les bornes inférieures et supérieures sont respectivement $S_{coeur}(E)$ et E :

Propriété 6.2.2. $S_{coeur}(E) \subseteq_S S_{car}(E) \subseteq_S E$

Cette contrainte aboutit à la définition formelle suivante :

Définition 6.2.3. Le problème de *l'extraction du schéma CMS sous-jacent à une réaction* d'équation E relativement à une fonction de score $s_E : S \mapsto s_E(S)$ associant à un schéma de réaction S, le score $s_E(S)$ dans un ensemble totalement ordonné (par exemple l'ensemble des nombres réels), est la détermination du ou des schémas réactionnels $S_{opt}(E)$ de plus grande score, parmi tous les schémas inclus dans l'intervalle $[S_{coeur}(E), E]$ (relativement à la relation $\subseteq_S$ d'inclusion des schémas de réaction).

La condition exigeant que l'ensemble des scores soit totalement ordonné est nécessaire pour pouvoir définir un schéma de plus grand score. Ce problème est reformulé de manière abstraite dans la section suivante, avant d'être abordé selon deux méthodes distinctes.

6.3 La recherche du motif optimal dans un intervalle de graphes

6.3.1 Définition du problème

Compte tenu de la propriété 4.5.6 d'isomorphisme entre l'ordre des schémas de réactions et l'ordre des graphes de réactions et de la propriété 4.5.5 de connexité des graphes de réaction,

[53] Le terme anglais *reaction center* est plus commun.

la propriété 6.2.2 peut de façon équivalente, s'exprimer en terme de graphes de réactions plutôt qu'en terme de schémas de réactions :

Proposition 6.3.1. *$\mathcal{G}(S_{coeur}(E)) \subseteq_G \mathcal{G}(S_{car}(E)) \subseteq_G \mathcal{G}(E)$ et $\mathcal{G}(S_{car}(E))$ est connexe*

La borne supérieure de l'intervalle est le graphe de réaction $\mathcal{G}(E)$ de l'équation E. La borne inférieure est le *graphe du cœur de la réaction*, noté $G_{coeur}(E) = \mathcal{G}(S_{coeur}(E))$, c'est-à-dire le graphe condensé de réaction du cœur de E. Le graphe de cœur de la réaction de la figure 6.11 est représenté sur la figure 6.14(b). Comme dans la recherche des schémas réactionnels fréquents au chapitre 4, le fait d'exprimer la contrainte d'intervalle en terme de graphes de réactions permet de manipuler des graphes connexes ordonnés selon la relation $\subseteq_G$ de sous-graphe isomorphe plus simple que la relation $\subseteq_S$ d'inclusion de schémas de réactions. L'intervalle de graphes associé à l'exemple de la figure 6.11 est représenté sur la figure 6.15. Le problème formel de l'extraction du schéma CMS d'une réaction (cf définition 6.2.3) peut

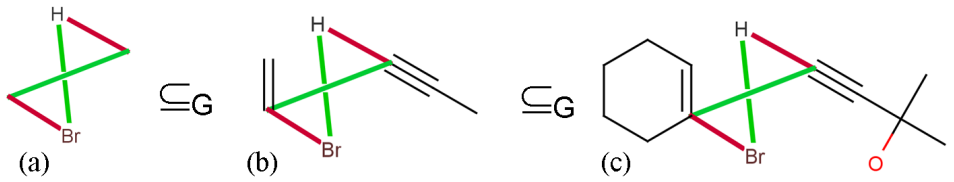


FIG. 6.15: La contrainte d'intervalle : $G_{coeur}(E)$ (a), $\mathcal{G}(S_{car}(E))$ (b) et $\mathcal{G}(E)$ (c)

donc, à l'aide des graphes de réactions, se réduire à un problème de fouille de graphes :

Définition 6.3.2. Étant données une fonction de score $s : g \mapsto s(g)$ associant à un graphe g le score $s(g)$ dans un ensemble totalement ordonné et une liste $((g_{inf_i}, g_{sup_i}))$ de paires de graphes telle que pour tout indice i, g_{inf_i} soit isomorphe à un sous-graphe partiel de g_{sup_i}, la *recherche des motifs optimaux dans les intervalles de graphes* $[g_{inf_i}, g_{sup_i}]$ consiste à trouver pour chaque indice i, le graphe (ou motif) optimal g_{opt_i} qui obtient le plus grand score $s(g)$, tout en étant connexe et en appartenant à l'intervalle de graphes $[g_{inf_i}, g_{sup_i}]$ (au sens de la relation $\subseteq_G$ de sous-graphe isomorphe).

Le fait de traiter non pas un seul mais plusieurs intervalles n'est pas rigoureusement nécessaire mais permet de comparer plus objectivement les performances des deux solutions proposées en section 6.3.2 et 6.3.3 (la solution de la section 6.3.2 n'étant pertinente que pour un nombre important d'intervalles).

Ce problème général est développé dans ce qui suit indépendamment de l'extraction des schémas caractéristiques, la fonction de score renfermant toute la spécificité de l'application traitée. Dans le cas d'un problème d'extraction de connaissances à partir de données, le score du motif g dépend entre autres facteurs de sa fréquence dans des données $\mathcal{D}$. Les fonctions de score informatives (cf définition 5.3.3) sont particulièrement adaptées pour extraire des motifs représentatifs des données. Dans le cas de l'extraction des schémas CMS sous-jacents aux réactions, la mise au point d'une fonction de score adaptée est un problème plus délicat. Cette difficulté a été passée outre afin de pouvoir tester malgré tout la méthode de recherche heuristique proposée. En lieu d'une fonction spécifique à la chimie, la fonction de score utilisée est une variante de la fonction s_i d'information (cf section 5.3.4). L'intuition se cachant derrière ce choix est que les schémas CMS des réactions (i.e. les schémas étendus des méthodes

de synthèse) sont les plus représentatifs des données dans la limite de leur intervalle. Cette fonction se définit ainsi :

Définition 6.3.3. La fonction s'_i d'un graphe de réaction g relativement à des données $\mathcal{D}$ et un intervalle $[g_{inf}, g_{sup}]$ est

$$s'_i(g, \mathcal{D}) \mapsto (I(g) - I(g_{inf})) \cdot \frac{\text{freq}_r(g, \mathcal{D})}{\text{freq}_r(g_{inf}, \mathcal{D})} \text{ avec}$$

$$I(g) = \sum_{v \in V(g)} i(l_v(v)) + \sum_{e \in E(g)} i(l_e(e)) \text{ où } i(l) = -\log_2 \left(\frac{n(l)}{\sum\limits_{l' \in \mathcal{L}} n(l')} \right)$$

où $n(l)$ désigne le nombre de sommets ou d'arêtes de $\mathcal{D}$ ayant pour étiquette l.

Comparée à la fonction d'information définie à la section 5.3.4 et utilisée au chapitre 5, la fonction s'_i remplace le facteur $I(g)$ de la fonction s_i par le facteur $I(g) - I(g_{inf})$, c'est-à-dire par le gain d'information $I(g|g_{inf} \subseteq_G g)$ sachant que g_{inf} est déjà connu pour être isomorphe à un sous-graphe de g. De même, la fonction s'_i remplace le facteur de la fréquence relative $\text{freq}_r(g, \mathcal{D})$ du motif g dans les données $\mathcal{D}$ par le rapport $\frac{\text{freq}_r(g,\mathcal{D})}{\text{freq}_r(g_{inf},\mathcal{D})}$, c'est-à-dire par la fréquence relative du motif g au sein de l'ensemble restreint des graphes de $\mathcal{D}$ ayant au moins un sous-graphe isomorphe à g_{inf}. Le dénominateur $\text{freq}_r(g_{inf}, \mathcal{D})$ est toutefois constant lors de la recherche du motif optimal et n'influe donc pas sur le classement des motifs triés selon leur score et donc sur la valeur de g_{opt}.

Deux solutions sont envisageables pour résoudre ce problème : l'une consiste à filtrer les motifs de graphes fréquents ; l'autre à rechercher directement le motif g_{opt} dans les données, à l'aide d'un algorithme de recherche heuristique baptisé `CrackReac`. Ces deux approches sont décrites dans les deux sections suivantes.

6.3.2 Une première solution fondée sur le filtrage des motifs fréquents

La première méthode consiste à filtrer les motifs fréquents afin d'extraire pour chaque intervalle considéré, le motif qui optimise une fonction de score s donnée. Formellement on considère une liste $(I_i) = ((g_{inf_i}, g_{sup_i}))$ d'intervalles de graphes spécifiés par leurs bornes inférieures g_{inf_i} et supérieures g_{sup_i} et une fonction de score s dont le score $s(g)$ dépend de la fréquence du motif g dans des données $\mathcal{D}$. Le problème consiste alors à déterminer le motif g_{opt_i} de score maximal inclus dans chaque intervalle I_i.

La solution la plus évidente pour répondre à ce problème consiste à appliquer la méthode résumée sur la figure 6.16. Cette méthode a l'avantage d'être exacte (elle garantit de trouver le ou les motifs optimaux) mais est en pratique inutilisable : le nombre de sous-graphes contenus dans un graphe de réaction de taille moyenne est en effet très grand, de l'ordre de plusieurs dizaines de millions, de sorte que le temps nécessaire pour les extraire lors de l'étape n° 1, multiplié par ailleurs par le nombre d'intervalles à traiter, devient exorbitant. À titre d'exemple, le graphe de réaction de la figure 6.15, qui est particulièrement petit par rapport à la taille moyenne des graphes traités, compte déjà plus de 685000 sous-graphes partiels non isomorphes deux-à-deux.

Une solution plus réaliste consiste à introduire un seuil de fréquence minimal f_{min} puis à appliquer la méthode de la figure 6.17. Les différentes étapes et données intermédiaires produites par la méthode sont représentées schématiquement sur la figure 6.18. L'étape n° 2

1. **Filtrage selon la borne supérieure.** Pour chaque intervalle I_i, produire l'ensemble $\mathcal{F}_i$ des sous-graphes partiels connexes contenus dans le graphe g_{sup_i} et définis à un isomorphisme près, en recherchant les motifs fréquents, de fréquence 1 dans les données constituées du singleton $\{g_{sup_i}\}$.
2. **Filtrage selon la borne inférieure.** Extraire, à l'aide d'un algorithme de détection de sous-graphe isomorphe, le sous-ensemble $\mathcal{F}'_i$ des motifs de $\mathcal{F}_i$ qui contiennent au moins un sous-graphe partiel isomorphe à g_{inf_i}.
3. **Calcul du score et sélection du maximum.** Pour chaque motif g de $\mathcal{F}'_i$, calculer son score $s(g)$ à partir de sa fréquence calculée à l'étape n° 1 et extraire le motif g_{opt_i} dont le score est maximal.

FIG. 6.16: Méthode inefficace fondée sur la recherche de sous-graphes fréquents

permet pde ne considérer que les données contenant la borne inférieure associée à un intervalle donné. Ce préfiltrage permet d'éliminer une grande partie des données à fouiller et ainsi de garder le seuil f_{min} relativement élevé pour que la recherche des sous-graphes fréquents à l'étape n° 3 puisse se faire en un temps acceptable. Ce préfiltrage est par ailleurs « mutualisé » pour l'ensemble des intervalles en entrée présentant la même borne inférieure, grâce à l'étape n° 1. La contre-partie est l'introduction du seuil f_{min} qui n'offre plus la garantie de trouver le motif optimal si celui-ci devait avoir une fréquence inférieure à f_{min}. En pratique les tests montrent que l'essentiel du temps de calcul n'est pas consacré à l'étape n° 3 de fouille de données mais à l'étape n° 6 (et dans une moindre mesure les étapes n° 4 et n° 5) qui nécessite de détecter pour chaque intervalle lesquels des nombreux motifs fréquents de $\mathcal{F}'_j$ sont isomorphes à un sous-graphe de g_{sup_i}. Cette observation est une des raisons qui ont motivé le développement d'un algorithme de recherche heuristique présenté dans la section suivante.

6.3.3 L'algorithme `CrackReac` de recherche heuristique dans un intervalle de graphes

Comme cela a été expliqué dans l'introduction, l'idée de la recherche heuristique consiste ici à rechercher dans l'ordre des motifs de graphes, vu comme un espace d'état, le motif de plus grand score. L'originalité de `CrackReac` comparativement à l'algorithme `Subdue`, est d'introduire une contrainte d'intervalle. Cette contrainte permet non seulement de réduire la taille de l'espace de recherche mais permet également de simplifier et donc, d'accélérer certaines primitives de calcul liées à la génération des motifs, comme expliqué ci-après.

La raison de cette simplification vient du fait que l'algorithme `CrackReac` ne cherche plus à générer des motifs de graphes non isomorphes deux à deux mais directement des sous-graphes partiels de la borne supérieure g_{sup} de l'intervalle $[g_{inf}, g_{sup}]$ considéré. Plus précisément, si $(g'_{inf_i})_{1 \leq i \leq n}$ est la liste des sous-graphes partiels de g_{sup} isomorphes à g_{inf}, la contrainte $g_{inf} \subseteq_G g \subseteq_G g_{sup}$ peut avantageusement être remplacée par une disjonction de n contraintes $g'_{inf_i} \subseteq g' \subseteq g_{sup}$ dans laquelle g' est isomorphe à g et $\subseteq$ n'est plus la relation $\subseteq_G$ de sous-graphe isomorphe mais la relation beaucoup plus simple de sous-graphe partiel (voir section 2.3.1). Cette réécriture tient lieu d'équivalence. En effet, chaque motif

1. **Partition des données.** Réaliser une partition de l'ensemble des intervalles $\mathcal{I} = \{I_i\}$ en des sous-ensembles $\mathcal{I}_j = \{I_{ij}\} = \{(g_{inf_j}, g_{sup_{ij}})\}$ tels que tous les intervalles I_{ij} de $\mathcal{I}_j$ partagent la même borne inférieure g_{inf_j} (à un isomorphisme près).
2. **Préfiltrage des données.** Pour chaque partie $\mathcal{I}_j$, extraire des données $\mathcal{D}$ le sous-ensemble $\mathcal{D}_j$ des graphes qui contiennent au moins un sous-graphe isomorphe à g_{inf_j}.
3. **Fouille des motifs fréquents.** Rechercher l'ensemble $\mathcal{F}_j$ des graphes fréquents de $\mathcal{D}_j$ dont la fréquence relative est supérieure ou égale à f_{min}.
4. **Post-filtrage selon la borne inférieure.** Extraire le sous-ensemble $\mathcal{F}'_j$ de $\mathcal{F}_j$ des graphes qui contiennent au moins un sous-graphe isomorphe à g_{inf_j}.
5. **Calcul des scores.** Calculer pour chaque motif $g \in \mathcal{F}'_j$ son score $s(g)$ à partir de sa fréquence calculée à l'étape n° 3.
6. **Post-filtrage selon la borne supérieure.** Pour chaque intervalle I_{ij} de $\mathcal{I}_j$ de borne supérieure $g_{sup_{ij}}$, extraire de $\mathcal{F}'_j$ le sous-ensemble $\mathcal{F}''_{ij}$ des graphes qui sont isomorphes à un sous-graphe de $g_{sup_{ij}}$.
7. **Sélection des maxima de scores.** Pour chaque intervalle I_{ij} de $\mathcal{I}_j$, retourner le ou les motifs $g_{opt_{ij}}$ de $\mathcal{F}''_{ij}$ dont le score est maximal.

FIG. 6.17: Méthode fondée sur la recherche de sous-graphes fréquents

g appartenant à l'intervalle $[g_{inf}, g_{sup}]$ selon $\subseteq_G$ est par définition, isomorphe à au moins un sous-graphe g' partiel de g_{sup}. De plus, un des sous-graphes partiels de g' isomorphe à g_{inf} est par transitivité de la relation $\subseteq$, un sous-graphe partiel de g_{sup} et est donc un des sous-graphes g'_{inf_i}. On a donc $g_{inf} \subseteq_G g \subseteq_G g_{sup} \Rightarrow \exists i, \exists g', g' \simeq_G g$ et $g'_{inf_i} \subseteq g' \subseteq g_{sup}$. La réciproque $\exists i, g'_{inf_i} \subseteq g' \subseteq g_{sup} \Rightarrow \exists g, g_{inf} \subseteq_G g \subseteq_G g_{sup}$ est immédiate.

Un inconvénient de `CrackReac` est de devoir traiter successivement et indépendamment les différents intervalles $[g'_{inf_i}, g_{sup}]$ contenus dans g_{sup}, ce qui peut avoir pour conséquence de traiter des sous-graphes identiques d'un intervalle à un autre. Toutefois dans le cas de la recherche de schémas caractéristiques, la borne inférieure correspond au cœur de la réaction fournie en entrée. Comme un graphe de réaction ne possède qu'un sous-graphe de cœur connexe, `CrackReac` traite en pratique un seul intervalle $[g'_{inf_i}, g_{sup}]$ par intervalle $[g_{inf}, g_{sup}]$.

Pour un indice i donné, l'espace de recherche associé est donc l'ensemble des sous-graphes connexes de g_{sup} qui contiennent le sous-graphe connexe g'_{inf_i}. La génération de ces sous-graphes est un problème beaucoup plus facile que la génération de graphes non isomorphes deux-à-deux. En effet, un sous-graphe connexe est défini par l'ensemble de ses arêtes. La génération des sous-graphes est donc équivalente à la génération de motifs d'attributs (i.e. chaque attribut représentant une arête) avec la contrainte supplémentaire que l'ensemble des arêtes du motif courant forme un graphe connexe. Contrairement à l'algorithme n° 2 du chapitre 5, il n'est donc pas nécessaire de calculer le représentant canonique du motif courant pour vérifier si un motif isomorphe a été généré précédemment. Ce bénéfice est toutefois contre-balancé par un effet négatif qui apparaît lorsque deux sous-graphes de g_{sup}, traités comme deux motifs distincts, sont isomorphes. Dans ce cas, les scores et donc aussi

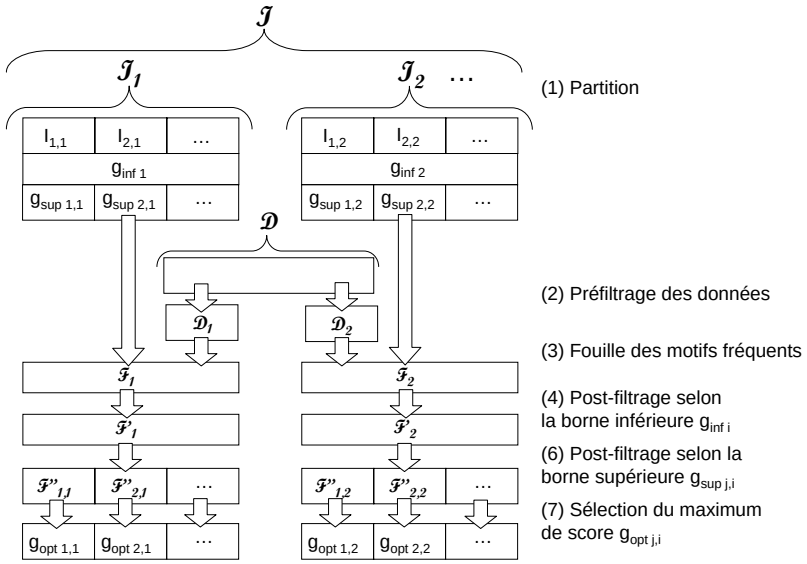


FIG. 6.18: Les différentes étapes et données intermédiaires de la méthode d'extraction fondée sur la recherche de sous-graphes fréquents. Les notations sont celles introduites dans la méthode exposée à la figure 6.17.

les fréquences de ces deux motifs, qui sont nécessairement égales, sont calculés deux fois, c'est-à-dire, une fois de trop.

L'exploration de cet espace d'état se fait selon un parcours en profondeur, afin de pouvoir exploiter la structure des listes d'occurrences (cf section 2.4.2) et ainsi de calculer rapidement les fréquences des motifs. L'exploration débute à partir de l'état initial, c'est-à-dire, du sous-graphe g'_{inf_i} de g_{sup}. Chaque état est un sous-graphe g de g_{sup} représenté par l'ensemble de ses arêtes $E(g) \subseteq E(g_{sup})$ dans g_{sup}. Les transitions état-à-état correspondent aux extensions qui permettent de passer du motif courant à un de ses successeurs immédiats. Dans le cas de la relation de sous-graphe partiel, une extension du sous-graphe g de g_{sup} consiste à ajouter une arête de g_{sup} qui n'est pas déjà dans g et qui est incidente à un sommet de g (afin de respecter la contrainte de connexité). La figure 6.19 illustre la génération de ces sous-graphes sur un exemple. Les arêtes surlignées en vert ou en bleu sont les arêtes faisant partie du motif courant, l'arête surlignée en bleue étant la dernière arête ajoutée. Le pseudo-code de la figure 6.20 décrit plus en détail l'algorithme `CrackReac`. Cet algorithme présente de nombreuses similitudes avec l'algorithme n° 2 du chapitre 5. Ainsi le parcours en profondeur de l'ordre des motifs se fait grâce aux appels récursifs à la fonction `developpe`, qui prend en arguments le motif courant g et son score s_g. Une différence importante – outre le fait déjà mentionné que l'espace de recherche est restreint aux sous-graphes de g_{sup} – est que `CrackReac` cherche l'optimum global et non plus tous les maxima locaux de la fonction de score. Contrairement à l'algorithme n° 2, l'algorithme de recherche est heuristique et ne garantit pas de trouver l'optimum global mais seulement un optimum local. `CrackReac` utilise pour ce faire une stratégie d'élagage originale. Cette stratégie repose sur la notion de branche et de niveau d'exploration associés au motif courant : la *branche courante* est l'ensemble des motifs intermédiaires qui ont été nécessaires pour construire le motif courant à partir du

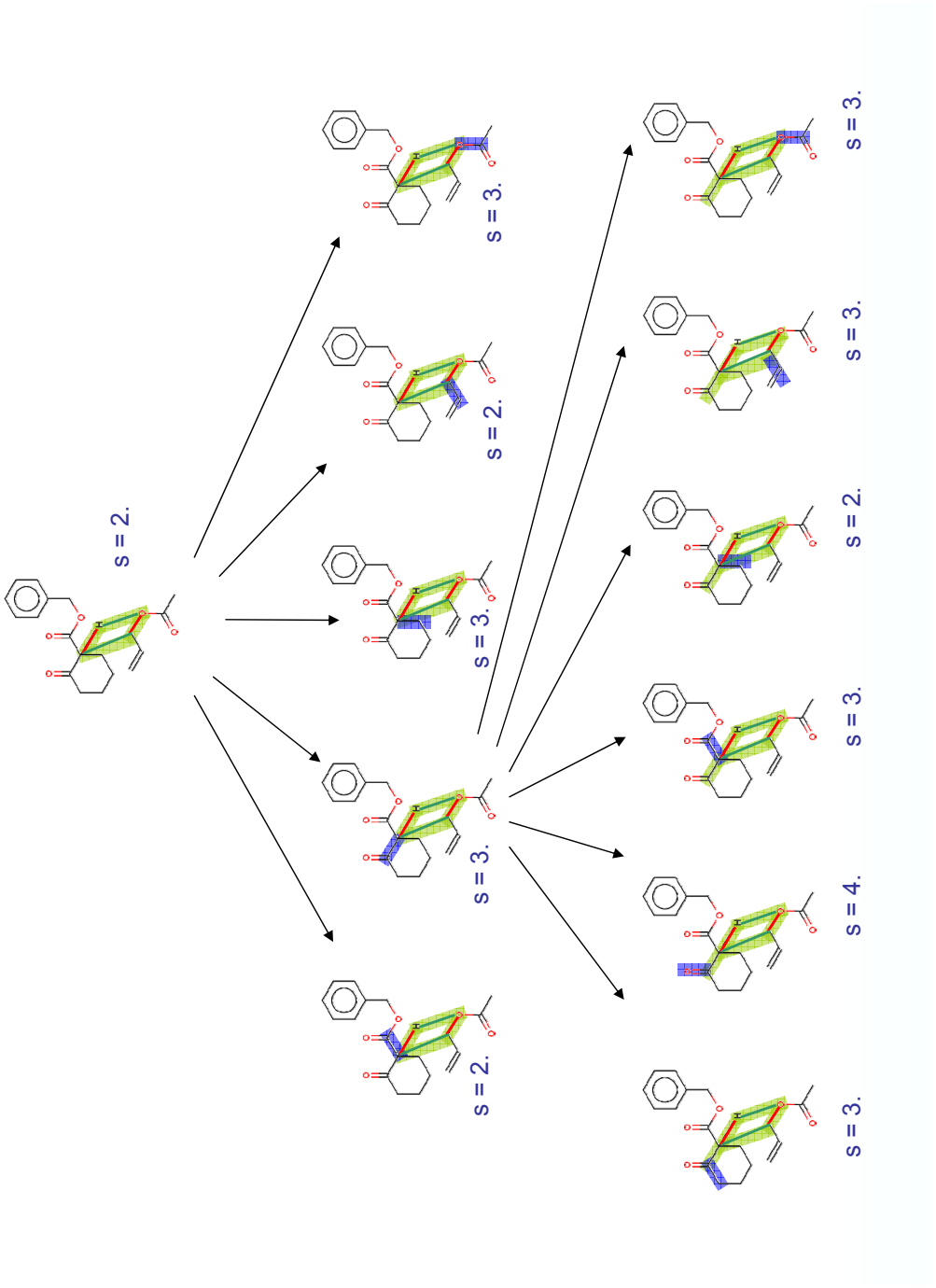


FIG. 6.19: Exemple de parcours en profondeur dans l'espace des sous-graphes de réaction

Données : Un intervalle (g_{inf}, g_{sup}), les données $\mathcal{D}$, une fonction de score s et son ordre des scores $(\mathbb{S}, \leq_{\mathbb{S}})$, les réels $\varepsilon_{branche}$ et $\varepsilon_{branche}$ compris entre 0 et 1
Résultat : Le motif optimal g_{opt} et son score s_{opt}

début
  Initialiser le score s_{opt} à 0 et créer le graphe g_{opt} vide ;
  Créer la liste vide D des extensions désactivées ;
  Créer le tableau $M_{niveau}[]$ des scores maximaux par niveau ;
1 Chercher les sous-graphes $(g_{inf_i})_{1 \leq i \leq n}$ de g_{sup} isomorphes à g_{inf} ;
  si $n > 0$ **alors**
    Calculer la fréquence $f_g \leftarrow \text{freq}_r(g_{inf_1}, \mathcal{D})$ et le score $s_g \leftarrow s(g_{inf_1}, f_g)$;
    pour chaque g_{inf_i} **faire**
      `developpe`$(g_{inf_i},\ s_g,\ s_g,\ |E(g_{inf_i})|)$
fin

fonction `developpe`(*motif courant* g*, son score* s_g*, score max. de la branche courante* $M_{branche}$*, niveau courant* d) **début**
  Créer la liste vide L ;
  pour chaque *extension* e *de* g *dans* g_{sup} *qui n'est pas dans* D **faire**
    Calculer la fréquence $f_e \leftarrow \text{freq}_r(e(g), \mathcal{D})$ et le score $s_e \leftarrow s(e(g), f_e)$ de $e(g)$;
    Ajouter (e, s_e) à L ;
    Mettre à jour $M_{niveau}[d] \leftarrow \max(M_{niveau}[d], s_e)$
  Trier par ordre décroissant de score les éléments de L ;
  pour chaque $(e, s_e) \in L$ **faire**
2   **if** $s_e \geq \max((1 - \varepsilon_{branche}) \cdot M_{branche}, (1 - \varepsilon_{niveau}) \cdot M_{niveau})$ **then**
      `developpe` $(e(g),\ s_e,\ \max(M_{branche}, s_e),\ d+1)$;
3     $D \leftarrow D \cup \{e\}$)
  pour chaque $(e, s_e) \in L$ **faire**
    $D \leftarrow D \setminus \{e\}$
  si $s_g > s_{opt}$ **alors**
    $s_{opt} \leftarrow s_g$; $g_{opt} \leftarrow g$
fin

FIG. 6.20: L'algorithme `CrackReac`

motif minimal initial g_{inf_i}. Le *niveau courant* est l'ensemble des motifs déjà générés qui sont associés à la même profondeur d'exploration, c'est-à-dire dans le cas présent, aux motifs déjà générés qui ont le même nombre d'arêtes que le motif courant. `CrackReac` maintient à jour le score maximal $M_{branche}$ de la branche courante et le score maximal $M_{niveau}(d)$ du niveau d courant afin d'effectuer un retour arrière dès que le score du motif courant g devient inférieur – à un facteur près – à l'un ou l'autre score maximal. Cette stratégie d'élagage est illustrée sur la figure 6.21. Les sigles en rouge représentent les branches élaguées. Une telle stratégie assure que l'élagage réalisé à un certain niveau, n'influe pas sur l'élagage aux niveaux inférieurs, et que le nombre de branches explorées (i.e de faisceaux de recherche) n'est par ailleurs pas limité. Ces deux caractéristiques réduisent ensemble le risque de convergence des faisceaux de recherche comme dans le cas de `Subdue`, du fait d'un nombre limité de faisceaux de recherche. Ainsi sur l'exemple de la figure 6.21, le fait que le meilleur score soit 4 au niveau 6, n'empêche pas `CrackReac` d'explorer les deux branches alternatives issues de motifs de score 3 au niveau 5. La condition précise d'élagage apparaît à la ligne 2 de l'algorithme 6.20. Les deux facteurs $1-\varepsilon_{branche}$ et $1-\varepsilon_{niveau}$ permettent de tolérer localement des décroissances de score. La ligne 3 interdit toute extension dont la branche a déjà été explorée de manière à ce que chaque motif – i.e. sous-graphe de g_{sup} – ne soit généré au plus qu'une fois (hors cas d'isomorphisme). Enfin dans le cas particulier de l'extraction de schémas CMS sous-jacents aux réactions, l'étape de recherche des sous-graphes de g_{sup} isomorphes à g_{inf} (cf ligne 1) peut être court-circuitée en initialisant n à 1 et en définissant g_{inf_1} comme le sous-graphe du graphe de réaction g_{sup} induit par les liaisons formées, modifiées et brisées.

Contrairement aux deux algorithmes présentés au chapitre 5, les deux algorithmes présentés dans ce chapitre ne sont pas complets et reposent soit sur un élagage des motifs dont la fréquence est inférieure à un seuil f_{min} fixé arbitrairement (dans le cas de la méthode de filtrage de motifs fréquents, cf section 6.3.2), soit sur l'algorithme de recherche heuristique présenté dans cette section et qui dépend également de paramètres ad hoc $\varepsilon_{branche}$ et ε_{niveau}. La section suivante vise à comparer l'efficacité respective des deux approches.

6.4 Expérimentation

Les tests qui ont été réalisés visent d'une part, à apprécier la qualité des résultats produits par la méthode dans le cadre de l'extraction des schémas CMS sous-jacents aux réactions et d'autre part, à comparer les performances – essentiellement le temps de calcul – de `CrackReac` et de la méthode proposée à la section 6.3.2, fondée sur la recherche des sous-graphes fréquents.

L'appréciation de la qualité des résultats consiste essentiellement à évaluer le niveau de pertinence et de robustesse de la fonction de score utilisée : une fonction de score est d'autant plus *pertinente* que les schémas $\mathcal{S}(g_{opt})$ de scores maximaux coïncident avec les schémas caractéristiques attendus. Une fonction de score est *robuste* si ces mêmes schémas de scores maximaux restent stables lorsque les données viennent s'enrichir de réactions de plus en plus variées. La pertinence est difficile à définir formellement lorsque la méthode est appliquée à un ensemble quelconque de réactions en entrée dans la mesure où leurs schémas caractéristiques sont indéterminés. C'est pourquoi les tests ont été réalisés sur des exemples de méthodes de synthèse dont on connaît le ou les schémas caractéristiques. Les méthodes étudiées sont ainsi la méthode de Sonogashira, l'époxydation assymétrique de Sharpless et la synthèse d'ester acéto-acétique, dont les schémas génériques caractéristiques sont représentés respectivement sur les figures 6.12, 6.22(a) et 6.22(b). Il devient alors possible d'évaluer quantitativement la pertinence d'une fonction de score en introduisant un critère d'erreur adapté, mesurant

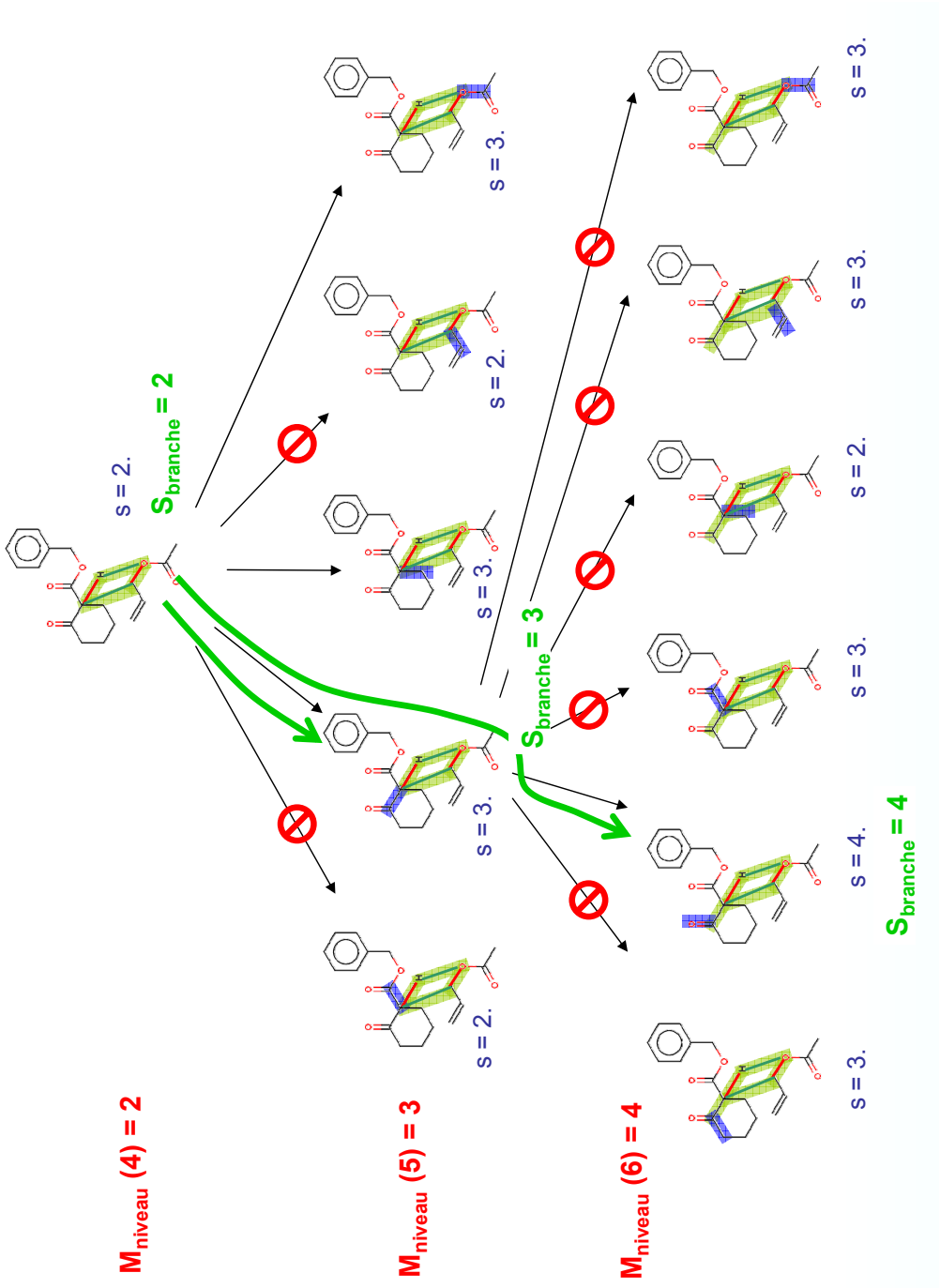


FIG. 6.21: Élagage selon le score maximal du niveau et de la branche courante (pour $\varepsilon_{branche} = \varepsilon_{niveau} = 0$)

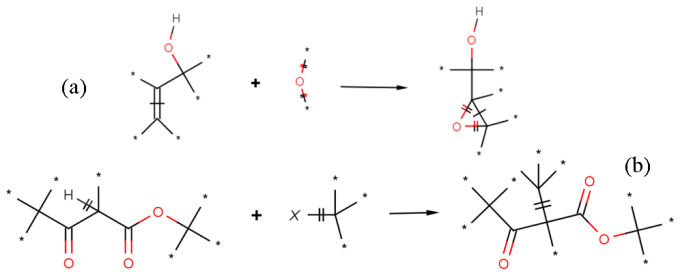


FIG. 6.22: Schémas caractéristiques généraux de l'epoxydation de Sharpless (a) et la synthèse d'ester acéto-acétique (b). Les étoiles représentent des atomes de type quelconque.

l'écart entre le schéma de score maximal et le schéma caractéristique attendu.

Définition 6.4.1. Étant donnés un sous-graphe g d'un graphe G et un graphe de référence g_{ref} isomorphe à au moins un sous-graphe de G, soit $(g_{ref_i})_{1\leq i\leq k}$ la liste des sous-graphes de G isomorphes à g_{ref}. Les arêtes *faussement positives* (resp. *faussement négatives*) de g relativement à g_{ref_i} sont les arêtes de g qui ne sont pas dans g_{ref_i} (resp. de g_{ref_i} qui ne sont pas dans g). Leurs nombres sont respectivement notés $\varepsilon^+_G(g, g_{ref_i}) = |E(g) \setminus E(g_{ref_i})|$ et $\varepsilon^-_G(g, g_{ref_i}) = |E(g_{ref_i}) \setminus E(g)|$. L'*erreur totale* ou simplement *erreur* $\varepsilon_G(g, g_{ref})$ de g relativement à g_{ref} est la valeur minimale atteinte par la somme des arêtes faussement positives et négatives :

$$\varepsilon_G(g, g_{ref}) = \min_{1\leq i\leq k} (\varepsilon^+_G(g, g_{ref_i}) + \varepsilon^-_G(g, g_{ref_i}))$$

Enfin l'*erreur positive* (resp. *négative*) est la valeur de $\varepsilon^+_G(g, g_{ref_i})$ (resp. $\varepsilon^-_G(g, g_{ref_i})$) pour le graphe g_{ref_i} qui minimise l'erreur totale.

Un test consiste alors, étant donnés un schéma CMS étendu S_{car} d'une méthode de synthèse et une liste $(E_i)_{1\leq i\leq n}$ d'équations chimiques de réactions pour lesquelles le schéma caractéristique attendu est S_{car}, à extraire le schéma caractéristique produit pour chaque graphe de réaction $\mathcal{G}(E_i)$, un sous-graphe $g_{opt_i} \subseteq \mathcal{G}(E_i)$. L'erreur moyenne $\overline{\varepsilon}$ est ensuite calculée pour quantifier la pertinence de la fonction de score :

$$\overline{\varepsilon} = \frac{\sum_i \varepsilon_{\mathcal{G}(E_i)}(g_{opt_i}, \mathcal{G}(S_{car}))}{n}$$

Cette erreur moyenne est calculée à l'issue de chaque test afin d'estimer la pertinence de la fonction de score. Par ailleurs chaque extraction d'un schéma caractéristique fouille un ensemble $\mathcal{D}$ d'exemples de réactions. Afin d'étudier la robustesse de la fonction de score, les données $\mathcal{D}$ sont constituées d'un mélange de 10 exemples[54] de la méthode étudiée (i.e. contenant le schéma S_{car}) et d'un nombre o variable d'autres réactions extraites aléatoirement des bases de données *ChemInform* et *RefLib*. Le mode opératoire de chaque test unitaire est résumé par le schéma synoptique de la figure 6.23.

L'effet des variations de o sur l'erreur moyenne permet d'apprécier la robustesse de la fonction de score, puisque l'ajout de réactions supplémentaires partageant le même cœur

[54]Ce choix arbitraire de 10 exemples se veut représentatif d'une méthode de synthèse peu représentée dans les données, mais suffisamment toutefois pour permettre une généralisation des exemples.

que la réaction présentée en entrée de l'algorithme, vont réduire progressivement le « pic » de score obtenu par S_{car} et ainsi perturber la convergence de l'algorithme vers S_{car}. Les erreurs positives, négatives et totales pour les différentes méthodes de synthèse considérées sont représentées sur les figures 6.25(a), 6.25(c) et 6.25(e) en fonction du nombre o des autres réactions. Les erreurs positives et négatives croissent de façon discontinue, lorsque le nombre o de réactions augmente. Les schémas produits en sortie sont très similaires au schéma caractéristique S_{car} de la méthode de synthèse pour les petites valeurs de o. Les schémas en sortie tendent toutefois à se contracter vers le cœur de la réaction lorsque o augmente. Ainsi alors que le schéma de score maximal se confond avec le schéma caractéristique de la méthode de Sonogashira lorsque les données contiennent 10000 réactions, le schéma de score maximal se désagrège dans le cas de la synthèse par époxydation asymétrique dès que o dépasse 750 (cf figures 6.24(a) et 6.24(b)). À l'inverse, l'erreur pour la synthèse des esters acéto-acétiques reste faible et constante, la fonction de score n'étant perturbée par aucune autre méthode concurrente (cf figure 6.24(c)). La faible robustesse de la fonction de score s_i utilisée est due au fait que cette fonction mesure davantage la représentativité d'un schéma que sa qualité à être un schéma CMS : tant que le schéma CMS S_{car} sous-jacent à la réaction E en entrée est représentatif des réactions des donnée qui partagent le même cœur que E, la convergence est facile. Mais lorsque o croit et que des réactions d'autres méthodes de synthèse partageant le même cœur sont insérées dans les données, la fréquence relative $\frac{\text{freq}_r(\mathcal{G}(S_{car}),\mathcal{D})}{\text{freq}_r(g_{inf},\mathcal{D})}$ du graphe $\mathcal{G}(S_{car})$ caractéristique attendu au sein des réactions partageant le même cœur, diminue et le pic de score associé à S_{car} s'estompe.

Du point de vue des performances, les figures 6.25(b), 6.25(d) et 6.25(f) comparent les temps de calcul[55] de `CrackReac` seul, de la méthode exposée à la section 6.3.2 utilisant `Gaston` (Nijssen et Kok, 2004) et un seuil f_{min} réglé à 0.8, et la méthode `CrackReac` avec prétraitement, c'est-à-dire en réalisant une partition $(\mathcal{I}_j)$ des intervalles, afin de réduire les données fouillées aux sous-ensembles $(\mathcal{D}_j)$ (cf section 6.3.2). `CrackReac` avec prétraitement apparaît être la solution la plus rapide, suivie de `CrackReac` seul et de la solution fondée sur la recherche de motifs fréquents. Toutefois, les résultats semblent trop variables d'un test à un autre pour pouvoir apporter une réponse définitive.

6.5 Conclusion

L'extraction des schémas CMS pose le problème de l'extraction de schémas réactionnels de très faibles fréquences. À des niveaux de fréquence inférieurs à 0,1 %, il devient difficile, voire impossible d'extraire de façon systématique tous les motifs les plus informatifs. Les algorithmes de recherche heuristique existants comme `Subdue` ne sont pas plus adaptés, dans la mesure où ces méthodes ont tendance à converger vers des motifs de score élevé, qui ne sont pas les plus pertinents. La solution proposée réduit le problème en introduisant un exemple (i.e. une réaction) particulier en entrée du processus, qui a pour effet de guider l'espace de recherche par l'introduction d'une contrainte supplémentaire. Cette contrainte permet d'accéder à des schémas réactionnels de très faibles fréquences et d'associer à chaque réaction fournie en entrée son schéma caractéristique. Cette association facilite la classification des réactions selon leur schéma caractéristique. La méthode proposée peut par son approche, être rattachée à l'apprentissage transductif. Formellement, le problème de l'extraction des schémas caractéristiques correspond, grâce au truchement du modèle des graphes de réactions, à un problème de recherche du graphe connexe maximisant une fonction de score au sein d'un in-

[55] Sur un Intel Core 2, 1,8 GHz

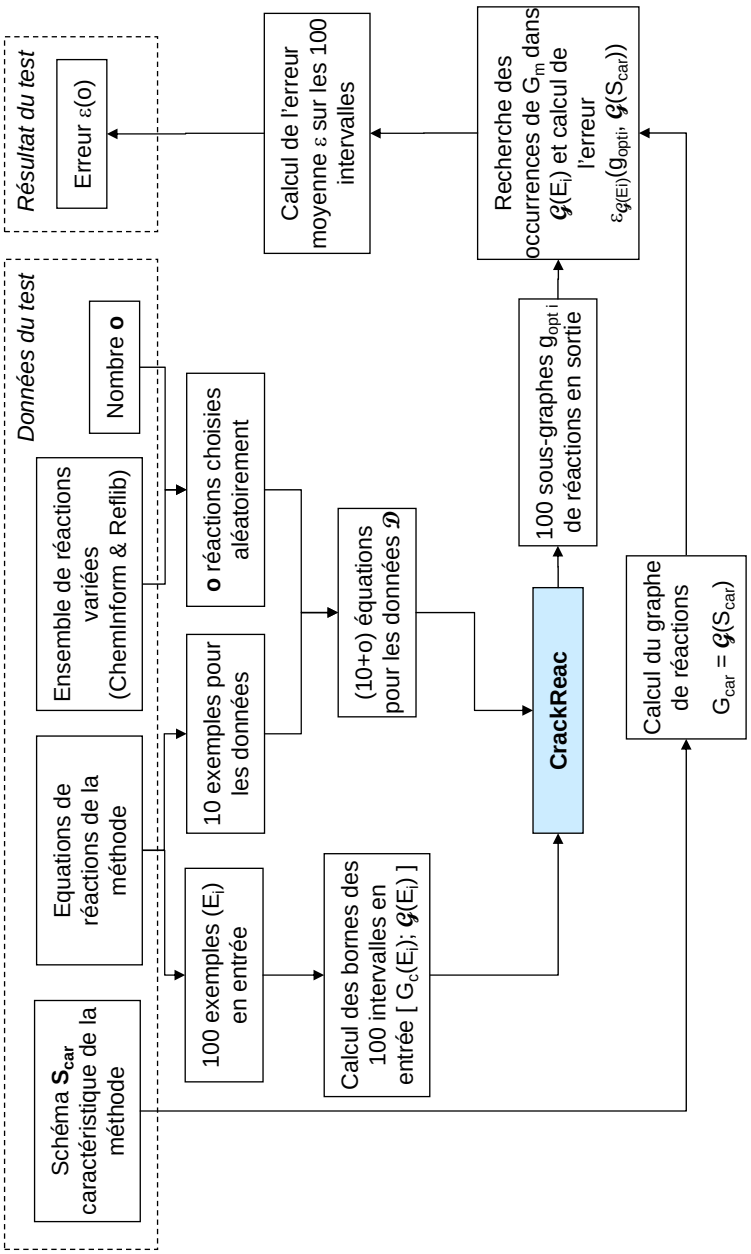


FIG. 6.23: Mode opératoire des tests

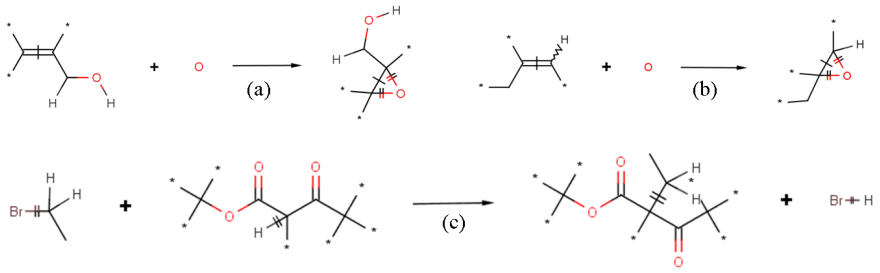


FIG. 6.24: Schémas optimaux pour l'époxydation de Sharpless, pour $o = 500$ et $o = 10000$ (a et b) et pour la synthèse d'ester acéto-acétique pour $o = 10000$ (c)

tervalle de graphes. Ce dernier problème peut se résoudre à l'aide d'une méthode utilisant les algorithmes existants de recherche de sous-graphes fréquents. Toutefois cette méthode n'exploite pas pleinement le fait de contraindre l'espace de recherche aux sous-graphes contenus dans un intervalle. Une approche alternative consiste à fouiller directement les sous-graphes connexes de la borne supérieure de l'intervalle, simplifiant ainsi grandement la procédure de génération des motifs puisqu'elle permet d'ignorer les conséquences néfastes intrinsèques à la fouille de graphes en cas de génération de motifs isomorphes. Cette observation a donné lieu à la mise au point de l'algorithme `CrackReac` de recherche heuristique au sein d'un intervalle de graphes. Les tests réalisés ont montré que `CrackReac` est plus rapide que la solution, pourtant optimisée, fondée sur les algorithmes de recherche de sous-graphes fréquents, tout en produisant des résultats identiques. L'exactitude de l'une ou l'autre des approches n'est toutefois pas garantie puisque `CrackReac` se fonde sur une recherche heuristique et que la méthode fondée sur la recherche de sous-graphes fréquents repose sur un élagage des motifs non fréquents. Mais surtout la fonction de score utilisée (qui est similaire à celle utilisée lors des tests au chapitre précédent) présente une robustesse limitée, cette fonction mesurant davantage la représentativité d'un schéma que sa qualité à être un schéma CMS.

Plusieurs axes d'amélioration sont donc envisageables. D'abord du point de vue de l'application à la synthèse organique, il est essentiel de mettre au point une fonction de score qui soit plus pertinente et plus robuste (au sens des définitions de la section 6.4). Afin que la fonction de score qualifie mieux les schémas CMS en tant que tels (et non en tant que schémas représentatifs selon s_i), il semble nécessaire d'intégrer davantage de connaissances du domaine dans la fonction de score, comme dans la modélisation des données (e.g. prise en compte des groupes fonctionnels, des conditions réactionnelles, etc). D'un point de vue informatique, une version exacte (i.e. non heuristique) de `CrackReac` est en cours de réalisation, fondée sur une approche Branch and Bound. L'étude approfondie des différentes stratégies de recherche heuristique peut également permettre de mieux comprendre et donc d'améliorer la convergence de l'algorithme. Enfin l'approche « transductive » de comparaison d'un graphe en entrée avec un ensemble d'exemples peut s'appliquer à d'autres problèmes. C'est notamment le cas du problème de la classification des sommets et des arêtes d'un graphe, abordé au chapitre suivant.

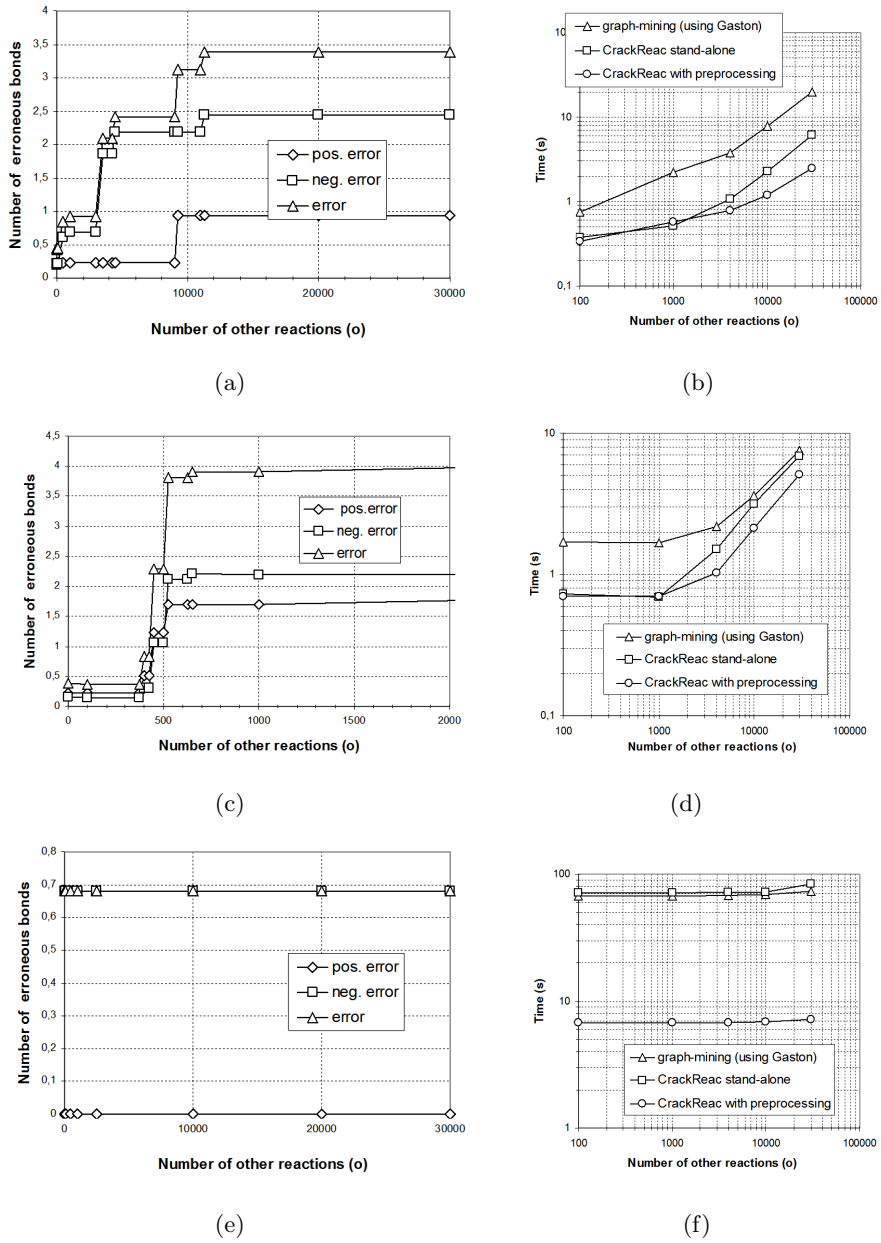


FIG. 6.25: Erreurs et temps de calcul pour les méthodes de Sonohashira (a et b), époxydation de Sharpless (c et d) et de synthèse des éthers acéto-acétique (e et f)

Chapitre 7

Méthode de classification des sommets fondée sur leur environnement. Application à la détermination des liaisons formables.

Sommaire

7.1 Introduction

Du point de vue applicatif, les chapitres précédents ont traité exclusivement de problèmes relatifs à la méthodologie de synthèse : les méthodes proposées ont fouillé les bases de données de réactions pour extraire des éléments de connaissances relatifs à ces réactions et aux méthodes de synthèse sous-jacentes. De ce point de vue, ce chapitre contraste avec les précédents puisqu'il aborde un problème lié davantage à la synthèse ciblée qu'à la méthodologie

de synthèse (cf chapitre 3) : autrement dit, l'objet n'est pas d'acquérir de nouvelles connaissances sur les réactions chimiques mais de fournir une information utile pour concevoir le plan de synthèse d'une molécule cible donnée. En particulier, ce chapitre s'intéresse à la notion d'*accessibilité synthétique*[56] : l'accessibilité synthétique d'une molécule exprime la facilité avec laquelle un expert peut établir un plan de synthèse opérationnel produisant cette molécule. Plus ce problème est difficile, plus faible est l'accessibilité synthétique de la molécule. Boda *et al.* (2007) rattache les travaux récents relatifs à l'accessibilité synthétique avec ceux plus anciens ayant trait à *faisabilité synthétique*. Alors que la première notion mesure la difficulté d'un problème sur une échelle, la seconde est un concept binaire : une molécule est « faisable synthétiquement » s'il est possible d'en faire la synthèse étant donnés certains moyens mis à disposition (matériel, temps disponible...).

Les travaux présentés traitent indirectement de l'accessibilité synthétique d'une molécule, en introduisant la notion originale de *formabilité* des liaisons. Une liaison est d'autant plus formable qu'il est facile pour un expert d'identifier une réaction chimique qui produise la molécule cible en formant cette liaison. Le degré de formabilité des liaisons d'une molécule est évidemment lié à l'accessibilité synthétique de cette molécule. En effet la synthèse d'une molécule ne peut être facilement envisagée que si certaines liaisons de cette molécule sont facilement formables. La réciproque est toutefois fausse : une molécule cible peut facilement être produite par une réaction, et comporter de ce fait plusieurs liaisons facilement formables, sans que pour autant la synthèse des réactants de cette réaction (i.e. les précurseurs) soit un problème facile.

L'objet de ce chapitre est de mettre au point une méthode capable d'estimer précisément la formabilité des liaisons d'une molécule cible à partir d'un algorithme de fouille de graphes rapide et capable de passer à l'échelle. Avant de présenter la manière dont ce problème est abordé, la section 7.1.1 montre en quoi l'information relative à l'accessibilité synthétique d'une molécule et donc à la formabilité des liaisons peut être exploitée par des applications telles que la rétrosynthèse ou le criblage virtuel. La section 7.1.2 analyse ensuite la notion de formabilité et identifie en conséquence les exigences que doit satisfaire un système d'apprentissage pour estimer la formabilité des liaisons. Cette analyse amène à la définition formelle du problème et à l'algorithme `GemsBond` développé pour le résoudre, présentés tous deux à la section 7.2.

7.1.1 L'accessibilité synthétique des molécules

La notion de formabilité des liaisons, même si elle n'apparaît pas sous ce terme, existe déjà en filigrane dans la rétrosynthèse à travers la notion de liaison stratégique. En effet, d'après la section 3.1.4, l'étape la plus déterminante de la rétrosynthèse est la phase stratégique lors de laquelle les objectifs structuraux présents dans la molécule cible sont identifiés. Parmi les cinq grandes classes de stratégies proposées par Corey, au moins trois d'entre elles (les stratégies fondées sur la topologie, la stéréochimie et les groupes fonctionnels) reviennent à identifier la ou les *liaisons stratégiques* qu'il faut chercher en priorité à déconnecter dans le sens rétrosynthétique, c'est-à-dire à former dans le sens synthétique. À l'origine, la définition de liaison stratégique selon Corey (cf pages 37 à 46 de Corey et Cheng (1995)) était essentiellement rattachée aux stratégies fondées sur l'analyse topologique de la cible : un ensemble de liaisons est stratégique du point de vue topologique si la suppression de ces liaisons dans le graphe moléculaire de la cible permet de réduire sensiblement la complexité topologique des fragments moléculaires résultants, encore appelés *synthons* par Corey et qui préfigurent la

[56] Traduction de « synthetic accessibility ».

structure des précurseurs. Un exemple simple de liaison stratégique est celui de la liaison en gras dans la molécule symétrique du squalène représentée sur la figure 7.1. Cette liaison est stratégique selon Corey (cf page 46 de Corey et Cheng (1995)) car sa déconnexion aboutit à deux synthons identiques et donc vraisemblablement, à des précurseurs deux fois plus petits que la cible initiale, réduisant ainsi d'autant la complexité globale du problème.

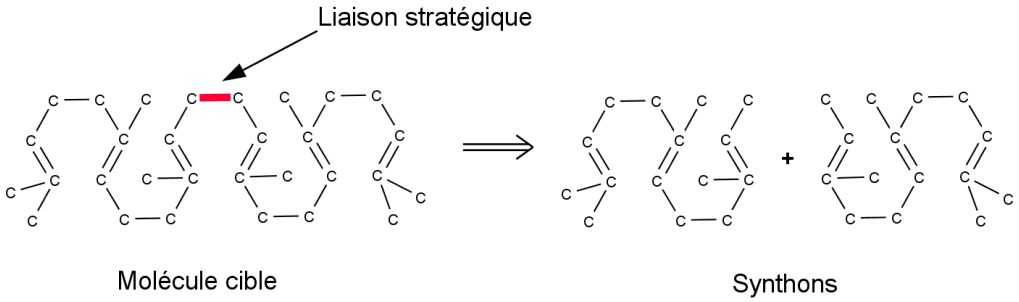


FIG. 7.1: La déconnexion d'une liaison stratégique dans la molécule symétrique du squalène conduit à deux synthons identiques.

Plusieurs méthodes d'analyse de graphes moléculaires ont été développées afin de proposer des liaisons stratégiques d'une molécule cible. Les premiers systèmes se sont orientés vers la définition topologique des liaisons stratégiques donnée par Corey. Toutefois ces systèmes ont depuis intégré d'autres facteurs, notamment de nature stéréochimique, de sorte qu'on peut dire aujourd'hui que le terme de liaison stratégique détient une acception plus large pour désigner toute liaison dont la déconnexion constitue un objectif prioritaire de la rétrosynthèse. Plusieurs fonctions de score ont été proposées pour quantifier la réduction de complexité topologique associée à une transformation rétrosynthétique. Une mesure proposée notamment par Bertz consiste à évaluer la diversité des sous-structures contenue dans les synthons obtenus après la déconnexion d'une liaison (Bertz et Sommer, 1997; Ruecker *et al.*, 2004). Une autre mesure plus facile à calculer et proposée par Tanaka, évalue la centralité des liaisons (Tanaka *et al.*, 2008b) : plus une liaison est centrale (i.e. dont la distance quadratique moyenne aux autres liaisons est faible), plus la déconnexion de la liaison aura des chances de simplifier le problème. Enfin le système `WODCA` d'assistance à la rétrosynthèse utilise des indicateurs simples comme la proximité d'un stéréocentre ou l'appartenance de la liaison à un cycle pour apprécier l'importance stratégique d'une liaison (Gasteiger *et al.*, 1992).

Ces scores présentent toutefois l'inconvénient de ne pas tenir compte de la formabilité des liaisons évaluées comme stratégiques. En effet, proposer une liaison éminemment stratégique mais dont la formation est impossible n'est en pratique d'aucune utilité puisqu'une telle proposition bloquera la progression de la rétrosynthèse. Certaines méthodes ont donc cherché à intégrer dans leur fonction de score non seulement la qualité stratégique de la liaison mais aussi sa formabilité. Toutefois, si la mesure de l'importance stratégique d'une liaison est un problème qui peut être formalisé assez facilement – en analysant la position relative de la liaison par rapport au système cyclique et aux stéréocentres de la cible – la notion de formabilité est plus difficile à apprécier et caractériser formellement. Le système `WODCA` intègre ainsi dans sa fonction de score des termes qui combinent différents effets physico-chimiques pour estimer la formabilité d'une liaison. Mais l'assemblage des différents facteurs et leur pondération ne s'appuie sur aucun argument théorique et reste du domaine de l'empirique. De même Tanaka propose d'associer à la centralité de la liaison d'autres facteurs relatifs à la formabilité des liaisons (Tanaka *et al.*, 2008a) mais tout aussi empiriques que ceux utilisés par

`WODCA`. Ces méthodes ne s'appuyant sur aucun modèle théorique de la formabilité, leur bien-fondé peut être remis en question. Tout au moins, une approche fondée sur l'apprentissage à partir d'exemples apparaît autant, sinon plus légitime pour apprécier la formabilité des liaisons. À notre connaissance, les seuls travaux qui ont permis de prédire les liaisons formables (sous le terme de liaisons stratégiques) à partir d'exemples ont été réalisés par Jean-Charles Régin à travers son algorithme `CNN` (Régin *et al.*, 1995; Régin, 1995). La qualité des résultats obtenus est d'ailleurs la principale raison qui a conduit à reconsidérer le problème de la découverte des liaisons stratégiques – rebaptisées liaisons formables pour être plus conforme à la réalité du problème traité – puis à développer un algorithme de fouille de graphes baptisé `GemsBond` pour le résoudre. La méthode `CNN` est décrit de façon détaillée puis comparée à `GemsBond` dans la section 7.3.4.

La notion de formabilité des liaisons n'est pas seulement utile dans le cadre de la rétrosynthèse, mais aussi dans celui du criblage virtuel à travers la notion émergente d'accessibilité synthétique. Le criblage virtuel (cf section 3.2.1) procède au filtrage d'un grand ensemble de molécules afin d'identifier les molécules candidates les plus adaptées à une application pharmaceutique donnée. Les molécules passées au crible sont soit issues de grandes bases de données soit générées à l'aide de modèles combinatoires. Les filtres utilisés mesurent généralement l'affinité avec laquelle ces molécules peuvent se lier en tant que ligands à un récepteur protéique donné mais d'autres critères importants comme la solubilité de la molécule sont également pris en compte. Toutefois l'accessibilité synthétique des molécules candidates n'est généralement pas intégrée dans le processus de criblage. La conséquence est que, de façon analogue à la prédiction des liaisons stratégiques en rétrosynthèse, le criblage peut proposer une molécule potentiellement très active mais dont la synthèse s'avère impossible ou tellement difficile que sa fabrication en deviendrait trop coûteuse. Boda *et al.* (2007) recense ainsi différents types de fonctions de score développées récemment à destination des méthodes de criblage virtuel, afin d'estimer grossièrement l'accessibilité synthétique d'une molécule. Ces fonctions reposent généralement sur la détection dans le graphe moléculaire cible, i) soit de la présence exacte de sous-structures qui sont réputées pour être difficiles ou au contraire faciles à synthétiser, ii) soit de la présence de structures significatives s'apparentant à des produits de départ disponibles, ce qui est censé rendre la cible plus facile à synthétiser. Le rôle de la fonction de score est alors de combiner empiriquement ces différentes informations pour produire en sortie un score unique. Tout comme pour la détermination des liaisons stratégiques, ces fonctions reposent sur des heuristiques qui sont discutables et/ou sur des calculs très lents par rapport aux exigences du criblage virtuel. Dans ce contexte, la conception d'un algorithme à la fois rapide et précis capable d'évaluer le niveau de formabilité des liaisons d'une molécule peut servir de filtre supplémentaire pour éliminer du crible les molécules dont aucune liaison n'est de formabilité suffisante.

7.1.2 La notion de formabilité des liaisons

Ce chapitre aborde le problème précédent de l'accessibilité synthétique en introduisant la notion de *formabilité* des liaisons des molécules. Intuitivement, la formabilité d'une liaison l dans une molécule cible M mesure la facilité avec laquelle un expert de la synthèse organique peut trouver une réaction qui synthétise la molécule cible M en formant la liaison l. Plus le problème de trouver une telle réaction est difficile, moins la liaison l est dite *formable* et plus faible est sa formabilité. La formabilité d'une liaison est une notion difficile à caractériser formellement même si elle correspond à une réalité perceptible par les chimistes. Ainsi la notion de liaison formable n'est pas un concept binaire mais une caractéristique relative : plusieurs

liaisons d'une même molécule cible sont généralement formables, mais le problème de trouver la réaction qui forme chacune d'entre elles peut, en fonction de la liaison considérée, s'avérer trivial ou au contraire être un véritable casse-tête. D'autre part la notion de formabilité est subjective dans le sens où elle repose sur les connaissances des chimistes forcément biaisées par leur propre expérience de la synthèse organique. Quoi qu'il en soit, un expert apprécie la formabilité d'une liaison principalement à partir de l'environnement structural dont dispose la liaison au sein de la molécule cible. Si par exemple, cet expert reconnaît dans l'environnement de la liaison un rétron, c'est-à-dire l'empreinte caractéristique laissée par une méthode de synthèse dans le produit principal de la réaction (et pour laquelle la liaison cible est formée), l'expert aura tendance à penser que cette liaison est vraisemblablement formable. Ainsi les deux liaisons en rouge et en gras dans le graphe moléculaire de la figure 7.2, sont incluses dans le rétron de la méthode de Diels-Alder (i.e. un cycle à six atomes de carbone caractéristique de l'empreinte laissée par la méthode de Diels-Alder et indiquée sur la figure 3.11(b)) dans lequel les liaisons en gras occupent la place des liaisons formées par la méthode. De plus l'en-

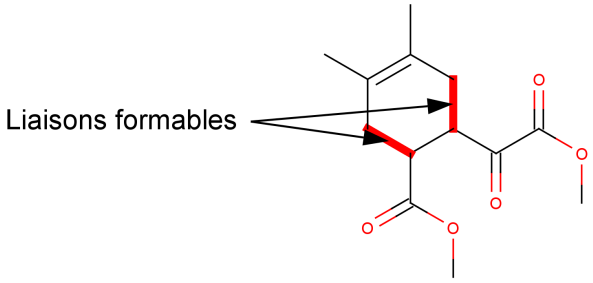


FIG. 7.2: Exemple de liaisons formables.

vironnement immédiat du rétron présente une fonctionnalité propice (sous la forme ici d'un groupe carboxyle et d'un groupe cétone) pour pouvoir appliquer la méthode de Diels-Alder (cf les explications fournies à ce sujet à la section 6.1.1). Pour ces raisons, l'expert identifiera vraisemblablement les deux liaisons rouges comme des liaisons formables.

Décider si une liaison est formable revient donc à décider si cette liaison dispose d'un environnement favorable à sa formation. Formaliser un algorithme qui puisse répondre à cette question n'est toutefois pas une tâche facile. Une des solutions les plus immédiates consiste à demander à un expert de spécifier la liste des environnements structuraux qu'il juge favorables à la formation d'une liaison. Un algorithme classifie alors une liaison comme formable s'il détecte, à l'aide d'une procédure de détection de sous-graphe isomorphe, la présence d'un environnement favorable autour de cette liaison. Dans le cas contraire, la liaison est classifiée non formable. Cependant les environnements ne sont pas tous autant favorables les uns que les autres. Certains sont même défavorables (i.e. leur présence handicape la formation de la liaison par exemple du fait d'un encombrement stérique). Pour prendre en compte cet effet, l'expert doit en plus distribuer les environnements spécifiés sur une « échelle de formabilité » allant du niveau très défavorable au niveau très favorable. L'algorithme de classification doit être modifié pour tenir compte du degré variable de formabilité. Une façon de faire est de déterminer le plus grand environnement que comporte la liaison cible parmi tous les environnements spécifiés par l'expert, et classifier la liaison cible en fonction du caractère favorable ou défavorable de cet environnement. Une telle méthode peut se comprendre comme un système expert, où les règles de décision ont pour hypothèses les environnements spécifiés par l'expert et pour conclusion le niveau de formabilité. Ce faisant, la solution proposée soulève les pro-

blèmes dont souffrent tous les systèmes experts. D'abord la distribution des environnements sur une échelle de formabilité est une tâche subjective qui repose sur les connaissances et la perception de l'expert. Ensuite se posent d'inévitables problèmes d'arbitrage quand plusieurs règles contradictoires s'appliquent. Enfin il est très difficile de mettre au point une base de connaissances qui soit exhaustive, cohérente et évolutive. En l'occurrence, il est très difficile, pour ne pas dire impossible, d'énumérer et d'étiqueter tous les environnements pertinents à prendre en compte.

Contrairement à ces systèmes experts, l'approche proposée dans ce chapitre se passe de l'intervention d'un expert, en se fondant sur l'apprentissage automatique à partir d'exemples d'environnements. Le principe fondamental de cette approche consiste à comparer les environnements de la liaison cible avec des exemples d'environnements de liaisons dont on connaît la formabilité. Cette approche adopte le « principe transductive » déjà décrit au chapitre 6, puisque la classification d'une liaison se fait directement en fouillant les exemples, sans extraire de modèle d'apprentissage intermédiaire (par exemple sous forme de règles de classification). La méthode bénéficie en outre de l'existence d'un grand nombre d'exemples de liaisons formées dans les BdR. Avant de décrire `GemsBond` plus en détail, il convient d'identifier les critères que doit satisfaire une telle méthode.

Granularité. La méthode d'apprentissage doit pouvoir accéder directement à toute la richesse combinatoire des graphes moléculaires, sans réduction préalable de l'information topologique. En effet, la formabilité d'une liaison est sensible à la plus légère modification de son environnement : le changement du type d'un atome ou d'une liaison à proximité de la liaison cible peut complètement changer le niveau de formabilité de la liaison. Or la plupart des méthodes QSAR/QSPR existantes utilisées pour résoudre des problèmes de classification et de régression à partir de graphes moléculaires (cf section 3.2.1) se décomposent en deux phases : i) en représentant l'information topologique d'une molécule par un vecteur de descripteurs numériques puis ii) en appliquant une méthode de classification (SVM, réseaux de neurones...) ou de régression numérique (régression linéaire, logistique, SVR ...) sur ces vecteurs. Les descripteurs numériques peuvent par exemple être des histogrammes de séquences d'atomes adjacents (Mahé *et al.*, 2005) ou de sous-graphes moléculaires dont la taille ne dépasse pas un certain seuil (Fröhlich *et al.*, 2005). Même si ce genre d'approche donne de bons résultats pour mesurer certaines grandeurs, par exemple le coefficient $logP$ de solubilité différentielle d'une molécule dans l'eau et l'octanol, le passage du graphe moléculaire à un vecteur de descripteurs, tout aussi grand soit-il, est forcément réducteur d'information. Pour cette raison, il est préférable pour prédire la formabilité des liaisons, de fouiller directement les sous-graphes des graphes moléculaires.

Autonomie. La méthode doit pouvoir fonctionner de façon autonome à partir des données brutes contenues dans les BdR, sans exiger l'intervention d'un expert. Dans l'idéal, la gestion du système se limite à l'ajout ou à la suppression de réactions dans l'ensemble des exemples fouillés.

Robustesse et mesurabilité. La méthode doit se fonder sur des indicateurs statistiques pour gérer l'incertitude liée aux données, par exemple pour tolérer des contradictions entre exemples ou une distribution déséquilibrée entre classes. Cette exigence élimine d'emblée les méthodes qui ne tolèrent pas la contradiction entre exemples, comme par exemple les méthodes fondées sur les espaces de versions (Mitchell, 1977) et adaptées aux graphes (Kramer et Raedt, 2001; Ganter *et al.*, 2004). Par ailleurs la formabilité d'une liaison n'est pas un concept binaire mais une notion relative : une liaison est plus

ou moins formable qu'une autre. Afin de rendre la méthode aussi précise que possible et être capable de trier les liaisons selon leur degré de formabilité, la méthode doit pouvoir associer à la liaison cible un score multi-valué exprimant la propension de cette liaison à être formable. Dans l'idéal, ce score est un indice de confiance, c'est-à-dire, un nombre réel borné entre 0 (la liaison n'a aucune chance d'être formée) et 1 (la liaison a toutes chances de pouvoir être formée).

Interprétabilité du résultat. La méthode doit pouvoir justifier du niveau de formabilité d'une liaison en accompagnant ce niveau d'une explication. Cette explication doit être interprétable par les chimistes et doit pouvoir contribuer à améliorer la connaissance relative à la formabilité des liaisons.

Rapidité et passage à l'échelle. La méthode doit être rapide et « capable de passer à l'échelle » dans la mesure où le problème traité nécessite de fouiller un grand nombre d'environnements dans un grand nombre d'exemples. La méthode doit être capable de fouiller au moins plusieurs milliers de graphes moléculaires pour pouvoir couvrir de façon représentative la plupart des environnements envisageables. La méthode doit idéalement présenter une complexité linéaire avec le nombre d'exemples, contrairement aux méthodes d'apprentissage par généralisation (i.e. de type bottom-up) dont la complexité est une fonction au moins quadratique avec le nombre d'exemples (Jauffret *et al.*, 1990; Régin *et al.*, 1995; Régin, 1995).

Ces différents critères justifient des choix de conception de la méthode `GemsBond` exposés dans la section suivante.

7.2 Une méthode de classification de sommets ou d'arêtes fondée sur la fouille de graphes

Cette section analyse le problème de l'estimation de la formabilité des liaisons et propose un algorithme de recherche heuristique baptisé `GemsBond` pour estimer la formabilité des liaisons à partir d'exemples. Pour ce faire, la section 7.2.1 formalise l'application considérée en des problèmes plus généraux de régression et de classification binaire supervisée. La section 7.2.2 analyse les tenants et aboutissants de ce problème, avant que la section 7.2.3 présente l'algorithme `GemsBond`. Enfin la section 7.2.4 montre comment les résultats produits par `GemsBond` peuvent servir à prédire les liaisons formables.

7.2.1 Formalisation du problème

Les problèmes de la découverte des liaisons formables et du classement des liaisons selon leur formabilité sont définis comme suit :

Définition 7.2.1. Soient :

- Une *molécule cible* représentée par son graphe moléculaire $G = (V(G), E(G))$.
- Une *liaison cible* $l \in E(G)$ particulière dans la molécule cible.
- Un ensemble $\mathcal{E}$ d'*exemples*, où chaque exemple est un graphe moléculaire $g \in \mathcal{E}$ d'une molécule auquel est associé le sous-ensemble $F(g) \subseteq E(g)$ des liaisons de g considérées comme formables.

Le problème de la *découverte des liaisons formables* consiste à décider si l'hypothèse « la liaison cible l est formable » est vraie étant donnés la position de l dans G et l'ensemble $\mathcal{E}$ des exemples. Le problème du *classement des liaisons selon leur formabilité* consiste à trier les liaisons de G selon leurs degrés de formabilité.

Ce dernier problème pour être bien défini, suppose l'existence d'une définition stricte de la formabilité, qui est, on l'a vu, une notion difficile à formaliser mais dont on peut donner toutefois une définition statistique approximative :

Définition 7.2.2. Étant données deux liaisons l_1 et l_2 d'une molécule cible G et une population d'experts à qui on demande de fournir l'ensemble des réactions qui, à leur connaissance, produisent la cible G, la liaison l_1 est dite *plus facilement formable* que la liaison l_2 s'il existe davantage de réactions connues des experts qui forment l_1 que l_2.

Le classement des liaisons selon leur formabilité revient à associer à chaque liaison de la molécule d'entrée G un *score* réel tel que ce score soit d'autant plus élevé que la liaison est formable. Le classement consiste alors à trier les liaisons par ordre décroissant de score. Comme cela a déjà été évoqué, la notion de liaison formable n'est pas un concept binaire, de sorte que le problème du classement des liaisons selon leur formabilité a plus de sens que celui de la découverte des liaisons formables. Ce dernier problème n'est considéré ici qu'afin de pouvoir évaluer la précision avec laquelle `GemsBond` classe les liaisons selon leur degré de formabilité. En effet la découverte de liaisons formables est un problème de classification binaire qui bénéficie d'une batterie de mesures statistiques établies (comme l'aire sous la courbe de ROC, la F-mesure...).

Indépendamment de la nature du problème traité (i.e. classement ou découverte des liaisons formables), il est utile de faire un autre distinguo selon la nature des exemples utilisés :

Étiquetage manuel. Dans le cas de données dites étiquetées manuellement, un expert de synthèse organique a identifié une par une, les liaisons formables (i.e. le sous-ensemble $F(g)$) dans une sélection d'exemples de graphes moléculaires. Même si un tel procédé repose forcément sur une perception subjective, cette approche permet d'aboutir à une annotation de qualité, tant que le nombre d'exemples ne dépasse pas quelques centaines. Au delà, l'étiquetage de milliers d'exemples devient hasardeux dans la mesure où cette tâche peut vite devenir ennuyeuse et de ce fait, occasionner des erreurs d'annotation.

Étiquetage automatique. Dans ce cas, les exemples sont constitués des produits principaux de réactions extraites de BdR. Les exemples de liaisons formables sont les liaisons formées par chaque réaction extraite. L'avantage de cette approche est de ne pas nécessiter l'intervention d'un expert et de disposer d'une quantité potentiellement illimitée d'exemples. Cependant cette approche présente aussi un inconvénient : puisqu'une molécule peut potentiellement être synthétisée par plus d'une réaction, une liaison qui n'est pas formée par la réaction extraite de la BdR peut très bien être formable par ailleurs. Certains exemples de liaisons étiquetées comme non formables sont donc en réalité des liaisons formables, en particulier dans les grandes molécules. Ce biais a pour effet une sous-estimation du nombre de liaisons formables dans les exemples.

Ces deux possibilités ont donc leurs avantages et leurs inconvénients respectifs. Cependant, comme l'objectif que nous privilégions est de pouvoir traiter de grandes quantités de données de façon autonome, les données utilisées par la suite correspondent à un étiquetage automatique, c'est-à-dire sont directement extraites des BdR.

Précisons enfin que le problème de la découverte des liaisons formables est une instance d'un problème plus général de classification supervisé qui a été introduit dans Pennerath *et al.* (2008a) comme le *problème de la classification de sommets (ou d'arêtes) fondée sur leur environnement*. On peut ainsi imaginer utiliser la méthode présentée pour classifier des objets (e.g. individus, sites web...) dans un réseau (e.g. réseau social, hyperliens...). En ce sens, la méthode est en rapport avec les problèmes traités par l'analyse de relations (cf

section 2.2.1), avec toutefois la faculté supplémentaire, de facilement tenir compte des cycles et autres motifs relationnels complexes. Le problème et la méthode `GemsBond` ne sont toutefois pas présentés ici dans leur cadre le plus général, pour des raisons de simplicité et parce que la méthode repose en partie sur le choix d'une heuristique spécifique à l'application.

7.2.2 Analyse du problème du classement des liaisons selon leur formabilité

Le problème du classement des liaisons formables revient, comme expliqué précédemment, à associer un score ou *indice de confiance*, noté $\text{conf}_G(l)$ mesurant la formabilité de la liaison l dans le graphe G. Cette confiance peut se calculer en fouillant dans les exemples $\mathcal{E}$ les occurrences des environnements que la liaison l présente dans la molécule cible G. Un *environnement* E d'une liaison l est défini formellement comme tout sous-graphe partiel connexe de G contenant l. La figure 7.3 présente deux environnements E_1 et E_2 d'une même liaison l. L'hypothèse de connexité des environnements est introduite en premier lieu pour réduire

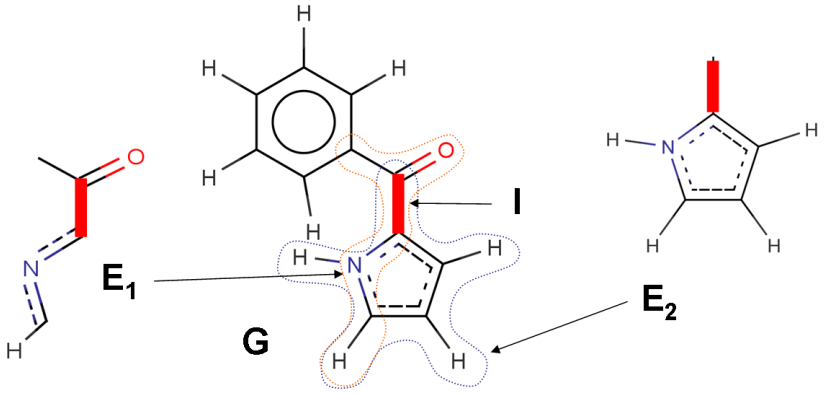


FIG. 7.3: Graphe moléculaire G de la molécule cible et deux environnements E_1 et E_2 de la liaison cible l.

efficacement le nombre d'environnements de l à considérer. Toutefois cette hypothèse n'est pas pour autant réductrice dans la mesure où un groupe d'atomes, comme un groupe fonctionnel, exerce son influence sur le caractère formable de la liaison l essentiellement à travers les liaisons covalentes qui le relient à la liaison l.

Une *occurrence* d'un environnement E dans un exemple $g \in \mathcal{E}$ est définie par un morphisme injectif μ de $V(E)$ vers $V(g)$ (cf définition de la section 2.3.1) préservant la relation d'incidence entre sommets et arêtes ainsi que les étiquetages de sommets et d'arêtes. La figure 7.4 représente trois occurrences de l'environnement E_1 introduit sur la figure 7.3 dans un ensemble de trois exemples. Une occurrence d'un environnement peut accréditer ou au contraire discréditer l'hypothèse selon laquelle la liaison l est formable. Une occurrence de E dans un exemple g est qualifiée de *positive* relativement à cette hypothèse si la liaison de g image de l selon μ est étiquetée formable, i.e. $\mu(l) \in F(g)$. Dans le cas contraire, l'occurrence est qualifiée de *négative*. Le nombre d'occurrences positives (resp. négatives) de E dans tous les exemples de $\mathcal{E}$ est noté $\text{occ}^+(E)$ (resp. $\text{occ}^-(E)$). Étant donné un unique environnement E de la liaison cible l, la probabilité pour que l soit formable connaissant son environnement E et relativement à la distribution des environnements de liaisons dans l'ensemble $\mathcal{E}$ des exemples,

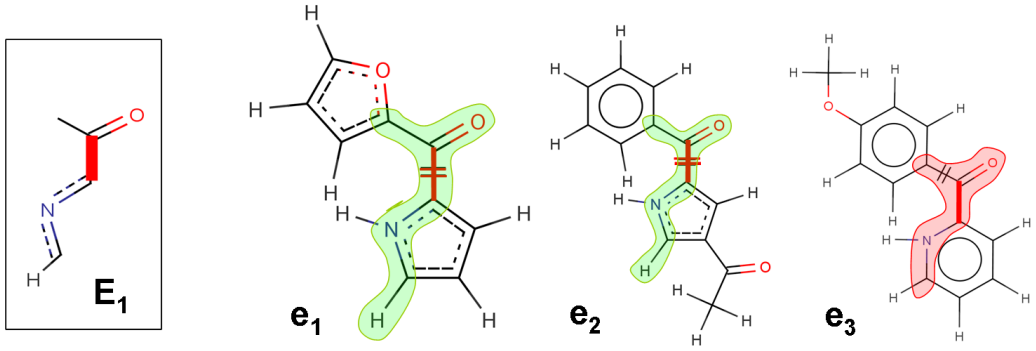


FIG. 7.4: Quelques occurrences de l'environnement E_1 dans trois exemples.

est :

$$P(l \text{ est formable} \mid \text{E est un environnement de } l) = \frac{\text{occ}^+(E)}{\text{occ}^+(E) + \text{occ}^-(E)}$$

Cette probabilité conditionnelle est appelée la *confiance* de l'environnement E dans l'hypothèse que la liaison l soit formable et est notée $\text{conf}(E)$. Sur l'exemple de la figure 7.4, seul l'exemple e_3 contient une occurrence négative puisque la liaison associée à l n'est pas formée par la réaction considérée. La confiance qui en résulte est donc égale à $\text{conf}(E) = 2/3$. Le terme confiance, comme le terme fréquence défini ultérieurement, sont empruntés au problème de l'extraction des règles d'association (Agrawal *et al.*, 1996), un environnement E pouvant être vu comme une règle associant à E l'hypothèse que l soit formable.

À supposer maintenant que l'on ne connaisse qu'un seul environnement E de l, cette liaison l devrait normalement être classifiée comme formable si et seulement si la confiance $\text{conf}(E)$ est plus grande que 0,5. En pratique, plusieurs raisons font que le problème est plus complexe. D'abord, dans le cas de l'étiquetage automatique dans lequel nous nous plaçons, le seuil de décision c_{min} est inconnu et inférieur à 0,5 puisque, comme cela a déjà été expliqué, le nombre d'exemples de liaisons formables est sous-estimé et rend ainsi la confiance de E inférieure à la probabilité conditionnelle qu'elle est censée estimée. Deuxièmement, le test suppose que l'environnement soit présent dans les exemples – i.e. $\text{occ}^+(E) + \text{occ}^-(E) > 0$ – pour que la confiance soit définie. Enfin et surtout, la liaison ne présente pas un mais plusieurs environnements qui doivent être considérés.

Le deuxième point, qui veut que l'environnement soit présent dans les exemples pour que la confiance associée soit définie, soulève la question plus générale de la représentativité d'un environnement dans les exemples : en effet, plus un environnement présente des occurrences dans des exemples nombreux et variés, plus l'estimation de la confiance sera représentative des exemples et par conséquent un support fiable pour prendre une décision. Afin d'associer un degré de vraisemblance à l'estimation de la confiance d'un environnement E, la *fréquence* $\text{freq}(E)$ est introduite comme étant la fréquence relative du graphe E dans les exemples de $\mathcal{E}$, c'est à dire comme la proportion des exemples qui contiennent au moins une occurrence de E, qu'elle soit positive ou négative. Plus élevée sera cette fréquence, plus fiable sera l'estimation de $\text{conf}(E)$. La fréquence $\text{freq}(E)$ est utilisée plutôt que le nombre $\text{occ}^+(E) + \text{occ}^-(E)$ total d'occurrences, car contrairement au nombre d'occurrences, la fréquence est une fonction décroissante dans l'ordre des motifs, conformément à l'idée de représentativité d'un motif. La confiance et la fréquence d'un environnement sont donc deux propriétés comprises entre 0 et 1.

En pratique, la liaison cible l n'a pas un, mais de nombreux environnements dans la molécule cible G. La détermination du niveau de formabilité de l nécessite de définir le sous-ensemble des environnements à considérer dans le calcul de l'indice de confiance global $\text{conf}_G(l)$. Il semble naturel de penser que plus un environnement E de l sera proche de G, plus sa confiance $\text{conf}(E)$ sera proche de $\text{conf}_G(l)$. Comme les environnements les plus proches de G sont aussi les plus grands (au sens de la taille, c'est à dire du nombre de liaisons de E), les environnements pertinents sont les environnements dits *maximaux* et notés $E_{\text{max. occ.}}$, c'est-à-dire les éléments maximaux parmi l'ensemble des environnements de l présents dans les exemples. Autrement dit, pour être maximal, un environnement E doit satisfaire simultanément les deux conditions i) la fréquence $\text{freq}(E)$ n'est pas nulle ii) tout environnement de l dans G qui contient strictement le sous-graphe $E_{\text{max. occ.}}$ a une fréquence nulle. De manière générale, il n'y a pas un mais plusieurs environnements maximaux $E_{\text{max. occ.}}$, comme l'illustre l'exemple de la figure 7.5. Dans cet exemple, les sous-graphes E_1 et E_2 du graphe G sont deux

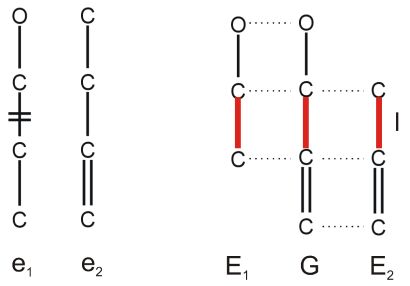


FIG. 7.5: Environnements maximaux E_1 et E_2 de l dans G par rapport aux données $\{e_1; e_2\}$.

environnements de l'arête l. Chacun de ces deux environnements E_1 et E_2 présente une seule occurrence dans les exemples, respectivement positive dans e_1 et négative dans e_2. Par ailleurs E_1 et E_2 sont maximaux car toute extension de E_1 ou de E_2 par l'ajout d'une liaison de G produit un environnement qui n'apparaît pas dans les exemples. Enfin tout environnement de l dans G de fréquence non nulle vis-à-vis des données $\{e_1, e_2\}$ est inclus soit dans E_1 soit dans E_2. Il s'ensuit que E_1 et E_2 sont les deux seuls environnements maximaux de l dans G. Or les confiances de E_1 et E_2, respectivement égales à 1 et 0, donnent des avis opposés quant à la formabilité de l alors que E_1 et E_2 sont difficilement distinguables par ailleurs puisqu'ils présentent la même taille (2 liaisons) et la même fréquence (1). Cet exemple montre combien la classification d'un sommet ou d'une arête fondée sur son environnement topologique est un problème difficile dans le cas général, pour lequel il ne semble pas devoir exister une méthode de résolution générale. Cependant il est possible de tirer avantage des particularités de l'application pour proposer une solution heuristique, comme cela est expliqué dans la section suivante.

7.2.3 L'algorithme `GemsBond` pour classer les liaisons selon leur formabilité

La classification d'un sommet ou d'une arête fondée sur son environnement topologique est un problème difficile dans le cas général. Afin de réduire la difficulté du problème, l'algorithme `GemsBond` (Pennerath *et al.*, 2008a) se fonde sur une heuristique spécifique au problème de la découverte des liaisons formables. Toutefois, si l'heuristique est spécifique à cette application, le principe général de fouille des environnements effectué par `GemsBond` est réutilisable pour tout problème de classification de sommets ou d'arêtes fondés sur leur environnement, du

moment qu'une heuristique adaptée au nouveau problème est disponible.

Choix d'une heuristique

L'heuristique choisie dans le cadre du classement des liaisons formables, considère que le caractère formable d'une liaison m dépend essentiellement de l'environnement de l qui est le plus favorable à l'hypothèse selon laquelle la liaison l est formable. En d'autres termes, l'heuristique introduit une asymétrie dans le problème de classification en considérant que les environnements de faible confiance (i.e. défavorables à ce que l soit formable) ont une influence négligeable, comparés aux environnements de confiance élevée (i.e. favorables à ce que l soit formable). À ce stade de l'analyse, ce choix est purement intuitif et peut à ce titre, être critiqué. Toutefois les tests réalisés et présentés à la section 7.3.1 montrent empiriquement le bien-fondé de ce choix. Formellement, cette heuristique signifie que la confiance globale $\text{conf}_G(l)$ est égale à la valeur maximale que la confiance $\text{conf}(E)$ peut atteindre parmi tous les environnements E de l :

$$\text{conf}_G(l) = \max_{E \text{ est un env. de } l}(\text{conf}(E))$$

La confiance maximale est atteinte par un *environnement explicatif* E_{max} ou simplement *explication* ainsi appelé dans la mesure où cet environnement justifie le niveau de confiance $\text{conf}_G(l)$ associé à la liaison l. Dans le cas où la confiance maximale est atteinte par plusieurs environnements, l'environnement le plus représentatif, i.e. dont la fréquence est la plus élevée, est choisi comme explication. La simplicité voulue de l'heuristique présente plusieurs avantages : d'abord l'explication se résume en un environnement unique qui facilite l'interprétation des résultats par un expert. Ensuite les tests réalisés montrent que les environnements explicatifs sélectionnés par l'heuristique sont généralement suffisamment fréquents pour être représentatifs, puisqu'ils sont de petite taille (généralement moins de 10 atomes) relativement à la taille des molécules cibles traitées. Enfin puisque les environnements E_{max} sont relativement petits, leurs occurrences dans les exemples peuvent être recherchées plus rapidement que celles d'environnements plus grands comme $E_{\text{max. occ.}}$, ce qui rend la méthode `GemsBond` d'autant plus rapide.

Algorithme de recherche

La détermination de la confiance d'une liaison consiste donc à rechercher l'environnement E_{max} de l qui présente la confiance la plus élevée. Tous les environnements de l ayant une fréquence non nulle dans les exemples sont des candidats valables pour être cet environnement E_{max}. Il n'est ainsi pas possible d'élaguer des branches de recherche comme dans le cas des motifs fréquents, dans la mesure où les confiances $(\text{conf}(E_i))_{1 \leq i \leq n}$ associées à une suite d'environnements imbriqués $E_1 \subseteq E_2 \cdots \subseteq E_n$, présentent des fluctuations imprévisibles, sans présenter de propriété régulière de décroissance ou croissance. Par ailleurs, les environnements de l de fréquence non nulle étant très nombreux, il est impossible de tous les énumérer. Enfin, l'évaluation de la confiance doit être répétée pour chaque liaison de la molécule cible G, afin de pouvoir classer les liaisons selon leur niveau de formabilité, et certaines applications, comme le criblage virtuel, peuvent nécessiter de traiter un grand nombre de molécules cibles. Pour toutes ces raisons, il est essentiel d'effectuer une recherche rapide de l'environnement E_{max}, quite à produire un résultat approximatif selon un algorithme de recherche heuristique.

La méthode de recherche développée présente dans son principe une forte similitude avec l'algorithme `CrackReac` présenté au chapitre 6 pour extraire le schéma CMS sous-jacent à une

réaction. Tout comme `CrackReac`, `GemsBond` fait croître un sous-graphe connexe (i.e. un environnement courant $E_{courant}$ selon un parcours en profondeur dans un espace d'état constitué des environnements de l. Dans le cas de `CrackReac` , le motif initial était le graphe de cœur de la réaction alors que dans le cas de `GemsBond` , le motif initial est le sous-graphe $g_G(l)$ de G induit par la liaison cible l (et plus généralement par tout sommet ou arête d'un graphe). `GemsBond` peut donc être vu comme une instance particulière de `CrackReac` associée à l'intervalle $[g_G(l); G]$ des environnements de l et à la fonction de score $E \mapsto \text{conf}(E)$ associant à l'environnement E sa confiance. De ce fait, `GemsBond` s'appuie sur le même algorithme d'énumération que `CrackReac` pour générer les motifs dans l'intervalle $[g_G(l); G]$: `GemsBond` génére récursivement l'ensemble des sous-graphes partiels connexes de G contenant l, en ajoutant au sous-graphe courant g de G toute arête dans $E(G) \setminus E(g)$ incidente à g. Plus fondamentalement, `GemsBond` adopte la même « approche transductive » que `CrackReac` : plutôt que d'extraire des exemples un modèle constitué d'un ensemble d'environnements de liaisons caractéristiques de différents niveaux de confiance (induction), puis d'utiliser ce modèle pour prédire le niveau de confiance d'une nouvelle liaison l (prédiction), les environnements de la liaison l sont directement comparés avec ceux des exemples pour déterminer le niveau de confiance de l (transduction). `CrackReac` et `GemsBond` traitent toutefois de problèmes différents : alors que `CrackReac` se rattache à un problème de classification non supervisée qui associe à un graphe un sous-graphe caractéristique, `GemsBond` se rattache à un problème de classification supervisée qui associe à un sommet ou une arête d'un graphe une catégorie (i.e. formable ou non formable). `CrackReac` et `GemsBond` diffèrent donc à la fois par la nature du problème de classification traité et par la nature des objets auxquels le problème de classification est rattaché (i.e. un graphe dans le cas de `CrackReac`, un sommet ou une arête dans le cas de `GemsBond`). Cette différence se retrouve dans le choix des fonctions de score utilisées (fonction de score informative pour `CrackReac`, confiance pour `GemsBond`).

Afin de ne pas énumérer tous les environnements de l – le nombre d'environnements de la liaison l croit exponentiellement avec leur taille comme le montre la courbe de la figure 7.20 – une stratégie d'élagage est utilisée pour limiter l'exploration de l'espace d'état. Le principe de l'élagage est similaire à celui de `CrackReac` au sens où `GemsBond` adopte le principe d'une recherche gloutonne, en choisissant comme extension e de l'environnement courant $E_{courant}$, celle qui conduit à la plus forte augmentation de confiance $\text{conf}(e(E_{courant})) - \text{conf}(E_{courant})$, jusqu'à ce qu'aucune extension ne puisse faire croître davantage la confiance. L'environnement $E_{courant}$ est alors un maximum local de la confiance et devient à ce titre le motif explicatif de la liaison. La convergence rapide vers un maximum local de l'algorithme glouton se fait au prix d'un risque accru que le maximum local trouvé soit très différent structurellement du maximum global de confiance recherché. Afin de réduire ce risque et d'éviter une convergence précoce vers un maximum local sous-optimal, toutes les extensions de l'environnement courant sont développées – indifféremment de leur confiance ou fréquence – tant que leur taille (i.e. le nombre d'arêtes) reste inférieure ou égale à un paramètre d_{min}. Afin également de contrôler la représentativité des environnements explicatifs retournés, un autre paramètre $f_{min} \in [0; 1]$ est introduit afin de ne fouiller que les environnements de fréquence supérieure ou égale à f_{min}. Le pseudo-code de l'algorithme `GemsBond` est détaillé sur la figure 7.6. La boucle principale énumère toutes les extensions possibles e de l'environnement courant $E_{courant}$ dans le graphe d'entrée G (cf ligne 1) et évalue la confiance c et la fréquence f de l'environnement $e(E_{courant})$ ainsi étendu (cf ligne 2). Seuls les environnements dont la fréquence est suffisante et dont, soit la confiance est maximale (localement), soit la taille est plus grande que d_{min}, sont développés davantage grâce à un appel récursif à la procédure `developpe` (cf ligne 6). Calculer la confiance et la fréquence de $E_{courant}$ (cf ligne 2) nécessite de compter toutes

Données : Le graphe G et l'arête l en entrée, l'ensemble d'exemples $\mathcal{E}$, les seuils d_{min} et f_{min}
Résultat : L'explication E_{max}, sa confiance c_{max} et sa fréquence f_{max}

début
  Créer $E_{courant}$ comme le sous-graphe de G induit par l'arête l ;
  Calculer la confiance $c \leftarrow \text{conf}(E_{courant}, \mathcal{E})$ et la fréquence $f \leftarrow \text{freq}_r(E_{courant}, \mathcal{E})$ de $E_{courant}$;
  Créer l'ensemble vide D des extensions désactivées ;
  `developpe` ($E_{courant}$, c, f, 1)
fin

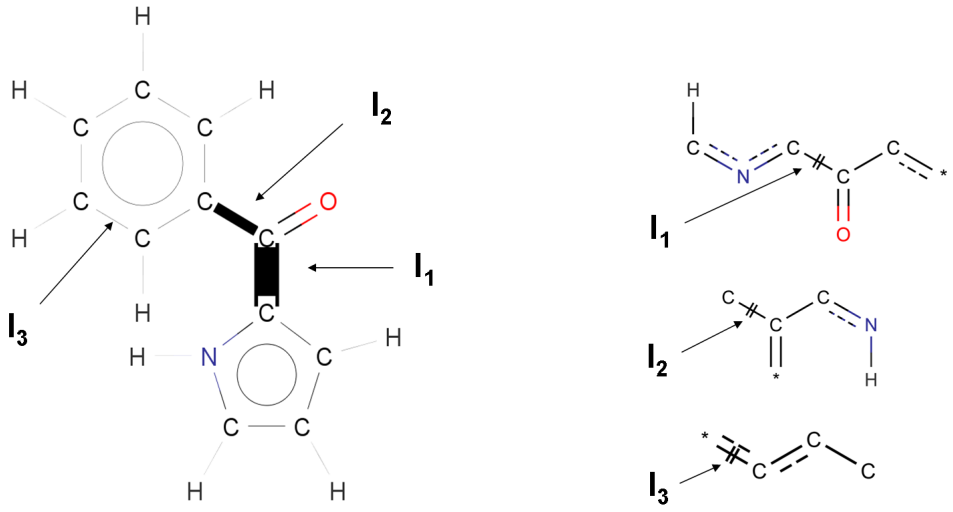
fonction `developpe`(*env.* $E_{courant}$, *conf.* $c_{courante}$, *fréq.* $f_{courante}$, *profondeur* d)
début
  Créer l'ensemble $C \leftarrow \emptyset$ des extensions à développer ;
  Créer l'ensemble vide D_e des extensions désactivées localement ;
  Créer la confiance $c_{\text{local max}} \leftarrow 0$;
1 **pour chaque** *extension* e *de* $E_{courant}$ *dans* G *mais pas dans* D **faire**
2   Calculer la confiance $c \leftarrow \text{conf}(e(E_{courant}), \mathcal{E})$ et la fréquence $f \leftarrow \text{freq}_r(e(E_{courant}), \mathcal{E})$ de $e(E_{courant})$;
3   **si** $f \geq f_{min}$ *et* $c > 0$ **alors**
       si $d < d_{min}$ **alors**
         $C \leftarrow C \cup \{(e, c, f)\}$
       sinon
4   **si** $c \geq c_{local\ max}$ *et* $c > c_{courante}$ **alors**
           si $c > c_{local\ max}$ **alors**
             $c_{\text{local max}} \leftarrow c$; $C \leftarrow \emptyset$
           $C \leftarrow C \cup \{(e, c, f)\}$
     sinon
       $D_e \leftarrow D_e \cup \{e\}$
5 $D \leftarrow D \cup D_e$;
   pour chaque $(e, c, f) \in C$ **faire**
6   `developpe` ($e(E_{courant})$, c, f, $d + 1$)
7 $D \leftarrow D \setminus D_e$;
   si $c_{courante} > c_{max}$ *ou* ($c_{courante} = c_{max}$ *et* $f_{courante} > f_{max}$) **alors**
     $c_{max} \leftarrow c_{courante}$; $f_{max} \leftarrow f_{courante}$; $E_{max} \leftarrow E_{courant}$
fin

FIG. 7.6: L'algorithme `GemsBond`

les occurrences positives et négatives de $E_{courant}$ dans les exemples $\mathcal{E}$. La recherche de ces occurrences s'appuie sur la structure de liste d'occurrences déjà introduite à la section 2.4.2. Cette structure a été modifiée afin de comptabiliser distinctement chaque occurrence selon qu'elle soit positive ou négative, et ce en une seule passe sur les données. Ainsi le calcul de la confiance d'un environnement ne prend pas plus de temps que celui de sa fréquence, même si en théorie la confiance est le rapport de deux nombres indépendants d'occurrences positives et négatives. Afin d'améliorer la rapidité de `GemsBond`, les extensions qui conduisent à un environnement dont la confiance est nulle ou dont la fréquence est inférieure au seuil f_{min} (cf ligne 3) sont élagués et ne sont plus considérés par les appels récursifs consécutifs (cf lines 5 et 7). Cet élagage permet de réduire les temps de calcul sans modifier les résultats puisque les deux prédicats $\text{conf}(E_{courant}) > 0$ et $\text{freq}_r(E_{courant}) \geq f_{min}$ sont anti-monotones (i.e. ne peuvent passer de faux à vrai lorsque l'environnement courant est amené à croître).

Résultats produits par `GemsBond`

`GemsBond` produit pour chaque liaison l de la molécule en entrée, une confiance $\text{conf}_G(l)$, un environnement explicatif E_{max} et sa fréquence $\text{freq}(E_{max})$. La confiance peut être utilisée pour moduler l'épaisseur de chaque liaison (i.e. plus le tracé d'une liaison est épais, plus la formabilité de la liaison est élevée) de manière à ce que l'expert puisse saisir rapidement et visuellement la distribution de la confiance dans la molécule cible. Un exemple de résultat est donné sur la figure 7.7(a) pour le graphe moléculaire particulièrement simple déjà donné en exemple à la figure 7.3. Cette molécule est le produit principal d'une réaction qui forme la



(a) Distribution de la confiance

(b) Explications associées

FIG. 7.7: Résultat produit par `GemsBond` pour la molécule de la figure 7.3

seule liaison l_1. `GemsBond` identifie bien l_1 comme la liaison la plus facilement formable (i.e. de plus forte confiance). Au contraire, le tracé très fin des liaisons aromatiques ou les liaisons C-H souligne leur faible formabilité, en accord avec les connaissances et la pratique de la synthèse organique. La figure 7.7(b) et le tableau de la figure 7.8 rapporte respectivement les

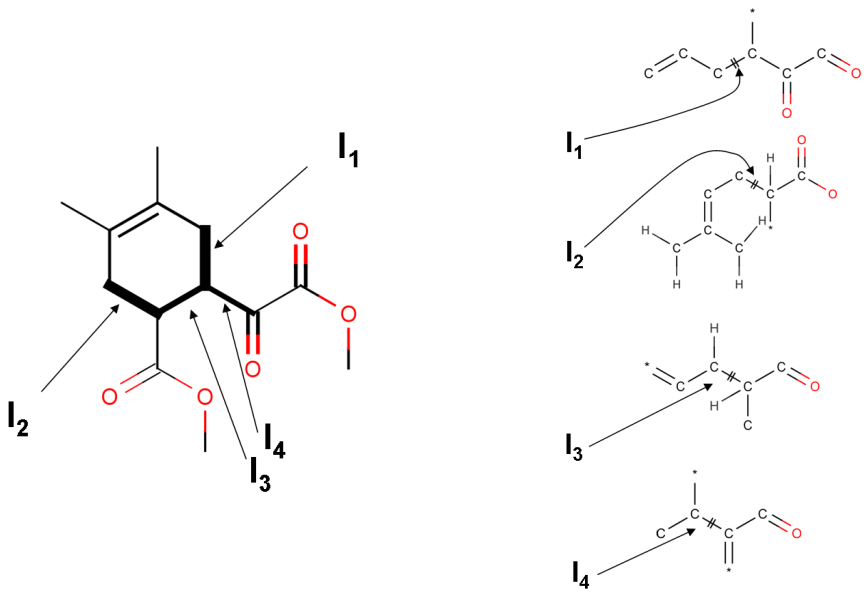
explications E_{max} ainsi que les confiances et fréquences associées aux trois liaisons identifiées l_1, l_2 et l_3 de la molécule. Chaque environnement explicatif représente conventionnellement la liaison à laquelle il se réfère par une liaison formée (traversée par deux traits orthogonaux). Les

Liaison l	conf(l)	freq$_r(l)$
l_1	0,89	$6/6738 = 0{,}1$ %
l_2	0,37	$57/6738 = 1$ %
l_3	0,002	$4235/6738 = 63$ %

FIG. 7.8: Confiances et fréquences absolues associées aux liaisons de la figure 7.7(a)

environnements explicatifs rattachés aux liaisons de confiance élevée (comme l_1) ont tendance à être plus grands et plus complexes que ceux des liaisons de faible confiance (comme l_3) et donc aussi de plus faible fréquence. Comme on pouvait s'y attendre, les environnements de forte confiance comportent également davantage de groupes fonctionnels ou du moins de fragments de groupes fonctionnels.

De manière analogue, l'exemple contenant le rétron de la méthode de Diels-Alder (cf figure 7.2) aboutit aux résultats présentés sur les figures 7.9(a) (graphe de sortie), 7.9(b) (explications) et 7.10 (confiances et fréquences). Les deux liaisons l_1 et l_2 les plus formables



(a) Distribution de la confiance

(b) Explications associées

FIG. 7.9: Résultat produit par `GemsBond` pour la molécule de la figure 7.2

(confiance de 1 et 0,95) correspondent exactement aux liaisons formées dans le rétron de la méthode de Diels-Alder. Les explications associées ne correspondent toutefois pas au rétron comme on aurait pu s'y attendre. `GemsBond` se contente en effet de l'environnement minimal (par la taille) qui maximise la confiance : `GemsBond` ne complète donc pas l'environnement explicatif de l_1, qui a déjà la plus grande confiance possible égale à 1, pour aboutir au rétron attendu. Cet écart entre l'environnement fourni par `GemsBond` et celui fourni par

Liaison l	conf(l)	$\text{freq}_r(l)$
l_1	1	7/7537 = 0,1 %
l_2	0,95	10/7537 = 0,1 %
l_3	0,57	236/7537 = 3 %
l_4	0,49	191/7537 = 2,5 %

FIG. 7.10: Confiances et fréquences absolues associées aux liaisons de la figure 7.9(a)

les chimistes pour expliquer la formabilité d'une liaison peut être perçu par les chimistes comme une faiblesse de la méthode. Une des améliorations envisageables de `GemsBond` est donc d'aménager l'algorithme pour qu'il réduise cette divergence d'explication (notamment en « forçant » l'inclusion dans les explications de groupes fonctionnels entiers et non pas seulement de fragments). Toutefois certaines divergences entre les explications fournies par `GemsBond` et celles attendues par les chimistes ne doivent pas forcément discréditer pour autant les explications produites par l'algorithme. Ces dernières peuvent dans certains cas, révéler des règles statistiques inattendues et ainsi apporter un point de vue complémentaire à celui des spécialistes de la synthèse, qui n'est pas biaisé par les connaissances du domaine. Par exemple, l'explication de la liaison l_1 présente deux groupes aldéhydes et un alcène qui pourront paraître insignifiants pour l'expert, en comparaison du rétron de la méthode de Diels-Alder attendu. Toutefois cet environnement a une confiance de 1 : autrement dit, cet environnement ne souffre d'aucun contre-exemple et lorsque celui-ci apparaît dans les exemples, la liaison de référence est systématiquement formée. Ce genre d'observations permet de soulever de nouvelles questions, qui sont autant de pistes d'investigation pour améliorer la connaissance relative à la pratique de la synthèse organique. Ainsi, dans la mesure où la fréquence absolue de l'environnement précédent est seulement de 7 parmi 7537 exemples, que se passe t-il si on élargit l'étude en recherchant l'environnement trouvé dans une grande BdR. Va t-on voir apparaître des contre-exemples à l'explication et dans quelles proportions ? Le cas échéant, quelles sont les éléments structuraux discriminants, qui sont présents dans les exemples et absents dans les contre-exemples, et inversement ? On le voit bien sur cet exemple : un outil comme celui de `GemsBond` ouvre la voie à une nouvelle manière d'interroger les bases de données de réactions ou de molécules.

`GemsBond` est donc non seulement une méthode utile pour évaluer la formabilité des liaisons mais aussi un outil d'extraction de connaissances. Ainsi, chaque fois que `GemsBond` traite de nouvelles molécules cibles, les explications nouvellement produites sont collectées et archivées avec celles obtenues lors des traitements précédents. La tête de la liste des explications archivées triées par ordre décroissant de confiance puis de fréquence permet d'identifier des environnements récurrents, caractéristiques de liaisons de formabilité élevée. La mise à disposition de ces environnements auprès des experts contribue à faire de `GemsBond` un outil d'extraction de connaissances à partir de BdR. La figure 7.11 donne par exemple les six premiers environnements de la liste des explications formant une liaison carbone-carbone, triées par ordre décroissant de confiance puis de fréquence. Cette liste est obtenue après avoir traité seulement 100 molécules. Ces environnements ont tous une confiance de 1. Par conséquent toutes leurs occurrences dans les exemples sont positives. Par ailleurs tous ces environnements ont une fréquence absolue non négligeable, comprise entre 20 et 40, permettant de calculer avec précision l'estimation de la confiance. Tous ces environnements, qui sont caractéristiques de liaisons hautement formables, font apparaître des combinaisons d'hétéroatomes

et de liaisons multiples ou aromatiques.

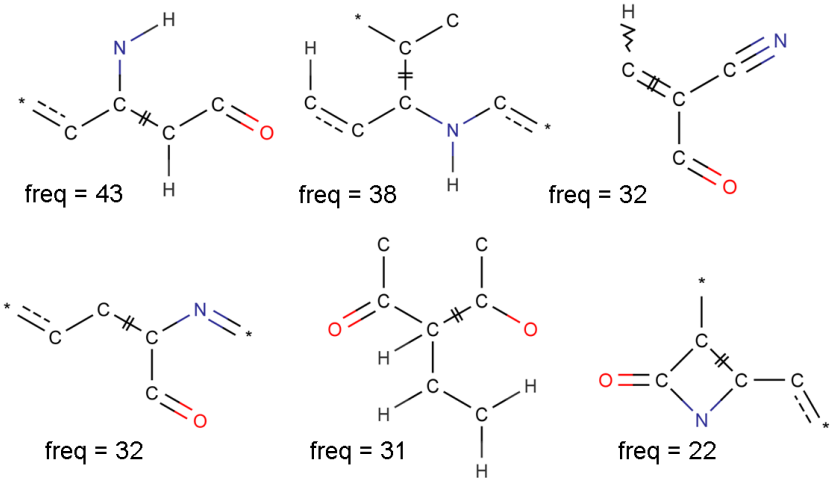


FIG. 7.11: Six environnements explicatifs de confiance 1, pour former des liaisons carbone-carbone

7.2.4 Classifieurs binaires pour prédire les liaisons formables

Une fois que les indices de confiance $\text{conf}_G(l)$ sont calculés pour toutes les liaisons l de G, il est possible de répondre simplement au premier problème introduit à la section 7.2.1 du classement des liaisons selon leur formabilité, en triant les liaisons de G par ordre décroissant de confiance. Afin de répondre au second problème de découverte des liaisons formables – qui pour rappel, n'a d'autre but que d'évaluer les performances de `GemsBond` à l'aide des mesures statistiques propres aux méthodes de classification supervisée – un classifieur binaire doit préalablement transformer les niveaux de confiance produits par `GemsBond` en une variable booléenne indiquant si oui ou non chaque liaison l de G est formable. Puisque les différents types de liaisons (simple, double, triple et aromatique) ont des distributions statistiques différentes dans les exemples et en particulier, présentent des proportions variables de liaisons formées, chaque type de liaison nécessite son propre classifieur. Si de nombreux classifieurs sont envisageables, seuls trois d'entre eux choisis parmi les plus évidents sont considérés ci-après :

Seuillage sur la confiance Un tel classifieur, dédié aux liaisons de type T et paramétré par un seuil $c^T_{min} \in [0;1]$, classifie une liaison l de type T comme formable si et seulement si $\text{conf}_G(l) \geq c^T_{min}$.

Seuillage sur le rang Un tel classifieur dédié aux liaisons de type T et paramétré par un seuil $r^T_{max} \in \mathbb{N}$, classifie une liaison l de type T dans un graphe moléculaire G comme formable si et seulement si le rang $r(l)$ de l dans la liste des liaisons de type T de G triées par ordre décroissant de confiance, est égal ou inférieur à r_{max}.

Seuillage sur le rang relatif Un tel classifieur dédié aux liaisons de type T et paramétré par un seuil $p^T_{max} \in [0;1]$, classifie une liaison l de type T dans un graphe moléculaire G comme formable si et seulement si le rang relatif de l est égal ou inférieur à p_{max},

où le rang relatif est la fraction du rang absolu $r(l)$ divisé par le nombre de liaisons de type T dans G.

Chacun de ces trois classifieurs correspond à une intuition différente : l'intuition sous-jacente au premier classifieur est que la formabilité d'une liaison l dépend seulement de sa situation dans G (i.e. de ses environnements), indépendamment des niveaux de formabilité des autres liaisons de G. Le second classifieur suppose que le nombre de liaisons formables d'une molécule est un nombre globalement constant et indépendant de la taille de la molécule. Enfin le troisième classifieur suppose que le nombre de liaisons formables est en moyenne proportionnel au nombre de liaisons de la molécule G. Intuitivement, le premier classifieur semble avoir plus de sens que le troisième qui lui-même semble plus pertinent que le second. Cependant dans le cas d'exemples étiquetés automatiquement, les liaisons étiquetées comme formables dans un graphe moléculaire G correspondent aux liaisons formées par une réaction donnée, synthétisant la molécule G. Comme une réaction forme un faible nombre de liaisons (2 en moyenne et rarement plus de 4), le nombre d'exemples de liaisons étiquetées comme formables tend à être constant plutôt que proportionnel à la taille de G. Du fait de cet artefact, on peut s'attendre à ce que le second classifieur donne de meilleurs résultats que le troisième.

7.3 Tests

Cette section présente les nombreux tests qui ont pu être réalisés et l'analyse quantitative comme qualitative des résultats qu'ils ont produits. La section 7.3.1 présente d'abord le jeu d'exemples sur lequel s'appuient tous les tests présentés. La section 7.3.2 présente ensuite la méthodologie générale utilisée par chacun des tests unitaires. Enfin la section 7.3.3 étudie l'influence des différents paramètres de test sur la qualité des résultats.

7.3.1 Sélection des données

Les données ont été extraites de deux BdR ChemInform et Reflib[57]. Ces bases de données ont été choisies pour regrouper des réactions collectées sur une longue période et pour offrir une grande diversité structurelle, en particulier en terme de topologie des substrats, de stéréo-chimie et d'effets de substituants. De nombreuses classes de composés organiques, de catalyseurs et de conditions expérimentales y sont également représentées. Afin de produire un ensemble de quelques milliers de réactions, différents filtres de sélection ont consécutivement été appliqués à ChemInform et Reflib pour aboutir à un ensemble de 7537 réactions. La liste de ces filtres consécutifs et de leur effet sur le nombre de réactions restantes est résumée sur la figure 7.12. La première étape consiste à éliminer toutes les réactions qui ne forment pas de liaison carbone carbone. La formation de ce type de liaisons est en effet un problème à la fois plus important et plus difficile de la synthèse organique que celui de la formation des liaisons impliquant des hétéroatomes. Un filtre sur le rendement exigé supérieur à 90 % permet ensuite de réduire le nombre de réactions de façon importante. Seules les réactions élémentaires, c'est-à-dire mono-étapes et mono-produits, sont conservées. Enfin les réactions comportant des atomes métalliques et plus généralement des éléments chimiques autres que les éléments les plus courants (B, C, N, O, F, Si, P, S, Cl, Br et I) sont éliminées car les molécules de ces réactions peuvent présenter des liaisons ioniques ou organo-métalliques que les graphes moléculaires ne peuvent modéliser fidèlement. Finalement,

[57]Ces deux bases de données sont des produits commerciaux de la société Symyx®/MDL®.

Opération de sélection	Nombre restant de réactions
ChemInform RX et Reference Library	1202174
Formation de liaisons C-C (tout type)	366264
Rendement $\geq$ 90 %	10774
Réaction mono-étape, mono-produit	10170
Filtrage à partir des éléments chimiques	8256
Prétraitement	7537

FIG. 7.12: Méthode de sélection des données

le prétraitement présenté au chapitre 4 conduit à un ensemble de 7537 produits de réactions saturés en hydrogène, dans lesquels les liaisons formées sont repérées par un étiquetage adapté.

Toutes les liaisons ne présentent pas le même intérêt pour le chimiste. C'est pourquoi les liaisons sont réparties en trois catégories : les liaisons carbone-carbone, abrégées par liaisons CC, sont les liaisons dont la formation est l'objectif le plus important en synthèse organique car ce sont elles qui forment le squelette des molécules cibles. Les liaisons incidentes à un atome d'hydrogène sont les plus nombreuses et sont celles qui sont obtenues par saturation des graphes moléculaires en hydrogène. Enfin les liaisons restantes sont les liaisons hétéroatomiques qui comprennent les liaisons de type carbone-hétéroatome et de type hétéroatome-hétéroatome. Le tableau de la figure 7.13 décrit la distribution statistique des liaisons dans le jeu d'exemples sélectionnés pour chaque type et catégorie de liaisons. Le tableau fournit pour chaque type et catégorie de liaisons, le nombre N de ces liaisons, leur fréquence relative N/N_{tot} parmi le nombre total $N_{tot} = 312018$ de liaisons, la proportion N_c/N des liaisons formées dans leur catégorie et la fréquence relative N_c/N_{tot} des liaisons formées. Le

Catégorie \ Type de liaison		Simple	Double	Triple	Aromatique	Total
Liaisons incidentes à un atome d'hydrogène	N	145823				145823
	N/N_{tot}	47 %				47 %
	N_c/N	2,6 %				2,6 %
	N_c/N_{tot}	1,2 %				1,2 %
Liaisons carbone-carbone	N	54146	4226	349	59209	117930
	N/N_{tot}	17 %	1,3 %	0,11 %	19 %	38 %
	N_c/N	12,5 %	20,4 %	2,3 %	1,6 %	7,3 %
	N_c/N_{tot}	2,2 %	0,3 %	0,003%	0,3 %	2,8 %
Liaisons hétéro-atomiques	N	32696	9455	732	5382	48265
	N/N_{tot}	10 %	3,0 %	2,3 %	1,7 %	15,5 %
	N_c/N	5,6 %	2,2 %	0,8 %	10 %	5,3 %
	N_c/N_{tot}	0,59 %	0,07 %	0,002%	0,17 %	0,8 %
Toutes les liaisons	N	232665	13681	1081	64591	312018
	N/N_{tot}	75 %	4,4 %	0,3 %	21 %	100 %
	N_c/N	5,3 %	7,8 %	1,3 %	2,3 %	4,8 %
	N_c/N_{tot}	3,95 %	0,34 %	0,004%	0,48 %	4,8 %

FIG. 7.13: Distribution statistique des liaisons dans les exemples

tableau suscite plusieurs commentaires. D'abord les classes de liaisons formées/non formées

sont comme prévu très déséquilibrées : sur les 312018 liaisons que comptent les exemples, moins de 5 % sont formées. Ensuite la catégorie des liaisons la plus représentée n'est pas celle des liaisons CC mais celles des liaisons incidentes à un atome d'hydrogène (47 % contre 38 %). Par ailleurs, alors que les chimistes perçoivent les hétéroatomes comme en moyenne plus réactifs (i.e. plus instables chimiquement et donc plus prompts à réagir en brisant d'anciennes liaisons incidentes pour en former de nouvelles) que les atomes de carbone ou d'hydrogène, la catégorie des liaisons les plus souvent formées n'est pas celle des liaisons hétéro-atomiques mais celles des liaisons carbone-carbone (12,5 % contre 10 % pour les liaisons simples et 20,4 % contre 3 % pour les liaisons doubles). Ce biais peut s'expliquer par deux facteurs cumulatifs : d'une part les réactions recensées dans les bases de données répondent à des problèmes de synthèse dont le but est de former le squelette carboné de la molécule cible, ce qui a pour conséquence d'amplifier le nombre des liaisons CC formées. D'autre part la méthode de sélection des données privilégie explicitement la formation de liaisons CC. La proportion inverse est toutefois observée concernant les liaisons aromatiques (les liaisons CC aromatiques comptent seulement 1,6 % de liaisons formées là où les liaisons hétéro-atomiques en comptent 10 %). Cette particularité s'explique par le fait que les cycles benzéniques sont peu réactifs et le problème de leur synthèse est généralement évité en recherchant ces cycles benzéniques directement dans des produits de départ. Enfin la formation des liaisons triples n'est pas suffisamment représentée pour pouvoir espérer une estimation fiable de leur formabilité : seulement 11 liaisons triples sont formées, soit 1,3 % des 1081 liaisons triples qui ne représentent que 0,3 % du total des liaisons.

7.3.2 Méthode de test

De nombreux tests ont été réalisés afin de mesurer l'effet des différents facteurs influant sur les résultats de `GemsBond`, tels que les différents types de liaisons, les différents classifieurs binaires, les différentes valeurs des paramètres d_{min} et f_{min} ou encore les différentes modélisations des graphes moléculaires. Le mode opératoire de chaque test unitaire, représenté sur la figure 7.14, est toujours le même. Les tests réalisés reposent sur le principe de validation croisée. Les 7537 produits de réactions sont divisés en 75 sous-ensembles de 100 molécules et un 76$^{\text{ème}}$ sous-ensemble de 37 réactions. Les confiances des liaisons contenues dans chacun des 76 sous-ensembles sont évaluées par `GemsBond` à partir des exemples constitués des 75 sous-ensembles restants. Chaque test unitaire soumet à `GemsBond` un ensemble $\mathcal{I}$ de graphe moléculaires en entrée dont le sous-ensemble F_e des liaisons formables est connu (en tant que liaisons formées par une réaction). En comparant le sous-ensemble F_s des liaisons classifiées comme étant formables par le classifieur binaire au sous-ensemble F_e, les liaisons de $\mathcal{I}$ se répartissent en quatre catégories : les *vrais positifs* (i.e. $F_s \cap F_e$), les *vrais négatifs* (i.e. $F_s^C \cap F_e^C$)[58], les *faux positifs* (i.e. $F_s \cap F_e^C$) et les *faux négatifs* (i.e. $F_s^C \cap F_e$). Ainsi sur l'exemple de la figure 7.9(a) dont les seules liaisons étiquetées comme formées sont les liaisons l_1 et l_2 (i.e. $F_e = \{l_1; l_2\}$), un seuillage sur la confiance avec un seuil c_{min} de 0,5 fait de l_1 et l_2 des vrais positifs, de l_3 un faux positif (puisque $\text{conf}(l_3) = 0,57$) et de l_4 un vrai négatif (puisque $\text{conf}(l_3) = 0,49$). Les nombres de vrais positifs, vrais négatifs, faux positifs et faux négatifs dans $\mathcal{I}$ notés VP, VN, FP et FN permettent de définir la *spécificité* SP, la *sensitivité* SE

[58] E^C désigne l'ensemble complémentaire de E dans l'ensemble des liaisons de $\mathcal{I}$.

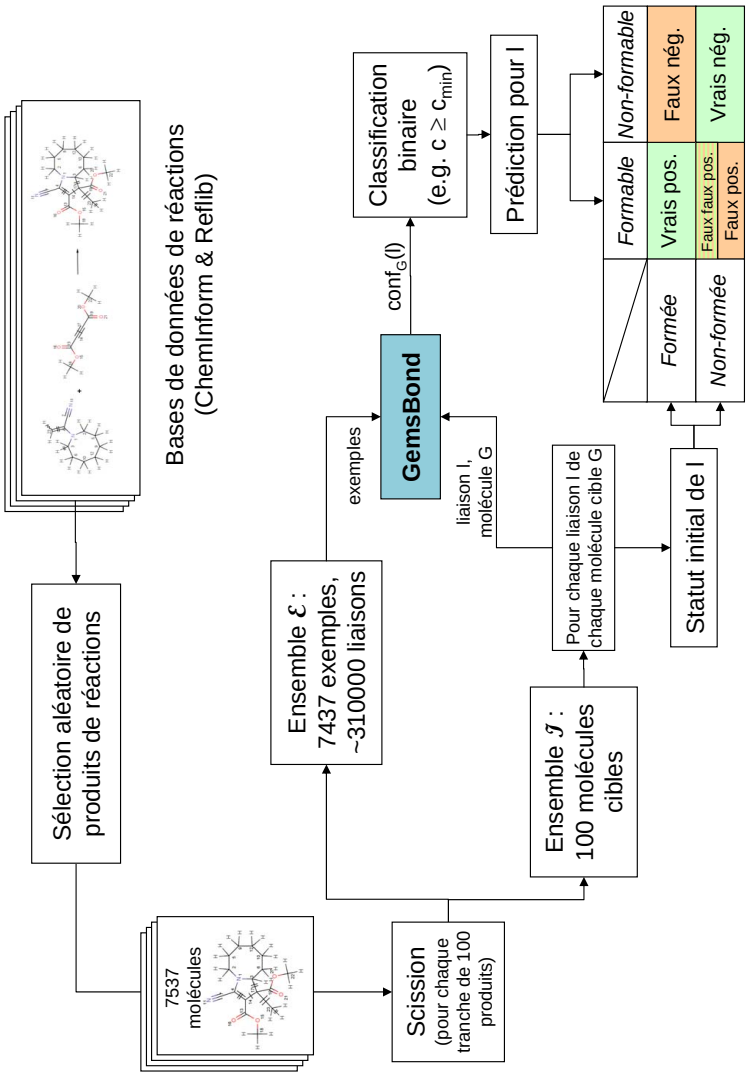


FIG. 7.14: Mode opératoire d'un test unitaire

et l'*exactitude* EXA[59] d'un classifieur :

$$SP = \frac{VN}{VN + FP} \tag{7.1}$$

$$SE = \frac{VP}{VP + FN} \tag{7.2}$$

$$EXA = \frac{VP + VN}{VP + FP + VN + FN} \tag{7.3}$$

Un classifieur est d'autant plus performant que les trois mesures précédentes s'approchent de la valeur maximale de 1. L'exactitude est la fraction du nombre d'exemples correctement classés sur le nombre total d'exemples traités, c'est-à-dire le complément à 1 du taux d'erreur. L'exactitude est toutefois une mesure biaisée par le déséquilibre entre les représentations des exemples positifs et négatifs, qui dans le cas présent, est particulièrement marqué (i.e. seulement 5 % de liaisons sont formées). Ce déséquilibre conduit au « paradoxe de l'exactitude » [60] bien connu dans le milieu de l'apprentissage automatique : le classifieur qui classe systématiquement toute liaison comme étant non formable obtient une exactitude égale à la proportion de liaisons non formées, soit un score excellent de 95 % alors que ce classifieur est aveugle et n'a par conséquent aucun pouvoir prédictif ! Pour cette raison, une mesure de l'exactitude ne saurait qualifier précisément les performances d'un classifieur et doit toujours être rapportée au test réalisé.

Contrairement à l'exactitude, la spécificité et la sensitivité permettent d'aboutir à des mesures non biaisées. La spécificité et la sensitivité sont en effet les proportions d'exemples correctement prédits parmi les exemples respectivement négatifs et positifs. De par leurs définitions, la spécificité et la sensitivité sont des mesures antinomiques : l'augmentation de l'une des deux mesures se fait au prix de la diminution de l'autre. En effet pour obtenir une spécificité élevée, c'est-à-dire peu de faux positifs, il suffit de ne classer positifs que les quelques exemples qui le sont de manière évidente, mais ce faisant, beaucoup d'exemples positifs sont classés négativement et la sensitivité s'en trouve amoindrie. Ce phénomène est particulièrement visible si on considère le classifieur aléatoire dont la probabilité de classer un exemple comme positif est de $\alpha \in [0; 1]$. La sensitivité et la spécificité apparaissent alors complémentaires, puisque respectivement égales à α et $1-\alpha$. Un classifieur sera donc d'autant plus performant qu'il sera capable de concilier une sensitivité et une spécificité élevées. La *courbe de ROC*[61] (Fawcett, 2006) permet de représenter graphiquement la qualité d'un tel classifieur. La construction de cette courbe de ROC dans le cas précis de la classification des liaisons formables, se fait de la manière suivante : étant donné un classifieur binaire pour un type de liaison T donné, tel qu'un seuillage fondé sur la confiance, le rang ou le rang relatif, le test réalisé pour une valeur précise du paramètre p associé au classifieur (i.e. c^T_{min}, r^T_{max} ou p^T_{max}), définit un couple $(SP(p), SE(p))$ de mesures spécificité - sensitivité. Ce couple peut se représenter par un point $(1 - SP(p), SE(p))$ dans le plan. La variation du paramètre p fait décrire à ce point une courbe croissante incluse dans le carré unité et appelée *courbe de ROC*. Cette courbe relie le point $(0, 0)$ (tous les exemples en entrée sont classés négativement lorsque $c^T_{min} > 1$ ou $r^T_{max} = p^T_{max} = 0$) au point $(1, 1)$ (tous les exemples en entrée sont classés positivement lorsque $c^T_{min} = 0$ ou $r^T_{max} = p^T_{max} = \infty$). Ainsi la courbe

[59] Le vocabulaire anglais en apprentissage statistique distingue les notions d'« accuracy » et de « precision ». Le terme *exactitude* est introduit ici pour faire référence à la notion d'« accuracy », le terme français précision étant naturellement associé à son homonyme anglais.

[60] Traduction littérale de « accuracy paradox ».

[61] Receiver Operating Characteristic

de ROC associée aux classifieurs aléatoires de paramètre α introduits précédemment décrit la première diagonale séparant les points $(0,0)$ et $(1,1)$. Le classifieur parfait qui ne commet aucune erreur correspond au point $(0,1)$. Le classifieur est donc d'autant meilleur que sa courbe de ROC s'approche de ce point $(0,1)$. L'*aire sous la courbe de ROC*, notée AUC[62] est donc une mesure naturelle pour qualifier les performances d'un classifieur :

$$AUC = \int_0^1 SE(SP^{-1}(x))dx \tag{7.4}$$

La courbe de ROC étant comprise dans le carré unitaire, l'AUC varie entre 0 et 1. En particulier, l'AUC de la famille des classifieurs aléatoires est de 0,5. Toutefois, si un classifieur C obtient un AUC inférieur à 0,5, le classifieur C' optant systématiquement pour le choix contraire à celui de C obtiendra un AUC supérieur à 0,5 (puisque $AUC(C') = 1 - AUC(C)$), de sorte qu'en réalité tout classifieur (ou son contraire) atteint au minimum un AUC de 0,5. En ce sens, la famille des classifieurs aléatoires obtient le plus faible AUC possible, ce qui est cohérent avec l'intuition, puisque les classifieurs aléatoires sont aveugles (i.e. ne tiennent pas compte de la description de l'objet à classifier).

L'AUC est un des critères les plus répandus pour apprécier la qualité d'un classifieur puisque cette mesure ne dépend pas du biais éventuel entre exemples positifs et négatifs. Une autre mesure très répandue est la *F-mesure* (van Rijsbergen, 1979) notée F_β qui est une moyenne harmonique pondérée de la *précision* notée $PREC$ et du *rappel*, qui n'est autre que la sensitivité SE définie précédemment :

$$PREC = \frac{VP}{VP + FP} \tag{7.5}$$

$$F_\beta = \frac{(1+\beta^2) \cdot PREC \cdot SE}{(\beta^2 \cdot PREC + SE)} \tag{7.6}$$

Le paramètre β permet de privilégier l'importance respective accordée à la précision et au rappel (puisque la précision et le rappel sont pondérés par des poids inverses β^{-1} et β). La F-mesure F_{mes} utilisée par défaut correspond au cas où précision et rappel revêtent la même importance, soit :

$$F_{mes} = F_1 = \frac{2 \cdot PREC \cdot SE}{(PREC + SE)} \tag{7.7}$$

À l'origine, la F-mesure a été introduite pour estimer les performances de méthodes de recherche d'information. En effet, la problématique abordée par la recherche d'information est légèrement différente de celle de la classification supervisée en apprentissage automatique puisque l'objectif de la recherche d'information est de retourner quelques documents pertinents répondant à une requête, quite à y inclure des faux positifs. Les vrais positifs ont donc bien plus de poids que les vrais négatifs. Cette asymétrie entre les rôles joués par les exemples positifs et négatifs se retrouve dans la F-mesure, puisque la formule de F_1 s'obtient simplement en substituant dans la formule de l'exactitude EXA les occurrences du nombre VN des vrais négatifs par celui VP des vrais positifs.

Dans la mesure où les tests réalisés reposent sur l'étiquetage automatique qui extrait les exemples de liaisons formables à partir des liaisons formées dans les BdR, il en résulte une sous-estimation du nombre de liaisons étiquetées comme formables. Les liaisons réellement

[62] Area Under the Curve

formables mais étiquetées comme non formables deviennent alors des « faux faux positifs » au lieu d'être de vrais positifs comme ils auraient dû. La conséquence est une surestimation du nombre FP de faux positifs qui entraîne une dégradation artificielle des performances, à travers la baisse de la spécificité et de la précision et donc de l'AUC et de la F-mesure. Les résultats affichés dans les sections suivantes constituent donc en réalité une borne inférieure pessimiste des véritables résultats obtenus non mesurables.

7.3.3 Résultats des tests

La suite de cette section décrit dans un premier temps les résultats obtenus pour un test dit de référence, puis dans un second temps, étudie l'influence des différentes variables du problème relativement au test de référence, que ce soit en terme de précision de la classification ou de temps de calcul.

Test de référence

Le test de référence utilise un seuillage sur la confiance, appliqué à un ensemble d'exemples saturés en atomes d'hydrogène et avec les paramètres d_{min} et f_{min} fixés respectivement à 5 et $5/7537 = 0,07\%$ (i.e. f_{min} correspond à un support minimal fixé à 5 exemples). La figure 7.15 affiche pour chaque type de liaisons, la distribution normalisée de la confiance au sein des liaisons formées et non formées. Excepté pour les liaisons triples qui ne sont pas

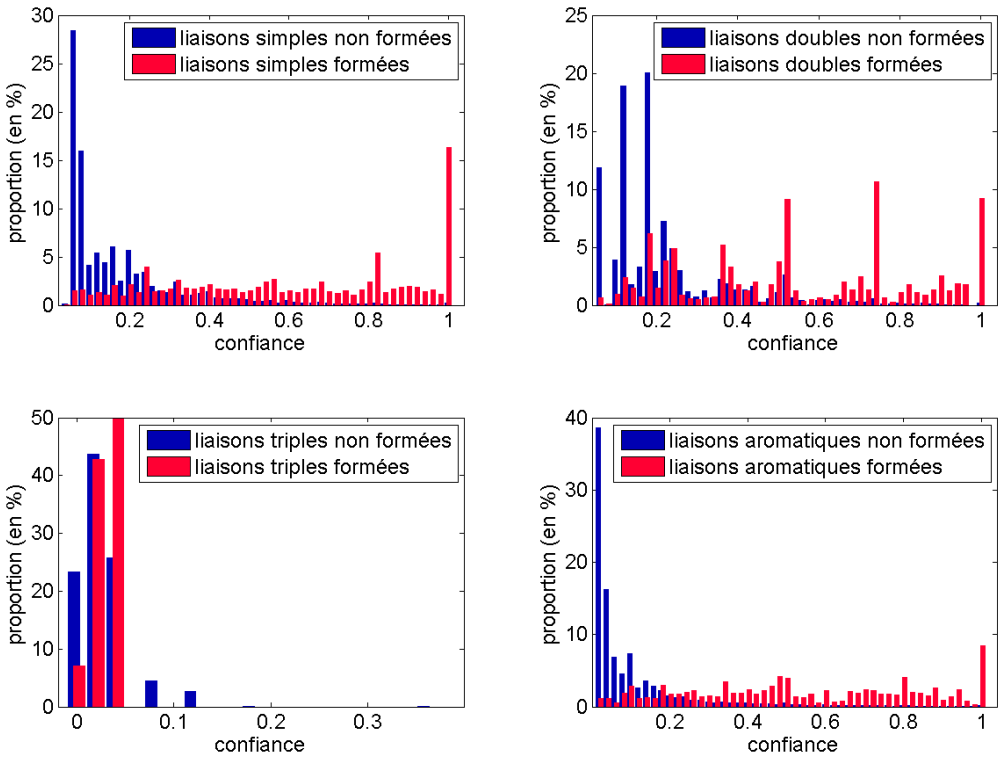


FIG. 7.15: Distributions normalisées des liaisons formées et non formées

suffisamment représentées dans les exemples (cf tableau de la figure 7.13) pour permettre une estimation fiable, `GemsBond` prédit clairement de plus grandes valeurs de confiance pour les liaisons formées que pour les liaisons non formées. La séparation grâce à un seuillage sur la confiance permet donc de séparer efficacement les deux distributions. Les courbes de ROC associées, représentées sur la figure 7.16, sont clairement au-dessus de la première diagonale, à l'exception de celle rattachée aux liaisons triples. Les mesures des différents indicateurs

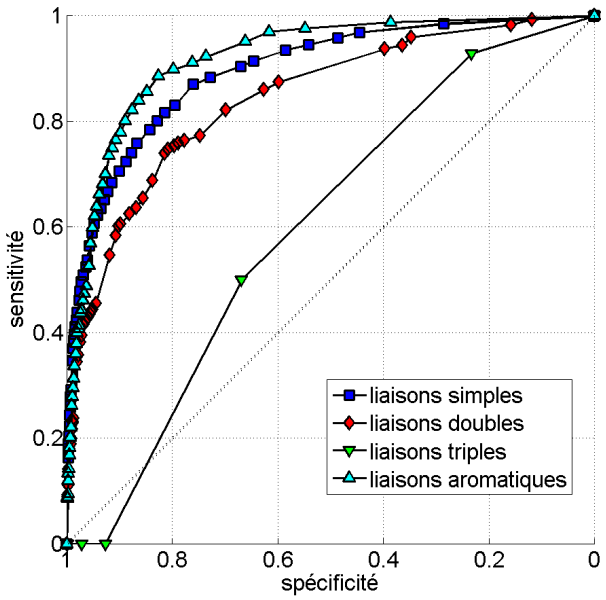


FIG. 7.16: Courbes de ROC pour un seuillage sur la confiance avec $d_{min} = 5$. Toutes les courbes apparaissent clairement au dessus de la première diagonale, à l'exception de celle rattachée aux liaisons triples.

introduits à la section 7.3.2 sont résumées dans le tableau de la figure 7.17.

Selon l'AUC, les liaisons dont la formabilité est la plus facile à prédire sont dans l'ordre les liaisons aromatiques, simples, puis doubles et enfin triples Selon la F-mesure, l'ordre est légèrement différent : viennent en tête les liaisons simples, puis doubles, aromatiques et enfin triples. Les mauvais résultats obtenus pour les liaisons triples étaient attendus dans la mesure où les données ne se focalisent pas sur les cas de formation de liaisons triples, particulièrement rares dans les BdR. À l'inverse, l'excellente performance obtenue dans la prédiction des liaisons aromatiques formables est une bonne surprise puisque l'étude de la formation des liaisons aromatiques n'était pas plus que celle des liaisons triples, l'objectif qui a guidé le processus de sélection des données (i.e. seulement 2,3 % des liaisons aromatiques sont créées). Mais contrairement aux liaisons triples, les liaisons aromatiques sont communes (21 % des liaisons) et présentent plus d'un millier d'exemples de liaisons formées.

Pour réaliser une comparaison qui soit indépendante du biais statistique introduit par la sur-représentation des liaisons non formées, la figure 7.17 présente le gain que fournit `GemsBond` relativement à la famille des classifieurs aléatoires pour les mesures que sont l'AUC, l'exactitude maximale et la F-mesure maximale. Le gain associé à une telle mesure M est

Catégorie de la liaison	Type de la liaison	AUC	Exactitude valeur maximale	Exactitude c_{min} associé	F-mesure valeur maximale	F-mesure c_{min} associé
Toutes les liaisons	Simple	0,90	0,96	0,82	0,50	0,64
	Double	0,84	0,93	0,90	0,46	0,66
	Triple	0,60	0,99	0,38	0,04	0,04
	Aromatique	0,92	0,97	0,94	0,36	0,66
	Tout type	0,90	0,96	–	0,47	–
Liaisons carbone-carbone	Simple	0,84	0,89	0,86	0,51	0,64
	Double	0,77	0,83	0,74	0,52	0,64
	Triple	0,50	0,98	0,38	0,05	0,04
	Aromatique	0,92	0,98	0,94	0,35	0,66
	Tout type	0,88	0,94	–	0,43	–

FIG. 7.17: Résultats obtenus pour le test de référence

défini comme le rapport :

$$G_M = \frac{M_{gemsbond} - M_{aleatoire}}{M_{optimal} - M_{aleatoire}} = \frac{M_{gemsbond} - M_{aleatoire}}{1 - M_{aleatoire}}$$

où $M_{gemsbond}$, $M_{optimal}$ et $M_{aleatoire}$ sont les valeurs de M fournies respectivement par `GemsBond` (pour la valeur c_{min} qui rend maximal $M_{gemsbond}$), par le classifieur idéal (qui vaut 1 pour l'AUC, la F-mesure ou l'exactitude) et par le classifieur aléatoire (pour la valeur α qui rend maximal $M_{aleatoire}$). Dans le cas du classifieur aléatoire, si $r_c = N_c/N$ désigne la proportion des liaisons formées pour le type et la catégorie de liaisons considérées (cf le tableau de la figure 7.13 pour les valeurs numériques), l'exactitude est maximale et égale à $1 - r_c$ pour $\alpha = 0$ et la F-mesure est maximale et égale à $\frac{2}{1+r_c^{-1}}$ pour $\alpha = 1$ tandis que l'AUC est de 0,5. Le tableau de la figure 7.18 fait apparaître des gains faibles et aléatoires sur l'exacti-

Catégorie de la liaison	Type de la liaison	Gain d'AUC	Gain de l'exactitude maximale	Gain de la F-mesure maximale
Toutes les liaisons	Simple	80 %	25 %	44 %
	Double	68 %	10 %	37 %
	Triple	20 %	23 %	1 %
	Aromatique	84 %	-30%	33 %
	Tout type	80 %	17 %	42 %
Liaisons carbone-carbone	Simple	68 %	12 %	37 %
	Double	54 %	17 %	27 %
	Triple	0 %	13 %	1 %
	Aromatique	84 %	-25%	33 %
	Tout type	76 %	18 %	34 %

FIG. 7.18: Gains relatifs vis-à-vis de la famille des classifieurs aléatoires

tude alors que dans le même temps des gains significatifs et plus réguliers sont observés pour

l'AUC et la F-mesure. Le « paradoxe de l'exactitude » apparaît de manière criante dans le cas des liaisons aromatiques puisque `GemsBond` obtient pour ce type de liaisons à la fois le meilleur gain de prédiction (+84% sur l'*AUC*) et à la fois le plus mauvais gain sur l'exactitude (−30%), paradoxalement bien plus faible que celle du classifieur aléatoire pourtant aveugle. Ces observations confortent l'idée déjà mentionnée selon laquelle l'AUC et la F-mesure sont des mesures plus objectives que l'exactitude pour évaluer la qualité d'un classifieur.

Influence du type de classifieur binaire

La figure 7.19 compare les performances des trois classifieurs binaires proposés à la section 7.2.4 (i.e. seuillages sur la confiance, le rang et le rang relatif), toutes choses égales par ailleurs. Le seuillage sur la confiance constitue le classifieur le plus précis. Cette observa-

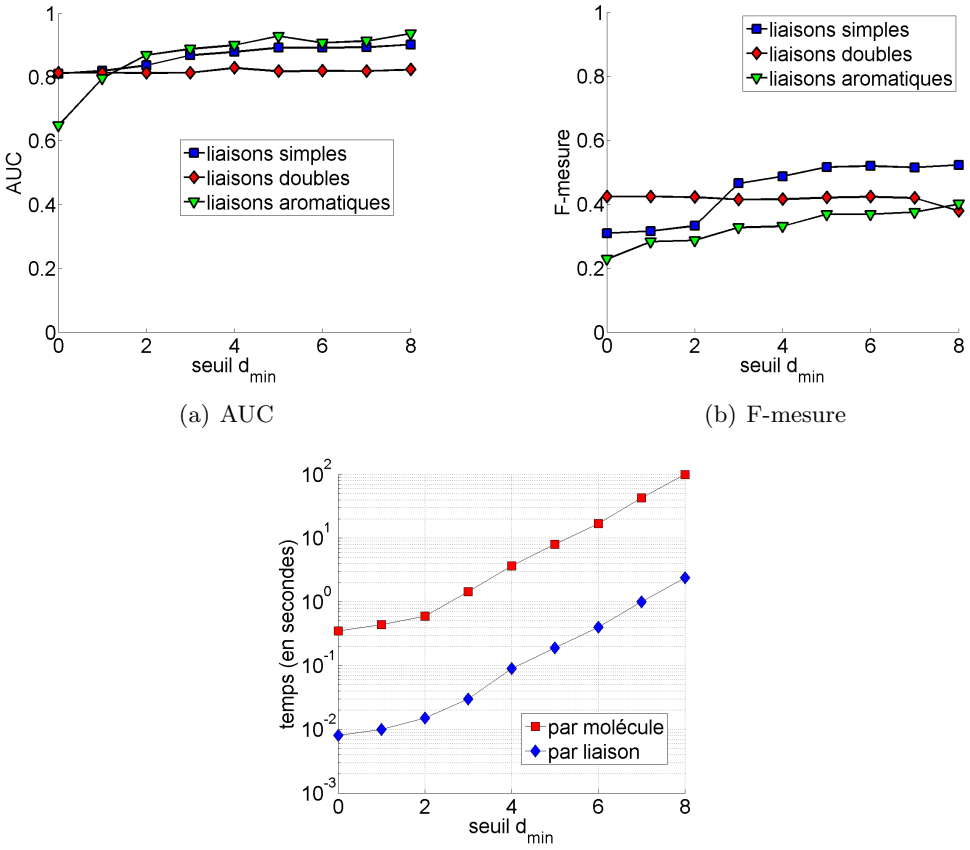
	Seuillage sur la confiance			Seuillage sur le rang			Seuillage sur le rang relatif		
		F-mesure			F-mesure			F-mesure	
Type de liaison	AUC	valeur max.	c_{min} assoc.	AUC	valeur max.	r_{max} assoc.	AUC	valeur max.	p_{max} assoc.
Simple	0,90	0,50	0,64	0,88	0,47	3	0,88	0,45	0,14
Double	0,84	0,46	0,66	0,69	0,22	2	0,69	0,22	0,12
Triple	0,60	0,37	0,04	0,47	0,03	3	0,48	0,03	0,56
Aromatique	0,92	0,36	0,66	0,72	0,10	3	0,73	0,10	0,20
Tout type	0,90	0,47	NR	0,84	0,38	NR	0,84	0,37	NR

FIG. 7.19: AUC et F-mesure des différents classifieurs

tion confirme que la formabilité d'une liaison dépend principalement de son environnement, indépendamment de la formabilité des autres liaisons dans la molécule. Plus étonnant est l'absence de différence entre l'efficacité du seuillage sur le rang ou sur le rang relatif. Le fait que les exemples de liaisons formables soient des liaisons formées extraites des BdR donnait pourtant un avantage théorique au classifieur fondé sur le rang.

Influence de la profondeur d_{min} minimale de recherche

`GemsBond` fouille tous les environnements de la liaison cible dont la taille, définie comme le nombre total d'atomes et de liaisons, est inférieur ou égal au seuil d_{min}. Plus grand est d_{min}, plus long sera le temps de calcul mais plus grandes seront les chances de trouver l'environnement E_{max} de confiance maximale. Ce dernier point peut influer sur la qualité de la classification. La figure 7.20 met en évidence l'influence du paramètre d_{min} sur le temps de calcul, l'AUC et la F-mesure. Sans surprise, le temps de calcul croit exponentiellement avec le seuil d_{min} puisque le nombre d'environnements qu'une liaison a et dont la taille est inférieure à un entier d_{min}, croit lui aussi exponentiellement avec d_{min}. L'AUC tend aussi à croître avec d_{min} mais très lentement et à des degrés variables : l'augmentation du paramètre d_{min} bénéficie davantage aux liaisons aromatiques et dans une moindre mesure aux liaisons simples, mais absolument pas aux liaisons doubles pour lesquelles une recherche gloutonne suffit. Un bon compromis entre temps de calcul et précision de la classification semble être atteint pour $d_{min} = 4$, où les AUCs des quatre types de liaisons ont presque atteint leur valeur optimale asymptotique alors que le temps nécessaire au traitement de toutes les liaisons d'une molécule cible n'excède pas quelques secondes en moyenne.



(a) AUC

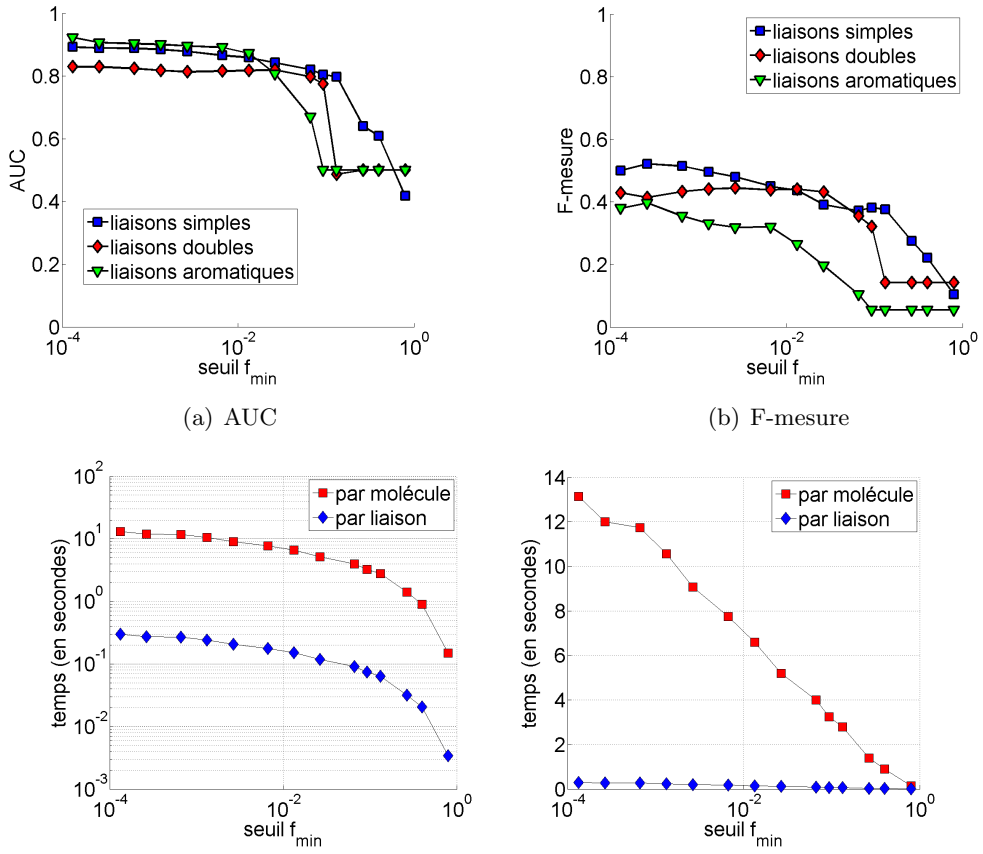
(b) F-mesure

(c) Temps de calcul

FIG. 7.20: Influence du paramètre d_{min}

Influence du seuil f_{min} de fréquence minimale

La figure 7.21 met en évidence l'influence du seuil f_{min} de fréquence minimale sur le temps de calcul et la précision de la classification. Le temps de calcul est une fonction décroissante



(a) AUC

(b) F-mesure

(c) Temps de calcul (loglog)

(d) Temps de calcul (logx)

FIG. 7.21: Influence du paramètre f_{min}

de f_{min} puisque le nombre d'environnements traités diminue lorsque f_{min} croît. Le temps de calcul apparaît être une fonction du type $t = a + b \cdot \log(1/f_{min})$ (cf figure 7.21(d)), suggérant que chaque extension de l'environnement courant fait décroître sa fréquence par un facteur constant. L'AUC et la F-mesure semblent peu dépendre du paramètre f_{min} tant que la valeur de ce dernier reste suffisamment faible. Lorsque la valeur de f_{min} devient élevée, les valeurs de l'AUC et de la F-mesure baissent brutalement. Cette chute peut s'expliquer par l'élagage de certains environnements explicatifs cruciaux, lorsque la fréquence de ces derniers devient inférieure à f_{min}. Ainsi lorsque f_{min} tend vers 1, l'AUC et la F-mesure tendent vers les valeurs théoriques de la famille des classifieurs aléatoires.

Influence des atomes d'hydrogène

Le prétraitement sature en atomes d'hydrogène les graphes moléculaires des exemples et des molécules en entrée puisque ces atomes d'hydrogène peuvent jouer un rôle dans le caractère formable des liaisons et donc apparaître dans les environnements explicatifs. La

figure 7.22 met en évidence l'influence que les atomes d'hydrogène ont sur l'*AUC* et la F-mesure dans la classification des liaisons carbone-carbone. Comme prévu, la présence des

Type de liaison	Avec atomes d'hydrogène		Sans atomes d'hydrogène	
	AUC	F-mesure	AUC	F-mesure
Simple	0,84	0,50	0,81	0,47
Double	0,77	0,52	0,77	0,52
Triple	0,50	0,05	0,50	0,04
Aromatique	0,92	0,35	0,93	0,32
Tout type	0,88	0,43	0,87	0,40

FIG. 7.22: Influence de la saturation en atomes d'hydrogène sur la classification des liaisons carbone-carbone

atomes d'hydrogène améliore les performances du classifieur, même si cette influence est globalement plutôt faible. Cette influence est confirmée par la présence d'atomes d'hydrogène dans les explications, y compris dans les explications de grande confiance. Cependant cette amélioration a un coût comme le montre la figure 7.23, puisque les atomes d'hydrogène sont très nombreux et augmentent ainsi de manière importante le nombre d'environnements à fouiller lorsque le paramètre d_{min} augmente.

Valeur de d_{min}	Temps de calcul par molécule (en secondes)	
	Sans hydrogène	Avec hydrogène
0	0,4	0,4
3	0,7	1,4
5	1,7	7,2
7	4,7	49

FIG. 7.23: Influence de la saturation en atomes d'hydrogène sur le temps de calcul pour différentes valeurs de d_{min}

7.3.4 Comparaison avec l'état de l'art

La section 7.1.1 a montré que le problème de l'évaluation de la formabilité des liaisons a déjà été abordé à travers l'identification des liaisons stratégiques et que pour y répondre, différentes méthodes ont été proposées. La seule de ces méthodes qui soit fondée sur l'apprentissage à partir d'exemples est l'algorithme `CNN` (Régin *et al.*, 1995; Régin, 1995). `CNN` et `GemsBond` sont donc très proches et il est naturel de les comparer. L'idée de `CNN` (pour Conceptual Nearest Neighbour) est d'adapter le principe de classification supervisée des méthodes de voisinage au cas où les exemples ne sont plus des vecteurs mais des objets symboliques disposant d'un opérateur d'intersection (en l'occurrence l'ensemble des sous-graphes communs maximaux ou SGCM dans le cas des graphes, cf section 2.3.1). En pratique `CNN` est dans son principe un algorithme d'apprentissage inductif de type bottom-up qui détermine des règles

de classification en généralisant des exemples. Ainsi comme `GemsBond`, `CNN` prend en entrée un objet symbolique e (i.e. un motif d'attributs, un graphe...) dont on cherche à déterminer la classe C^+ ou C^- et des ensembles $\mathcal{E}^+$ et $\mathcal{E}^-$ d'exemples positifs et négatifs connus pour appartenir respectivement à C^+ ou C^-. La première étape consiste à calculer les motifs M, appelés *ressemblances symboliques*, qui sont à l'intersection de e et des exemples de $\mathcal{E}^+$ et $\mathcal{E}^-$. Ces ressemblances peuvent être vues comme des règles R de classification du type $M \rightarrow C$ où C vaut C^+ ou C^- selon que M couvre plus d'exemples positifs ou négatifs. Un score, appelé « valeur », associé à chaque règle R est ensuite calculé pour estimer le pouvoir discriminant de R. Seules les règles les plus discriminantes (i.e. qui couvrent de nombreux exemples positifs et peu d'exemples négatifs ou l'inverse) sont ensuite retenues. Ces ressemblances sont alors réparties dans une hiérarchie, selon la relation de subsomption (relation de sous-graphe isomorphe dans le cas des graphes) induite par le relation de couverture des exemples par les ressemblances. Les règles qui sont plus générales ou plus spécifiques qu'une règle de score plus élevé sont ensuite écartées. Les règles restantes couvrent alors un ensemble d'exemples positifs et négatifs qui se prononcent quant à la classe de e. Le vote majoritaire détermine la classe de l'objet.

`CNN` a donné d'excellents résultats dans la prédiction des liaisons formables sur un petit ensemble de 694 liaisons simples CC contenues dans 75 molécules. Les informations rapportées dans Régin (1995) et obtenus par un test de type Jack Knife (i.e. un test de validation croisé réalisé sur une molécule en utilisant comme exemples les 74 restantes) ont permis de reconstruire les résultats, notamment en terme d'AUC et de F-mesure, résumés dans le tableau de la figure 7.24. L'AUCb (pour Balanced AUC) est la moyenne de la sensitivité et

Nombre de molécules	75	Temps de calcul (heures)	72
Nombre de liaisons formées	91	Nombre de liaisons non formées	603
Faux négatifs	9	Faux positifs	18
Spécificité	90 %	Sensitivité	97 %
Précision	82 %	Exactitude	96 %
AUCb	93 %	F-mesure	88 %

FIG. 7.24: Résultats rapportés dans Régin (1995) concernant `CNN`

de la spécificité. L'appellation AUCb vient du fait que cette mesure non biaisée correspond à l'aire sous la courbe de ROC associé à un classifieur dont on ne dispose que d'un seul point de mesure (SP, SE). Ces résultats apparaissent meilleurs que ceux obtenus lors des tests de `GemsBond`. Cette comparaison a toutefois une portée limitée. D'abord il est difficile de tirer des conclusions définitives à partir de résultats conduits sur un seul test et sur un jeu d'exemples aussi réduit. Ensuite et surtout, les conditions dans lesquelles a été réalisé ce test diffèrent considérablement de celles utilisées pour évaluer `GemsBond` puisque des experts ont annoté manuellement les graphes moléculaires des exemples en y ajoutant des informations supplémentaires (comme l'effet mésomère ou l'effet inductif). Cette observation est étayée par le contraste qui existe entre les excellents résultats obtenus par `CNN`, et les règles de classification associées (fournies à la page 331 de Régin (1995)) qui sont similaires, en terme de taille et de complexité, aux environnements explicatifs renvoyés par `GemsBond`. Il est toutefois probable que `CNN` soit plus efficace que `GemsBond` sur un jeu réduit d'exemples dans la mesure où `CNN` repose sur un algorithme de classification sophistiqué. En particulier `CNN` classifie un exemple en tenant compte de plusieurs environnements discriminants, quand au contraire

GemsBond fait le choix de la simplicité, en prenant sa décision à partir d'un seul environnement E_{max}.

Toutefois l'avantage de la précision concédé à CNN se fait au prix de nombreux calculs particulièrement coûteux. CNN fait en effet une grande consommation du calcul NP-difficile des SGCM et du calcul NP-complet de détection de sous-graphe isomorphe, sur des graphes moléculaires qui comptent couramment entre 50 et 100 atomes et présentent de nombreux sous-graphes communs. Par ailleurs la complexité de CNN est au moins quadratique avec le nombre n d'exemples quand la complexité de GemsBond est en $O(n)$ (pour un nombre constant d'environnements fouillés). À l'inverse, GemsBond est capable de traiter de grands volumes de données et son approche top-down permet de converger rapidement vers l'environnement E_{max}. Si aucune comparaison directe entre les deux algorithmes n'a pu être réalisée (la mise en œuvre de CNN étant impossible), il est vraisemblable que GemsBond soit une solution plus rapide et « passant mieux à l'échelle » que ne l'est CNN. Cette hypothèse s'appuie sur les résultats rapportés dans Régin (1995). L'évaluation de la formabilité de 694 liaisons à partir de 75 exemples de graphes moléculaires a en effet nécessité 72 heures de temps de calcul sur une machine SPARC 2. La classification d'une liaison a donc pris pas moins de 6 minutes en moyenne et ce à partir de 74 exemples. En comparaison le test de référence de GemsBond (pour $d_{min} = 5$) a nécessité seulement 0,2 secondes en moyenne sur un processeur Opteron, en fouillant non pas 75 mais 7500 exemples. Le fait que CNN a été écrit dans le langage Fortran sur une architecture matérielle qui date du début des années 90, peut expliquer en partie un tel écart (i.e. un facteur cumulé de 1800 sur le temps de calcul et de 100 sur les données), mais pas nécessairement la totalité.

Par ailleurs les résultats obtenus pour le test de référence ont été expertisés par Gilles Niel, expert en synthèse organique et chargé de recherche au CNRS au sein de l'Institut Charles Gerhardt de Montpellier. Des exemples d'environnements structuraux produits par GemsBond tels que ceux qui lui ont été remis sont donnés en annexe C.2. Chaque liaison de chaque molécule est en outre associée à son environnement explicatif ainsi qu'à sa confiance et fréquence. Ces travaux d'analyse ont abouti à la rédaction d'un rapport d'expertise rédigé en anglais et fourni en annexe D. Sa conclusion est la suivante :

> En résumé, GemsBond reconnaît dans 86 % des molécules étudiées au moins une liaison formable. Il propose une explication qui contient, dans de nombreux cas, le rétron d'une méthode de synthèse connue ainsi que la ou les fonctions constituant l'environnement nécessaire pour que cette liaison soit formée.

7.4 Conclusions

En résumé, ce chapitre a introduit la notion de formabilité des liaisons d'une molécule et s'est intéressé au problème de la quantification du degré de formabilité des liaisons. Contrairement aux problèmes traités jusqu'à présent dans les chapitres précédents, le problème de l'estimation de la formabilité d'une liaison se rattache plus au domaine de la synthèse ciblée qu'à celui de la méthodologie de synthèse : la question n'est plus d'analyser le contenu des BdR mais d'aider à synthétiser une molécule cible. Ainsi le degré de formabilité des liaisons d'une molécule est une information intéressante qui peut être intégrée dans les systèmes d'aide à la rétrosynthèse au niveau de l'analyse stratégique de la molécule cible, ou dans la procédure de « scoring » d'un système de criblage virtuel, afin de tenir compte de l'accessibilité synthétique des molécules testées. L'estimation de la formabilité des liaisons conduit à définir formellement les problèmes dits du classement et de la découverte des liaisons formables, qui

peuvent être abstraits en un problème de classification supervisée de sommets ou d'arêtes fondée sur leur environnement. L'approche préconisée pour résoudre ce problème est celle de la recherche heuristique « transductive » déjà introduite au chapitre 6, même si l'utilisation de cette approche se fait dans un contexte différent : l'extraction du schéma CMS sous-jacent à une réaction présentée au chapitre 6 se rattache en effet à un problème de classification non supervisée de graphes alors que la découverte des liaisons formables est un problème de classification supervisée des sommets ou des arêtes d'un graphe. L'algorithme développé, baptisé `GemsBond` , associe à chaque environnement de la liaison considérée, un degré de confiance en l'hypothèse selon laquelle la liaison est formable étant donné sa position dans un graphe moléculaire. `GemsBond` recourt alors à une heuristique adaptée à l'application considérée, pour classifier rapidement la liaison grâce à un algorithme glouton. Cette heuristique consiste à fonder la classification uniquement sur l'environnement qui présente le degré de confiance maximal vis-à-vis de l'hypothèse.

Les nombreux tests réalisés ont validé empiriquement la pertinence de cette heuristique et démontrent plus généralement que `GemsBond` est une méthode précise, rapide et « passant à l'échelle » pour découvrir les liaisons formables d'une molécule. Par rapport aux méthodes d'apprentissage inductif réalisées antérieurement, l'approche de `GemsBond` semble conduire à une précision légèrement plus faible mais à une rapidité et un passage à l'échelle bien plus élevés. De nombreuses perspectives de développement émergent de ces résultats. Du point de vue informatique, l'algorithme de recherche peut être amélioré pour augmenter la précision sans que cela se fasse trop au détriment de l'efficacité et du passage à l'échelle. En particulier, l'heuristique peut être améliorée pour prendre en compte plusieurs environnements, éventuellement défavorables à l'hypothèse de formabilité d'une liaison. Du point de vue de la chémoinformatique, une étude doit être entreprise pour comprendre comment la procédure de sélection des exemples peut influencer les performances de la classification. Même si `GemsBond` peut traiter des milliers d'exemples de molécules, il est tout aussi évident que `GemsBond` n'est pas capable de fouiller les millions de réactions disponibles dans les BdR. Il est donc essentiel de définir une méthodologie capable de construire un ensemble restreint de quelques milliers de réactions qui soit aussi représentatif que possible de la variété des méthodes de synthèse présentes dans les BdR. En ce sens, les travaux présentés au chapitre 6 peuvent contribuer à améliorer la prédiction des liaisons formables. Enfin l'intégration de connaissances du domaine dans la modélisation des données, comme par exemple l'identification des groupes fonctionnels dans les graphes moléculaires, doit pouvoir améliorer les résultats produits par `GemsBond`. Non seulement cela peut augmenter la précision de la méthode mais aussi cela peut fournir aux chimistes des environnements explicatifs plus proches de ceux qu'ils auraient spontanément proposés. Enfin `GemsBond` pourrait à l'avenir être appliqué à d'autres problèmes de classification d'atomes ou de liaisons, voire à d'autres domaines d'application que la chimie, en adaptant l'heuristique au problème de fouille de données traité.

Chapitre 8

Bilan et perspectives

La progression logique des quatre derniers chapitres correspond à peu de chose près à leur déroulement chronologique[63]. Cette coïncidence n'est pas fortuite : tout travail de recherche semble devoir avancer au gré de déceptions, de remises en questions mais aussi de succès qui mis bout-à-bout, forment un développement cohérent. Ce chapitre retrace ainsi l'enchaînement de ces idées et résultats qui ont jalonné cette histoire, avant de conclure sur les principales contributions et perspectives de ce travail.

Histoire passée ...

Ainsi ce récit débute naïvement en apparence au chapitre 4 par la recherche des schémas de réactions fréquents dans un ensemble d'équations de réactions. Je dis « naïvement en apparence » car les motifs et les règles d'associations fréquents – et plus particulièrement les motifs de graphes fréquents – sont connus pour être si nombreux que leur analyse est impossible sans autre forme de filtrage. La résolution de ce problème n'était donc pas véritablement une fin en soi mais plutôt le point de départ de l'exploration des BdR et d'une certaine manière aussi, un passage obligé pour pouvoir justifier objectivement de l'utilité des motifs les plus informatifs. Ce problème devait aussi me servir à me rapprocher de l'objectif – un peu trop ambitieux, il faut le reconnaître – que j'avais en ligne de mire : partir à la conquête des schémas caractéristiques des méthodes de synthèse contenus dans les BdR. Plutôt que d'essayer de fouiller directement les schémas de réactions dans un ensemble d'équations chimiques, il semblait préférable de rapprocher le problème considéré de celui de la recherche de sous-graphes fréquents, car ce dernier disposait déjà d'algorithmes efficaces pour sa résolution. Ce rapprochement fut rendu possible en transformant les schémas de réactions par des graphes connexes équivalents (Pennerath et Napoli, 2006). Il s'avérera plus tard que ces graphes alors baptisés graphes de réactions correspondent aux graphes déjà connus sous le nom de graphes condensés de réactions. Par simple transposition des données dans ce nouveau modèle, la recherche des schémas de réactions fréquents pouvait alors se résoudre à l'aide des algorithmes existants de recherche de sous-graphes fréquents. Cette idée simple dans son principe a toutefois nécessité le développement d'une méthode complexe de prétraitement, afin de tenir compte des nombreuses particularités que présentent les réactions des BdR (Pennerath *et al.*, 2008c).

En parallèle à ce développement, j'ai introduit le modèle général des motifs les plus informatifs présenté au chapitre 5, en vue de l'appliquer aux schémas de réactions fréquents

[63]Exception faite des travaux sur les liaisons formables du chapitre 7 qui, pour des raisons de gestion de priorités, ont légèrement précédé les travaux du chapitre 6.

(Pennerath et Napoli, 2007). L'objectif visé par ce modèle était de résumer le contenu de données par un sous-ensemble de motifs fréquents, dont les éléments sont simultanément peu nombreux, représentatifs des données et structurellement peu redondants, afin de pouvoir être analysés directement par l'œil d'un expert. Cette idée anticipait un véritable besoin, mis en évidence lorsque l'achèvement de la méthode de prétraitement des BdR permit d'extraire les premiers schémas de réactions fréquents et fermés fréquents : ces derniers sont en effet apparus tellement nombreux et répétitifs que même un tri par ordre décroissant de score rendait toute tentative d'analyse directe impossible. Le développement d'un premier algorithme prototype – alors très inefficace – pour extraire les motifs les plus informatifs en fouillant directement les données m'a permis de valider l'intérêt du nouveau modèle (Pennerath et Napoli, 2008a,b, 2009) : les schémas les plus informatifs étaient peu nombreux et présentaient une redondance structurelle moindre. Des tests ultérieurs plus aboutis montreront que les fragments de molécules (i.e. de schémas dégénérés) les plus informatifs s'apparentent souvent à des groupes fonctionnels et que la plupart des schémas de réactions non dégénérés sont eux caractéristiques de grandes familles de réactions. Même si les motifs les plus informatifs peuvent ensuite servir à indexer les données, l'objectif premier était d'abord de pouvoir résumer visuellement le contenu des données et ainsi de permettre une analyse de ces dernières par un expert. De ce point de vue les résultats obtenus sur les réactions sont encourageants et affichent un net progrès par rapport à d'autres familles de motifs, y compris certaines familles condensées de motifs comme les fermés fréquents. L'extraction des motifs les plus informatifs restait toutefois relativement coûteuse en temps de calcul. Le développement ultérieur d'un nouvel algorithme fondé sur le filtrage des motifs fréquents s'avérera plus efficace, en particulier lorsque les motifs sont des graphes.

Toutefois les schémas de réactions les plus informatifs n'apportaient pas entière satisfaction au regard de l'objectif que je m'étais fixé initialement, à savoir, l'extraction de schémas caractéristiques de méthodes de synthèse (ou schémas CMS). De ce point de vue, force était de constater que les schémas les plus informatifs n'étaient pas exactement les schémas que je recherchais. Là encore ce résultat était prévisible en raison du manque de discernement des fonctions de score utilisées, qui ne faisaient qu'évaluer la « représentativité » d'un motif. Pour mieux caractériser les schémas CMS, il aurait fallu intégrer dans la méthode davantage de connaissances du domaine, et en particulier les conditions réactionnelles. Ce travail m'est toutefois apparu délicat et très spécifique au domaine d'application. Il m'a semblé plus judicieux de me concentrer sur un autre problème de nature plus informatique, que soulevait l'extraction des motifs les plus informatifs fréquents et plus généralement des motifs fréquents.

En effet les schémas CMS se sont révélés avoir des fréquences très faibles (moins de 0,1 %). Par conséquent, même en supposant disposer de l'algorithme idéal de filtrage des schémas fréquents, qui soit capable de qualifier parfaitement les schémas CMS, la sélection de ces schémas est apparue impossible ou du moins très difficile tant le nombre de schémas de réactions fréquents à extraire s'est révélé élevé. Plutôt que de vouloir extraire l'ensemble des schémas CMS contenus dans un ensemble de réactions, m'est venue l'idée de restreindre la recherche à celle du schéma CMS sous-jacent à une réaction spécifique. Cette idée a été ultérieurement rattachée au principe de transduction qui est une notion utilisée plus particulièrement en apprentissage numérique, pour désigner une méthode qui construit pour chaque exemple de test un modèle spécifique à partir des exemples d'apprentissage plutôt qu'un modèle général unique valable pour tous les exemples de tests. Le problème de l'extraction du schéma CMS sous-jacent à une réaction a pu être abstrait en un problème de recherche de sous-graphe optimal inclus dans un intervalle de graphes et a conduit à développer l'algorithme `CrackReac` (Pennerath *et al.*, 2008b) présenté au chapitre 6. Cet algorithme a ensuite été testé

pour extraire des BdR les schémas CMS sous-jacents aux réactions. Si la fonction de score – semblable à celle utilisée au chapitre 5 – apparaît peu adaptée pour qualifier de manière robuste les schémas CMS, les résultats sont très encourageants d'un point de vue informatique puisque, grâce à l'introduction de la contrainte supplémentaire d'intervalle, `CrackReac` n'a aucune peine à converger vers des motifs de très faible fréquence.

Avant que le développement de `CrackReac` aboutisse, les discussions avec les chimistes Claude Laurenço et Gilles Niel, ont pu faire apparaître des divergences d'appréciation : l'intérêt que je portais au problème de l'extraction des schémas CMS leur a paru assez éloigné de leur sujet de prédilection, à savoir, la synthèse ciblée. Claude Laurenço m'a alors invité à étudier de plus près les travaux réalisés par Jean-Charles Régin sur la prédiction des liaisons stratégiques. Il se trouve par le plus heureux des hasards, que ce problème pouvait se traiter relativement aisément en adaptant le principe de `CrackReac` alors en cours de conception. Cet adaptation a conduit au développement de l'algorithme `GemsBond` (Pennerath *et al.*, 2008a) présenté au chapitre 7, pour la classification supervisée des sommets et des arêtes d'un graphe fondée sur leur environnement topologique. L'application de `GemsBond` à l'évaluation du niveau de formabilité des liaisons des molécules a donné lieu à des résultats expérimentaux très encourageants et sans aucun doute les plus aboutis de ce mémoire. En outre ce travail est celui qui a donné le plus d'échanges fructueux avec les chimistes. Il a notamment permis de stimuler un débat sur les définitions associées aux concepts de liaisons stratégiques et de formabilité des liaisons et a également permis de rapporter notre contribution aux travaux préexistants sur l'accessibilité synthétique.

. . . présent et à venir

Pour clore l'histoire de cette thèse, j'espère que les travaux présentés dans ce mémoire contribueront à faire progresser la fouille de données et la chémoinformatique. Je devine à ce sujet que l'intérêt relatif que les lecteurs éprouveront à la lecture des différents chapitres ne se distribuera pas de la même manière selon qu'ils sont spécialistes de la fouille de données ou chimistes chémoinformaticiens. Ainsi, si j'avais à classer par ordre d'intérêt les chapitres du point de vue de la fouille de données, mes préférences iraient au modèle des motifs les plus informatifs du chapitre 5 suivi par l'« approche transductive » de recherche d'un sous-graphe optimal commune aux chapitres 6 et 7. Le chapitre 4 assez technique sur l'extraction des schémas de réactions fréquents se limite essentiellement à une application des méthodes existantes de fouille de graphes à un problème de chimie. À l'inverse, du point de vue de la chémoinformatique, les chapitres 4 et 7 me semblent les plus porteurs. L'extraction des schémas de réactions fréquents me semble résoudre un problème fondamental de chémoinformatique, qui potentiellement ouvre une voie intéressante pour de nombreuses applications. La prédiction de la formabilité des liaisons présente quant-à-elle l'intérêt de produire de bons résultats directement exploitables dans le cadre de problèmes d'aide à la conception de plans de synthèse ou éventuellement de criblage virtuel. L'extraction des schémas CMS abordée aux chapitres 5 et 6 reste un problème difficile qui nécessite d'intégrer plus finement les connaissances des chimistes avant de pouvoir espérer en exploiter les résultats. De manière générale, les problèmes et méthodes abordés au cours des différents chapitres acceptent de nombreux développements et études complémentaires. On peut citer ici quelques pistes envisageables à court ou moyen terme.

Du point de vue de la fouille de données d'abord, le modèle des motifs les plus informatifs doit être testé dans le cadre d'autres applications et éventuellement pour d'autres types de motifs et structures. Les algorithmes d'extraction exacte peuvent être avantageusement

remplacés par des algorithmes incomplets capables de fournir rapidement une grande partie des motifs les plus informatifs. La notion de redondance structurelle doit également être formalisée afin de quantifier la diminution de redondance qu'apporte le modèle au regard des autres familles de motifs. Par ailleurs, l'« approche transductive » des algorithmes `CrackReac` et `GemsBond` peut être appliquée à d'autres problèmes de classification supervisée ou non supervisée de graphes ou de sommets en adaptant les heuristiques et les fonctions de score utilisées. D'autres stratégies d'exploration et d'élagage de l'espace de recherche peuvent être proposées et l'étude de leur influence sur la qualité des résultats comme sur le temps de calcul doit être approfondie.

Du point de vue des applications en chémoinformatique, toutes les méthodes développées laissent largement la place à des améliorations, que ce soit pour mieux caractériser les schémas CMS ou les liaisons formables. De manière générale, ces améliorations passent par l'injection de connaissances du domaine dans le processus de fouille, et ce en perfectionnant à la fois la fonction de score, la modélisation des données (par l'ajout d'informations supplémentaires dans les graphes, comme la reconnaissance des groupes fonctionnels) et la sélection des réactions fouillées (pour couvrir un ensemble représentatif des différentes méthodes de synthèse). Une analyse par les chimistes des résultats produits par `GemsBond` et `CrackReac` pourrait permettre d'identifier les cas problématiques et d'affiner en conséquence les méthodes selon un processus d'amélioration itératif. `GemsBond` peut également être utilisé pour prédire d'autres caractéristiques que la formabilité des liaisons. Ainsi, on peut penser à l'issue de ces travaux, que la fouille de graphes est un outil idéal pour la chémoinformatique, offrant un compromis idéal entre expressivité du modèle et efficacité des calculs nécessaires. En conséquence, son utilisation devrait vraisemblablement se généraliser dans les années à venir, pour répondre à différents problèmes que se posent les chémoinformaticiens.

Bibliographie

Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), 9-12 December 2002, Maebashi City, Japan, 2002. IEEE Computer Society. ISBN 0-7695-1754-4.

Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), 19-22 December 2003, Melbourne, Florida, USA, 2003. IEEE Computer Society. ISBN 0-7695-1978-4.

Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), 1-4 November 2004, Brighton, UK, 2004. IEEE Computer Society. ISBN 0-7695-2142-8.

R. Agrawal, T. Imielinski et A. Swami : Database mining : A performance perspective. *IEEE T. Knowl. Data. En.*, 5(6):914–925, 1993a.

R. Agrawal et R. Srikant : Fast algorithms for mining association rules in large databases. *In Proceedings of the 20th Conference on Very Large Data Bases (VLDB-94)*, p. 478–499, 1994.

R. Agrawal et R. Srikant : Mining sequential patterns. *In* P. Yu et A. P. Chen, éds : *Proceedings of the Eleventh International Conference on Data Engineering, (ICDE-95), Taipei, Taiwan*, p. 3–14. IEEE Computer Society, 1995.

R. Agrawal, T. Imielinski et A. N. Swami : Mining association rules between sets of items in large databases. *In* P. Buneman et S. Jajodia, éds : *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993*, p. 207–216. ACM Press, 1993b.

R. Agrawal, H. Mannila, R. Srikant, H. Toivonen et A. I. Verkamo : Fast discovery of association rules. *In Advances in Knowledge Discovery and Data Mining*, p. 307–328. AAAI/MIT Press, 1996. ISBN 0-262-56097-6.

T. Anwar, H. Beck et S. Navathe : Knowledge mining by imprecise querying : A classification-based approach. *IEEE Intl. Conf On Data Eng*, p. 622–630, 1992.

T. Asai, K. Abe, S. Kawasoe, H. Arimura, H. Sakamoto et S. Arikawa : Efficient substructure discovery from large semi-structured data. *In* R. L. Grossman, J. Han, V. Kumar, H. Mannila et R. Motwani, éds : *SDM*. SIAM, 2002. ISBN 0-89871-517-2.

T. Asai, H. Arimura, T. Uno et S.-I. Nakano : Discovering frequent substructures in large unordered trees. *In* G. Grieser, Y. Tanaka et A. Yamamoto, éds : *Discovery Science*, vol. 2843 de *Lecture Notes in Computer Science*, p. 47–61. Springer, 2003. ISBN 3-540-20293-5.

J. Auer et J. Bajorath : Distinguishing between bioactive and modeled compound conformations through mining of emerging chemical patterns. *J. Chem. Inf. Model.*, 48(9):1747–1753, 2008.

F. Baader, D. Calvanese, D. McGuinness, D. Nardi et P. Patel-Schneider, éds. *The Description Logic Handbook*. Cambridge University Press, 2002.

L. Babai, D. Y. Grigoryev et D. M. Mount : Isomorphism of graphs with bounded eigenvalue multiplicity. *In STOC '82 : Proceedings of the fourteenth annual ACM symposium on Theory of computing*, p. 310–324, New York, NY, USA, 1982. ACM. ISBN 0-89791-070-2.

J. Bailey, T. Manoukian et K. Ramamohanarao : Fast algorithms for mining emerging patterns. *In* T. Elomaa, H. Mannila et H. Toivonen, éds : *PKDD*, vol. 2431 de *Lecture Notes in Computer Science*, p. 39–50. Springer, 2002. ISBN 3-540-44037-2.

M. Barbut et B. Monjardet : *Ordre et classification – Algèbre et combinatoire (2 tomes)*. Hachette, Paris, 1970.

Y. Bastide, N. Pasquier, R. Taouil, G. Stumme et L. Lakhal : Mining minimal non-redundant association rules using frequent closed itemsets. *In* J. W. Lloyd, V. Dahl, U. Furbach, M. Kerber, K.-K. Lau, C. Palamidessi, L. M. Pereira, Y. Sagiv et P. J. Stuckey, éds : *Computational Logic*, vol. 1861 de *Lecture Notes in Computer Science*, p. 972–986. Springer, 2000a. ISBN 3-540-67797-6.

Y. Bastide, R. Taouil, N. Pasquier, G. Stumme et L. Lakhal : Mining frequent patterns with counting inference. *ACM SIGKDD Explorations*, 2(2):66–75, 2000b.

R. J. Bayardo : Efficiently mining long patterns from databases. *In* Haas et Tiwary (1998), p. 85–93. ISBN 0-89791-995-5.

S. Berasaluce : *Fouille de données et acquisition de connaissances à partir de bases de données de réactions chimiques*. Thèse de chimie informatique et théorique, Université Henri Poincaré Nancy 1, 2002.

S. Berasaluce, C. Laurenço, A. Napoli et G. Niel : An experiment on knowledge discovery in chemical databases. *In* J.-F. Boulicaut, F. Esposito, F. Giannotti et D. Pedreschi, éds : *Knowledge Discovery in Databases : PKDD 2004, 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, Pisa, Italy, September 20-24, 2004, Proceedings*, vol. 3202 de *Lecture Notes in Computer Science*, p. 39–51. Springer, 2004. ISBN 3-540-23108-0.

C. Berge : *The Theory of Graphs and Its Applications*. Wiley, 1962.

S. H. Bertz et T. J. Sommer : Rigorous mathematical approaches to strategic bonds and synthetic analysis based on conceptually simple new complexity indexes. *Chem. Commun. (Cambridge)*, 24(24):2409–2410, 1997.

G. Birkhoff : Lattice theory (first edition). 25, 1940.

H. Blockeel, L. D. Raedt, N. Jacobs et B. Demoen : Scaling up inductive logic programming by learning from interpretations. *Data Min. Knowl. Discov.*, 3(1):59–93, 1999.

K. Boda, T. Seidel et J. Gasteiger : Structure and reaction based evaluation of synthetic accessibility. *Journal of Computer-Aided Molecular Design*, 21(6):311–325, June 2007.

M. Boley, T. Horváth, A. Poigné et S. Wrobel : Efficient closed pattern mining in strongly accessible set systems (extended abstract). *In* J. N. Kok, J. Koronacki, R. L. de Mántaras, S. Matwin, D. Mladenic et A. Skowron, éds : *PKDD*, vol. 4702 de *Lecture Notes in Computer Science*, p. 382–389. Springer, 2007. ISBN 978-3-540-74975-2.

B. Bollobás : *Modern Graph Theory*. Springer-Verlag, 1998.

C. Borgelt et M. R. Berthold : Mining molecular fragments : Finding relevant substructures of molecules. *In Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM '02), 9-12 December 2002, Maebashi City, Japan*, p. 51. IEEE Computer Society, 2002. ISBN 0-7695-1754-4.

J.-F. Boulicaut, M. R. Berthold et T. Horváth, éds. *Discovery Science, 11th International Conference, DS 2008, Budapest, Hungary, October 13-16, 2008. Proceedings*, vol. 5255 de *Lecture Notes in Computer Science*, 2008. Springer. ISBN 978-3-540-88410-1.

J.-F. Boulicaut, A. Bykowski et C. Rigotti : Free-sets : A condensed representation of boolean data for the approximation of frequency queries. *Data Min. Knowl. Discov.*, 7 (1):5–22, 2003.

L. Breiman, J. H. Friedman, R. A. Olshen et C. J. Stone : *Classification and Regression Trees*. Wadsworth, 1984. ISBN 0-534-98053-8.

S. Brin, R. Motwani, J. D. Ullman et S. Tsur : Dynamic itemset counting and implication rules for market basket data. *In SIGMOD '97 : Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, p. 255–264, New York, NY, USA, 1997. ACM. ISBN 0-89791-911-4.

B. Bringmann et S. Nijssen : What is frequent in a single graph ? *In* Washio *et al.* (2008), p. 858–863. ISBN 978-3-540-68124-3.

B. Bringmann et A. Zimmermann : The chosen few : On identifying valuable patterns. *In Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), October 28-31, 2007, Omaha, Nebraska, USA*, p. 63–72, 2007.

F. K. Brown : Chapter 35. chemoinformatics : What is it and how does it impact drug discovery. *Annual Reports in Med. Chem.*, 33:375–384, 1998.

B. Buchanan et E. Feigenbaum : Dendral and meta-dendral : Their applications dimensions. 11:5–24, 1978.

H. Bunke, P. Foggia, C. Guidobaldi, C. Sansone et M. Vento : A comparison of algorithms for maximum common subgraph on randomly connected graphs. *In* T. Caelli, A. Amin, R. P. W. Duin, M. S. Kamel et D. de Ridder, éds : *SSPR/SPR*, vol. 2396 de *Lecture Notes in Computer Science*, p. 123–132. Springer, 2002. ISBN 3-540-44011-9.

H. Bunke et K. Shearer : A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3-4):255–259, 1998.

C. Chen, X. Yan, P. S. Yu, J. Han, D.-Q. Zhang et X. Gu : Towards graph containment search and indexing. *In* C. Koch, J. Gehrke, M. N. Garofalakis, D. Srivastava, K. Aberer, A. Deshpande, D. Florescu, C. Y. Chan, V. Ganti, C.-C. Kanne, W. Klas et E. J. Neuhold, éds : *VLDB*, p. 926–937. ACM, 2007. ISBN 978-1-59593-649-3.

W. L. Chen : Chemoinformatics : Past, present, and future. *J. Chem. Inf. Model.*, 46 (6):2230–2255, 2006. URL `http ://pubs.acs.org/doi/abs/10.1021/ci060016u`.

H. Cheng, X. Yan, J. Han et C.-W. Hsu : Discriminative frequent pattern analysis for effective classification. *In ICDE*, p. 716–725. IEEE, 2007.

Y. Chi, R. R. Muntz, S. Nijssen et J. N. Kok : Frequent subtree mining - an overview. *Fundam. Inform.*, 66(1-2):161–198, 2005a.

Y. Chi, Y. Xia, Y. Yang et R. R. Muntz : Mining closed and maximal frequent subtrees from databases of labeled rooted trees. *IEEE Trans. Knowl. Data Eng.*, 17(2):190–202, 2005b.

Y. Chi, Y. Yang et R. R. Muntz : Indexing and mining free trees. *In ICDM* icd (2003), p. 509–512. ISBN 0-7695-1978-4.

Y. Chi, Y. Yang et R. R. Muntz : Hybridtreeminer : An efficient algorithm for mining frequent rooted trees and free trees using canonical form. *In SSDBM*, p. 11–20. IEEE Computer Society, 2004a. ISBN 0-7695-2146-0.

Y. Chi, Y. Yang, Y. Xia et R. R. Muntz : Cmtreeminer : Mining both closed and maximal frequent subtrees. *In* H. Dai, R. Srikant et C. Zhang, éds : *PAKDD*, vol. 3056 de *Lecture Notes in Computer Science*, p. 63–73. Springer, 2004b. ISBN 3-540-22064-X.

E. F. Codd : Recent investigations in relational data base systems. *In IFIP Congress*, p. 1017–1021, 1974.

A. Colmerauer et P. Roussel : The birth of prolog. *In HOPL Preprints*, p. 37–52, 1993.

D. J. Cook et L. B. Holder : Substructure discovery using minimum description length and background knowledge. *J. of Art. Intell. Res.*, 1:231–255, 1994. URL `citeseer.ist.psu.edu/article/cook94substructure.html`.

D. J. Cook et L. B. Holder : Graph-based data mining. *IEEE Intell. Syst.*, 15(2):32–41, 2000.

D. J. Cook et L. B. Holder, éds. *Mining Graph Data*. Wiley-Interscience, 2006.

E. Corey : Computer–assisted analysis of complex synthetic problems. *Quaterly Review of the Chemical Society*, 25:455–482, 1971.

E. Corey et X. Cheng : *The Logic of Chemical Synthesis*. John Wiley & Sons, New York, 1995.

E. Corey et W. Wipke : Computer–assisted analysis of complex syntheses. *Science*, 166:178–192, 1969.

D. G. Corneil et C. C. Gotlieb : An efficient algorithm for graph isomorphism. *J. ACM*, 17(1):51–64, 1970.

T. Cover et P. Hart : Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967. URL `http ://ieeexplore.ieee.org/xpls/abs_all.jsp ?arnumber=1053964`.

B. Crémilleux et A. Soulet : Discovering knowledge from local patterns with global constraints. *In* O. Gervasi, B. Murgante, A. Laganà, D. Taniar, Y. Mun et M. L. Gavrilova, éds : *ICCSA (2)*, vol. 5073 de *Lecture Notes in Computer Science*, p. 1242–1257. Springer, 2008. ISBN 978-3-540-69840-1.

B. Cuissart : *Plus grande structure commune à deux graphes : méthode de calcul et intérêt dans un contexte SAR*. Thèse de doctorat, Université de Caen Basse-Normandie, 2004.

L. Dehaspe et L. D. Raedt : Mining association rules in multiple relations. *In* N. Lavrac et S. Dzeroski, éds : *ILP*, vol. 1297 de *Lecture Notes in Computer Science*, p. 125–132. Springer, 1997. ISBN 3-540-63514-9.

L. Dehaspe, H. Toivonen et R. D. King : Finding frequent substructures in chemical compounds. *In* R. Agrawal, P. Stolorz et G. Piatetsky-Shapiro, éds : *4th International Conference on Knowledge Discovery and Data Mining*, p. 30–36. AAAI Press., 1998. URL `citeseer.ist.psu.edu/dehaspe98finding.html`.

M. Deshpande, M. Kuramochi et G. Karypis : Frequent sub-structure-based approaches for classifying chemical compounds. *In ICDM* icd (2003), p. 35–42. ISBN 0-7695-1978-4.

G. Dong et J. Li : Efficient mining of emerging patterns : discovering trends and differences. *In KDD '99 : Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 43–52, New York, NY, USA, 1999. ACM. ISBN 1-58113-143-7.

J.-E. Dubois et Y. Sobel : Darc system for documentation and artificial intelligence in chemistry. *Journal of Chemical Information and Computer Sciences*, 25(3):326–333, 1985.

H. Ehrig, K. Ehrig, U. Prange et G. Taentzer : *Fundamentals of Algebraic Graph Transformation*. Monographs in Theoretical Computer Science. An EATCS Series. Springer-Verlag, 2006. ISBN 978-3-540-31187-4.

T. Fawcett : An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874, 2006. ISSN 0167-8655.

U. Fayyad, G. Piatetsky-Shapiro et P. Smyth : Knowledge discovery and data mining : Towards a unifying framework. p. 82–88, 1996a.

U. Fayyad, G. Piatetsky-Shapiro, P. Smyth et R. Uthurusamy, éds. *Advances in Knowledge Discovery and Data Mining*. AAAI Press / MIT Press, Menlo Park, California, 1996b.

U. M. Fayyad, G. Piatetsky-Shapiro et P. Smyth : From data mining to knowledge discovery : An overview. *In Advances in Knowledge Discovery and Data Mining*, p. 1–34. MIT Press, 1996c.

U. M. Fayyad, N. Weir et S. G. Djorgovski : Skicat : A machine learning system for automated cataloging of large scale sky surveys. *In ICML*, p. 112–119, 1993.

S. Ferré, O. Ridoux et B. Sigonneau : Arbitrary relations in formal concept analysis and logical information systems. *In* F. Dau, M.-L. Mugnier et G. Stumme, éds : *ICCS*, vol. 3596 de *Lecture Notes in Computer Science*, p. 166–180. Springer, 2005. ISBN 3-540-27783-8.

S. Ferré : *Systèmes d'information logiques : un paradigme logico-contextuel pour interroger, naviguer et apprendre*. Thèse de doctorat, Université de Rennes 1, octobre 2002.

M. Fiedler et C. Borgelt : Support computation for mining frequent subgraphs in a single graph. *In* P. Frasconi, K. Kersting et K. Tsuda, éds : *MLG*, 2007.

I. Fischer et T. Meinl : Graph based molecular data mining - an overview. *In Proceedings of the IEEE International Conference on Systems, Man & Cybernetics : The Hague, Netherlands, 10-13 October 2004*, p. 4578–4582. IEEE, 2004. ISBN 0-7803-8566-7.

H. Fröhlich, J. K. Wegner, F. Sieker et A. Zell : Optimal assignment kernels for attributed molecular graphs. *In* L. D. Raedt et S. Wrobel, éds : *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, vol. 119 de *ACM International Conference Proceeding Series*, p. 225–232. ACM, 2005. ISBN 1-59593-180-5.

S. Fujita : Description of organic reactions based on imaginary transition structures. 1. introduction of new concepts. *Journal of Chemical Information and Computer Sciences*, 26(4):205–212, 1986.

B. Ganter et R. Wille : *Formal Concept Analysis*. Springer, Berlin, 1999.

B. Ganter, P. A. Grigoriev, S. O. Kuznetsov et M. V. Samokhin : Concept-based data mining with scaled labeled graphs. *In* K. E. Wolff, H. D. Pfeiffer et H. S. Delugach, éds : *ICCS*, vol. 3127 de *Lecture Notes in Computer Science*, p. 94–108. Springer, 2004. ISBN 3-540-22392-4.

B. Ganter et S. O. Kuznetsov : Pattern structures and their projections. *In* H. S. Delugach et G. Stumme, éds : *ICCS*, vol. 2120 de *Lecture Notes in Computer Science*, p. 129–142. Springer, 2001. ISBN 3-540-42344-3.

B. Ganter et S. O. Kuznetsov : Hypotheses and version spaces. *In* A. de Moor, W. Lex et B. Ganter, éds : *ICCS*, vol. 2746 de *Lecture Notes in Computer Science*, p. 83–95. Springer, 2003. ISBN 3-540-40576-3.

B. Ganter, G. Stumme et R. Wille, éds. *Formal Concept Analysis, Foundations and Applications*, vol. 3626 de *Lecture Notes in Computer Science*, 2005. Springer. ISBN 3-540-27891-5.

M. R. Garey et D. S. Johnson : *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979. ISBN 0716710447.

T. Gärtner : A survey of kernels for structured data. *SIGKDD Explorations*, 5(1):49–58, 2003.

J. Gasteiger, W. D. Ihlenfeldt et P. Röse : A collection of computer methods for synthesis design and reaction prediction. *Recl. Trav. Chim.*, 111:270–290, 1992.

J. Gasteiger, W. D. Ihlenfeldt, P. Röse et R. Wanke : Computer-assisted reaction prediction and synthesis design. *Analytica Chimica Acta*, 235:65–75, 1990.

J. Gasteiger et T. Engel, éds. *Chemoinformatics : A Textbook.* John Wiley & Sons, 2004.

J. Gennari, P. Langley et D. Fisher : Models of incremental concept formation. 40:11–61, 1989.

L. Getoor et C. P. Diehl : Link mining : a survey. *SIGKDD Explorations*, 7(2):3–12, 2005.

L. Getoor et B. Taskar, éds. *Introduction to Statistical Relational Learning.* The MIT Press, Cambridge (MA), 2007.

O. Gien : *Modélisation de la synthèse organique multi-étapes. Développement d'outils informatiques d'aide à la conception de plans de synthèse.* Thèse de chimie moléculaire, Université de Montpellier II Sciences et Techniques du Languedoc, 1998.

K. Gouda et M. J. Zaki : Genmax : An efficient algorithm for mining maximal frequent itemsets. *Data Min. Knowl. Discov.*, 11(3):223–242, 2005.

E. Grosholz et R. Hoffmann : How symbolic and iconic languages bridge the two worlds of the chemist : A case study from contemporary bioorganic chemistry. *In* N. B. Rosenfeld, éd. : *Of Minds and Molecules : New Philosophical Perspectives on Chemistry*, chap. 13, p. 230–257. Oxford University Press, 2000.

L. M. Haas et A. Tiwary, éds. *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA*, 1998. ACM Press. ISBN 0-89791-995-5.

M. R. Hacene, M. Huchard, A. Napoli et P. Valtchev : A proposal for combining formal concept analysis and description logics for mining relational data. *In* S. O. Kuznetsov et S. Schmidt, éds : *ICFCA*, vol. 4390 de *Lecture Notes in Computer Science*, p. 51–65. Springer, 2007. ISBN 3-540-70828-6.

J. Han, Y. Cai et N. Cercone : Knowledge discovery in databases : An attribute-oriented approach. *In Proceedings of the 18th Conference on "Very Large Databases (VLDB'92)*, p. 547–559, 1992.

J. Han : *Data Mining : Concepts and Techniques.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005. ISBN 1558609016.

J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal et M. Hsu : Freespan : frequent pattern-projected sequential pattern mining. *In KDD*, p. 355–359, 2000a.

J. Han, J. Pei et Y. Yin : Mining frequent patterns without candidate generation. *In* W. Chen, J. F. Naughton et P. A. Bernstein, éds : *SIGMOD Conference*, p. 1–12. ACM, 2000b. ISBN 1-58113-218-2.

J. Han, J. Pei, Y. Yin et R. Mao : Mining frequent patterns without candidate generation : A frequent-pattern tree approach. *Data Min. Knowl. Discov.*, 8(1):53–87, 2004.

D. Hand, H. Mannila et P. Smyth : *Principles of Data Mining.* The MIT Press, Cambridge (MA), 2001.

S. Hanessian : Man, machine and visual imagery in strategic synthesis planning : computer-perceived precursors for drug candidates. *Curr. Opin. Drug Discovery Dev.*, 8(6):798–819, 2005.

S. Hanessian, J. Franco et B. Larouche : The psychobiological basis of heuristic synthesis planning–man, machine and the chiron approach. *Pure & Applied Chemistry*, 62(2):1887–1910, 1990.

R. Haralick et G. Elliott : Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, p. 263–313, 1980.

M. A. Hasan, V. Chaoji, S. Salem, J. Besson et M. J. Zaki : Origami : Mining representative orthogonal graph patterns. *In Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), October 28-31, 2007, Omaha, Nebraska, USA*, p. 153–162. IEEE Computer Society, 2007.

C. Helma, T. Cramer, S. Kramer et L. D. Raedt : Data mining and machine learning techniques for the identification of mutagenicity inducing substructures and structure activity relationships of noncongeneric compounds. *J. Chem. Inf. Comput. Sci.*, 44 (4):1402–1411, 2004.

J. B. Hendrickson : Comprehensive system for classification and nomenclature of organic reactions. *Journal of Chemical Information and Computer Sciences*, 37(5):852–860, 1997.

J. B. Hendrickson et A. G. Toczko : Syngen program for synthesis design : basic computing techniques. *Journal of Chemical Information and Computer Sciences*, 29(3):137–145, 1989.

M. Holsheimer, M. Kersten, H. Mannila et H. Toivonen : A perspective on databases and data mining. *In Proceedings of the First International Conference on Knowledge Discovery & Data Mining (KDD-95), Montréal*, p. 150–155, 1995.

J. E. Hopcroft et J. K. Wong : Linear time algorithm for isomorphism of planar graphs (preliminary report). *In STOC*, p. 172–184. ACM, 1974.

T. Horváth, T. Gärtner et S. Wrobel : Cyclic pattern kernels for predictive graph mining. *In* Kim *et al.* (2004), p. 158–167. ISBN 1-58113-888-1.

J. Huan, W. Wang et J. Prins : Efficient mining of frequent subgraph in the presence of isomorphism. *In In ICDM*, p. 549–552, 2003.

J. Huan, W. Wang, J. Prins et J. Yang : Spin : mining maximal frequent subgraphs from graph databases. *In* Kim *et al.* (2004), p. 581–586. ISBN 1-58113-888-1.

M. Huchard, M. R. Hacene, C. Roume et P. Valtchev : Relational concept discovery in structured datasets. *Ann. Math. Artif. Intell.*, 49(1-4):39–76, 2007.

T. Imielinski et H. Mannila : A database perspective on knowledge discovery. *Commun. ACM*, 39(11):58–64, 1996.

A. Inokuchi et H. Kashima : Mining significant pairs of patterns from graph structures with class labels. *In ICDM* icd (2003), p. 83–90. ISBN 0-7695-1978-4.

A. Inokuchi, T. Washio et H. Motoda : An apriori-based algorithm for mining frequent substructures from graph data. *In PKDD '00 : Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, p. 13–23. Springer-Verlag, 2000. ISBN 3-540-41066-X.

P. Jauffret, T. Hander, C. Tonnelier et G. Kaufmann : Machine learning of generic reactions : I) scope of the project. *Tetrahedron Computer Methodology*, 3:323–333, 1990.

P. Jauffret, T. Hanser, J. F. Marchaland, H. Vogel, T. Fang et G. Kaufmann : Grams : A network generator for synthetic methods learning in organic chemistry. *In The first European conference on computational chemistry (E.C.C.C1).*, vol. 330, p. 569–574. AIP Conference Proceedings, 1995.

I. Jonyer, D. J. Cook et L. B. Holder : Graph-based hierarchical conceptual clustering. *Journal of Machine Learning Research*, 2:19–43, 2001.

A. Jorge, L. Torgo, P. Brazdil, R. Camacho et J. Gama, éds. *Knowledge Discovery in Databases : PKDD 2005, 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Porto, Portugal, October 3-7, 2005, Proceedings*, vol. 3721 de *Lecture Notes in Computer Science*, 2005. Springer. ISBN 3-540-29244-6.

W. L. Jorgensen, E. R. Laird, A. J. Gushurst, J. M. Fleischer, S. A. Gothe, H. E. Helson, G. D. Paderes et S. Sinclair : Cameo : a program for the logical prediction of the products of organic reactions. *Pure and Applied Chemistry*, 62(10):1921–1932, 1990.

A. Karwath et L. D. Raedt : Predictive graph mining. *In* E. Suzuki et S. Arikawa, éds : *Discovery Science*, vol. 3245 de *Lecture Notes in Computer Science*, p. 1–15. Springer, 2004. ISBN 3-540-23357-1.

H. Kashima et Y. Tsuboi : Kernel-based discriminative learning algorithms for labeling sequences, trees, and graphs. *In* C. E. Brodley, éd. : *ICML*, vol. 69 de *ACM International Conference Proceeding Series*. ACM, 2004.

J. Kazius, S. Nijssen, J. Kok, T. Bäck, et A. P. IJzerman : Substructure mining using elaborate chemical representation. *J. Chem. Inf. Model.*, 46(2):597–605, 2006.

P. J. Kelly : A congruence theorem for trees. *Pacific Journal of Mathematics*, 7(1):961–968, 1957. URL `http ://ProjectEuclid.org/getRecord?id=euclid.pjm/1103043674`.

W. Kim, R. Kohavi, J. Gehrke et W. DuMouchel, éds. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, 2004. ACM. ISBN 1-58113-888-1.

G. Klopman : Artificial intelligence approach to structure-activity studies. computer automated structure evaluation of biological activity of organic molecules. *Journal of the American Chemical Society*, 106(24):7315–7321, 1984. URL `http ://pubs.acs.org/doi/abs/10.1021/ja00336a004`.

R. A. Kowalski : Predicate logic as programming language. *In IFIP Congress*, p. 569–574, 1974.

S. Kramer et L. D. Raedt : Feature construction with version spaces for biochemical applications. *In* C. E. Brodley et A. P. Danyluk, éds : *ICML*, p. 258–265. Morgan Kaufmann, 2001. ISBN 1-55860-778-1.

S. Kramer, L. D. Raedt et C. Helma : Molecular feature mining in hiv data. *In KDD*, p. 136–143, 2001.

M. Kuramochi et G. Karypis : Frequent subgraph discovery. *In* N. Cercone, T. Y. Lin et X. Wu, éds : *Proceedings of the 2001 IEEE International Conference on Data Mining, 29 November - 2 December 2001, San Jose, California, USA*, p. 313–320. IEEE Computer Society, 2001. ISBN 0-7695-1119-8.

M. Kuramochi et G. Karypis : An efficient algorithm for discovering frequent subgraphs. *IEEE Trans. Knowl. Data Eng.*, 16(9):1038–1051, 2004a.

M. Kuramochi et G. Karypis : Grew-a scalable frequent subgraph discovery algorithm. *In ICDM* icd (2004), p. 439–442. ISBN 0-7695-2142-8.

S. O. Kuznetsov : Learning of simple conceptual graphs from positive and negative examples. *In* J. M. Zytkow et J. Rauch, éds : *PKDD*, vol. 1704 de *Lecture Notes in Computer Science*, p. 384–391. Springer, 1999. ISBN 3-540-66490-4.

P. Langley : *Elements of Machine Learning*. Morgan Kaufmann Publishers, San Francisco, California, 1996.

C. Laurenço : *Synthèse organique assistée par ordinateur*. Thèse de doctorat d'État, Université Louis Pasteur, Strasbourg, 1985.

C. Laurenço : Rapport d'activité du gdr 1093 du cnrs. traitement informatique de la connaissance en chimie organique. Rap. tech., CCIPE, 1998.

A. R. Leach et V. J. Gillet : *An Introduction to Chemoinformatics*. Springer, 2003.

N. Lesh, M. J. Zaki et M. Ogihara : Scalable feature mining for sequential data. *IEEE Intell. Syst.*, 15(2):48–56, 2000. URL `citeseer.ist.psu.edu/lesh00scalable.html`.

M. Liquiere et J. Sallantin : Structural machine learning with galois lattice and graphs. *In* J. W. Shavlik, éd. : *ICML*, p. 305–313. Morgan Kaufmann, 1998. ISBN 1-55860-556-8.

H. Liu, H. Lu, L. Feng et F. Hussain : Efficient search of reliable exceptions. *In* N. Zhong et L. Zhou, éds : *PAKDD*, vol. 1574 de *Lecture Notes in Computer Science*, p. 194–203. Springer, 1999. ISBN 3-540-65866-1.

D. J. Lubinsky : Discovery from databases : A review of ai and statistical techniques. *In Proceedings of IJCAI-89 Workshop on Knowledge Discovery in Databases*, p. 204–218, 1989.

E. M. Luks : Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comput. Syst. Sci.*, 25(1):42–65, 1982.

P. Mahé, N. Ueda, T. Akutsu, J. L. Perret et J. P. Vert : Graph kernels for molecular structure-activity relationship analysis with support vector machines. *J. Chem. Inf. Model.*, 45(4):939–951, 2005. ISSN 1549-9596. URL `http ://dx.doi.org/10.1021/ci050039t`.

H. Mannila et H. Toivonen : Discovering generalized episodes using minimal occurrences. *In KDD*, p. 146–151, 1996.

H. Mannila, H. Toivonen et A. I. Verkamo : Discovering frequent episodes in sequences. *In KDD*, p. 210–215, 1995.

J. J. Mcgregor : Backtrack search algorithms and the maximal common subgraph problem. *SOFTWARE-PRACT. AND EXPER.*, 12(1):23–34, 1982.

B. D. McKay : Practical graph isomorphism. *Congr. Numer.*, 30:45–87, 1981. URL `http ://cs.anu.edu.au/~bdm/nauty/PGI`.

R. Michalski et R. Stepp : Automated construction of classifications : Conceptual clustering versus numerical taxonomy. 5(4):396–410, 1983.

R. S. Michalski, L. Kerschberg, K. A. Kaufman et J. S. Ribeiro : Mining for knowledge in databases : The inlen architecture, initial implementation and first results. *J. Intell. Inf. Syst.*, 1(1):85–113, 1992.

T. Mitchell : *Machine Learning*. McGraw-Hill, Boston, Massachusetts, 1997.

T. M. Mitchell : Version spaces : A candidate elimination approach to rule learning. *In IJCAI*, p. 305–310, 1977.

S. Muggleton : Inductive logic programming. *In ALT*, p. 42–62, 1990.

S. Muggleton : Inductive logic programming. *New Generation Comput.*, 8(4):295–, 1991.

S. Muggleton : Inverse entailment and progol. *New Generation Comput.*, 13(3&4):245–286, 1995.

S. Muggleton, A. Srinivasan, R. D. King et M. J. E. Sternberg : Biochemical knowledge discovery using inductive logic programming. *In DS '98 : Proceedings of the First International Conference on Discovery Science*, p. 326–341. Springer-Verlag, 1998. ISBN 3-540-65390-2.

A. Napoli : *Représentations à objets et raisonnement par classification en intelligence artificielle*. Thèse d'état, Université de Nancy 1, 1992.

A. Napoli : Une introduction aux logiques de descriptions. Rap. tech. RR-3314, LORIA, Nancy, France, 1997.

A. Napoli, C. Laurenço et R. Ducournau : An object-based representation system for organic synthesis planning. *Int. J. Hum.-Comput. Stud.*, 41(1-2):5–32, 1994.

R. T. Ng, L. V. S. Lakshmanan, J. Han et A. Pang : Exploratory mining and pruning optimizations of constrained association rules. *In* Haas et Tiwary (1998), p. 13–24. ISBN 0-89791-995-5.

S. Nijssen et J. Kok : Efficient discovery of frequent unordered trees. *In Proceedings of the first International Workshop on Mining Graphs, Trees and Sequences (MGTS2003)*, 2003.

S. Nijssen et J. N. Kok : Faster association rules for multiple relations. *In* B. Nebel, éd. : *IJCAI*, p. 891–896. Morgan Kaufmann, 2001. ISBN 1-55860-777-3.

S. Nijssen et J. N. Kok : A quickstart in frequent structure mining can make a difference. *In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, p. 647–652. ACM Press, 2004. ISBN 1-58113-888-9.

E. R. Omiecinski : Alternative interest measures for mining associations in databases. *IEEE Trans. on Knowl. and Data Eng.*, 15(1):57–69, 2003. ISSN 1041-4347.

M. Ott : Cheminformatics and organic chemistry. computer-assisted synthetic analysis. *Cheminformatics*, 1(1-2):83–109, 2004.

A. N. Papadopoulos, A. Lyritsis et Y. Manolopoulos : Skygraph : an algorithm for important subgraph discovery in relational graphs. *Data Min. Knowl. Discov.*, 17(1):57–76, 2008. ISSN 1384-5810.

N. Pasquier, Y. Bastide, R. Taouil et L. Lakhal : Discovering frequent closed itemsets for association rules. *In* C. Beeri et P. Buneman, éds : *Proceedings of the 7th International Conference on Database Theory (ICDT'99), Jerusalem, Israel*, Lecture Notes in Computer Science 1540, p. 398–416. Springer, 1999a.

N. Pasquier, Y. Bastide, R. Taouil et L. Lakhal : Efficient mining of association rules using closed itemset lattices. *International Journal of Information Systems*, 24(1):25–46, 1999b.

N. Pasquier, R. Taouil, Y. Bastide, G. Stumme et L. Lakhal : Generating a condensed representation for association rules. *J. Intell. Inf. Syst.*, 24(1):29–60, 2005.

J. Pei, J. Han et L. V. S. Lakshmanan : Pushing convertible constraints in frequent itemset mining. *Data Min. Knowl. Discov.*, 8(3):227–252, 2004a.

J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal et M. Hsu : Mining sequential patterns by pattern-growth : The prefixspan approach. *IEEE Trans. Knowl. Data Eng.*, 16(11):1424–1440, 2004b.

F. Pennerath et A. Napoli : La fouille de graphes dans les bases de données réactionnelles au service de la synthèse en chimie organique. *In* G. Ritschard et C. Djeraba, éds : *Extraction et gestion des connaissances (EGC'2006), Actes des sixièmes journées Extraction et Gestion des Connaissances, Lille, France, 17-20 janvier 2006, 2 Volumes*, vol. RNTI-E-6 de *Revue des Nouvelles Technologies de l'Information*, p. 517–528. Cépaduès-Éditions, 2006. ISBN 2-85428-718-5.

F. Pennerath et A. Napoli : Mining most informative subgraphs. *Poster Session of Mining and Learning With Graphs 2007, Firenze*, 2007.

F. Pennerath et A. Napoli : La famille des motifs les plus informatifs. application à l'extraction de graphes en chimie organique. *Revue I3*, 8(2):252 pp, 2008a.

F. Pennerath et A. Napoli : Le problème de l'extraction des graphes d'intérêt maximal. application à la fouille de réactions chimiques. *In* P. Marquis et I. Bloch, éds : *Actes du 16ème Congrès Francophone sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA 2008), Amiens, France*, p. 133–142, 2008b.

F. Pennerath et A. Napoli : The model of most informative patterns and its application to knowledge extraction from graph databases. *In* W. Buntimes, M. Grobelnik et J. Shawe-Taylor, éds : *ECML/PKDD (2)*, vol. 5782 de *Lecture Notes in Artificial Intelligence*, p. 205–220. Springer, 2009.

F. Pennerath, G. Polaillon et A. Napoli : A method for classifying vertices of labeled graphs applied to knowledge discovery from molecules. *In Proceedings of the 18th European Conference on Artificial Intelligence (ECAI'08), Patras, Greece*, p. 147–151, 2008a.

F. Pennerath, G. Polaillon et A. Napoli : Mining intervals of graphs to extract characteristic reaction patterns. *In* Boulicaut *et al.* (2008), p. 210–221. ISBN 978-3-540-88410-1.

F. Pennerath, G. Polaillon et A. Napoli : Prétraitement des bases de données de réactions chimiques pour la fouille de schémas de réactions. *In* F. Guillet et B. Trousse, éds : *Extraction et gestion des connaissances (EGC'2008), Actes des 8èmes journées Extraction et Gestion des Connaissances, Sophia-Antipolis, France, 29 janvier au 1er février 2008, 2 Volumes*, vol. RNTI-E-11 de *Revue des Nouvelles Technologies de l'Information*, p. 547–558. Cépaduès-Éditions, 2008c.

M. Pfoertner et M. Sitzmann : Computer-assisted synthesis design by wodca (casd). *In Handbook of chemoinformatics*, p. 1457–1507. Wiley-VCH, Weinheim, 2003.

G. Piatetsky-Shapiro et W. Frawley, éds. *Knowledge-Discovery in Databases*. AAAI Press / MIT Press, Menlo Park (Ca) and Cambridge (Ma), 1991.

G. Piatetsky-Shapiro : Discovery, analysis, and presentation of strong rules. *In Knowledge Discovery in Databases*, p. 229–248. AAAI/MIT Press, 1991. ISBN 0-262-62080-4.

G. Polaillon : *Organisation et interprétation par les treillis de Galois de données de type multi-valué, intervalle ou histogramme*. Thèse d'informatique, Université Paris IX (Dauphine), 1998.

G. Polaillon et E. G. Diday : Symbolic galois lattices of multivariate and interval tables. *In Proceedings of OSDA'97, Darmstadt.1997*, 1997.

E. Proschak, J. K. Wegner, A. Schüller, G. Schneider et U. Fechner : Molecular query language (mql)a context-free grammar for substructure matching. *J. Chem. Inf. Model.*, 47(2):295–301, 2007.

E. Prud'hommeaux et A. Seaborne : Sparql query language for rdf, 2008. URL `http ://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115`.

J. R. Quinlan : *C4.5 : programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.

J. Quinlan : Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

L. D. Raedt, P. Idestam-Almquist et G. Sablon : θ-subsumption for structural matching. *In* M. van Someren et G. Widmer, éds : *Machine Learning : ECML97*, Lecture Notes in Artificial Intelligence 1224, p. 73–84. Springer Verlag, Berlin, 1997.

L. D. Raedt : Logical settings for concept-learning. *Artif. Intell.*, 95(1):187–201, 1997.

L. D. Raedt : Statistical relational learning : An inductive logic programming perspective. *In* Jorge *et al.* (2005), p. 3–5. ISBN 3-540-29244-6.

L. D. Raedt : *Logical and Relational Learning*. Cognitive Technologies. Springer, 2008. URL `http ://www.springer.com/computer/artificial/book/978-3-540-20040-6`.

L. D. Raedt, T. G. Dietterich, L. Getoor, K. Kersting et S. Muggleton, éds. *Probabilistic, Logical and Relational Learning - A Further Synthesis, 15.04. - 20.04.2007*, vol. 07161 de *Dagstuhl Seminar Proceedings*, 2008a. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany.

L. D. Raedt et S. Dzeroski : First-order jk-clausal theories are pac-learnable. *Artif. Intell.*, 70(1-2):375–392, 1994.

L. D. Raedt, T. Guns et S. Nijssen : Constraint programming for itemset mining. *In* Y. Li, B. Liu et S. Sarawagi, éds : *KDD*, p. 204–212. ACM, 2008b. ISBN 978-1-60558-193-4.

L. D. Raedt, M. Jaeger, S. D. Lee et H. Mannila : A theory of inductive query answering. *In ICDM* icd (2002), p. 123–130. ISBN 0-7695-1754-4.

C. Raïssi, T. Calders et P. Poncelet : Mining conjunctive sequential patterns. *In* W. Daelemans, B. Goethals et K. Morik, éds : *ECML/PKDD (1)*, vol. 5211 de *Lecture Notes in Computer Science*, p. 19. Springer, 2008. ISBN 978-3-540-87478-2.

J.-C. Régin : *Développement d'outils algorithmiques pour l'intelligence artificielle. Application à la chimie organique*. Thèse de doctorat, Université des Sciences et Techniques du Languedoc, Montpellier, 1995.

J.-C. Régin, O. Gascuel et C. Laurenço : Machine learning of strategic knowledge in organic synthesis from reaction databases. *In* F. Bernardi et J. Rivail, éds : *Proceedings of the First European Conference on Computational Chemistry (E.C.C.C-1), Nancy, France, May 23-27*, p. 618–623, Woodbury, NY, 1995. AIP Press.

J. Rissanen : Modeling by shortest data description. *Autom.*, 14:465–471, 1978.

J. R. Rose et J. Gasteiger : Horace : An automatic system for the hierarchical classification of chemical reactions. *Journal of Chemical Information and Computer Sciences*, 34(1):74–90, 1994.

U. Rückert et S. Kramer : Frequent free tree discovery in graph data. *In* H. Haddad, A. Omicini, R. L. Wainwright et L. M. Liebrock, éds : *SAC*, p. 564–570. ACM, 2004. ISBN 1-58113-812-1.

C. Ruecker, G. Ruecker et S. H. Bertz : Organic synthesis-art or science ? *J. Chem. Inf. Comput. Sci.*, 44(2):378–386, 2004.

S. J. Russell et P. Norvig : *Artificial Intelligence : A Modern Approach*. Prentice Hall, 2 édn, 2002.

H. Satoh, O. Sacher, T. Nakata, L. Chen, J. Gasteiger et K. Funatsu : Classification of organic reactions : ; similarity of reactions based on changes in the electronic features of oxygen atoms at the reaction sites1. *Journal of Chemical Information and Computer Sciences*, 38(2):210–219, 1998. URL `http ://pubs.acs.org/doi/abs/10.1021/ci9701190`.

D. C. Schmidt et L. E. Druffel : A fast backtracking algorithm to test directed graphs for isomorphism using distance matrices. *J. ACM*, 23(3):433–445, 1976.

S. Sena, H. Agarwalb et S. Senc : Chemical equation balancing : An integer programming approach. *Mathematical and Computer Modelling*, 44:678–691, 2006.

E. Y. Shapiro : *Algorithmic Program DeBugging*. MIT Press, Cambridge, MA, USA, 1983. ISBN 0262192187.

A. Siebes, J. Vreeken et M. van Leeuwen : Item sets that compress. *In* J. Ghosh, D. Lambert, D. B. Skillicorn et J. Srivastava, éds : *SDM*. SIAM, 2006. ISBN 0-89871-611-X.

R. F. Simmons : Storage and retrieval of aspects of meaning in directed graph structures. *Commun. ACM*, 9(3):211–215, 1966.

A. Soulet et B. Crémilleux : An efficient framework for mining flexible constraints. *In* T. B. Ho, D. W.-L. Cheung et H. Liu, éds : *PAKDD*, vol. 3518 de *Lecture Notes in Computer Science*, p. 661–671. Springer, 2005. ISBN 3-540-26076-5.

A. Soulet et B. Crémilleux : Extraction des top-k motifs par approximer-et-pousser. *In* M. Noirhomme-Fraiture et G. Venturini, éds : *Extraction et gestion des connaissances (EGC'2007), Actes des cinquièmes journées Extraction et Gestion des Connaissances, Namur, Belgique, 23-26 janvier 2007, 2 Volumes*, vol. RNTI-E-9 de *Revue des Nouvelles Technologies de l'Information*, p. 271–282. Cépaduès-Éditions, 2007. ISBN 978-2-85428-763-9.

A. Soulet et B. Crémilleux : Adequate condensed representations of patterns. *Data Min. Knowl. Discov.*, 17(1):94–110, 2008. ISSN 1384-5810.

J. F. Sowa : Conceptual graphs for a data base interface. *IBM Journal of Research and Development*, 20(4):336–357, 1976.

J. F. Sowa : *Knowledge Representation : Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove, CA, 1999.

R. Srikant et R. Agrawal : Mining sequential patterns : Generalizations and performance improvements. *In* M. Bouzeghoub et G. Gardarin, éds : *Advances in Database Technology - EDBT'96, 5th International Conference on Extending Database Technology, Avignon, France*, Lecture Notes in Computer Science 1057, p. 3–17. Springer, 1996.

M. Stonebraker, R. Agrawal, U. Dayal, E. J. Neuhold et A. Reuter : Dbms research at a crossroads : The vienna update. *In* R. Agrawal, S. Baker et D. A. Bell, éds : *VLDB*, p. 688–692. Morgan Kaufmann, 1993. ISBN 1-55860-152-X.

L. Szathmary, A. Napoli et S. O. Kuznetsov : Zart : A multifunctional itemset mining algorithm. *In* P. W. Eklund, J. Diatta et M. Liquiere, éds : *CLA*, vol. 331 de *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

L. Szathmary, P. Valtchev, A. Napoli et R. Godin : Constructing iceberg lattices from frequent closures using generators. *In* Boulicaut *et al.* (2008), p. 136–147. ISBN 978-3-540-88410-1.

A. Tanaka, T. Kawai, T. Matsumoto, M. Fujii, T. Takabatake, H. Okamoto et K. Funatsu : Construction of a statistical evaluation model based on molecular centrality to find retrosynthetically important bonds in organic compounds. *European Journal of Organic Chemistry*, 2008(35):5995–6007, 2008a. URL `http ://dx.doi.org/10.1002/ejoc.200800392`.

A. Tanaka, T. Kawaia, M. Fujiia, T. Matsumotoa, T. Takabatakea, H. Okamotob et K. Funatsuc : Molecular centrality for synthetic design of convergent reactions. *Tetrahedron*, 64(20):4602–4612, May 2008b.

A. Termier, M.-C. Rousset et M. Sebag : Treefinder : a first step towards xml data mining. *In ICDM* icd (2002), p. 450–457. ISBN 0-7695-1754-4.

A. Termier, M.-C. Rousset et M. Sebag : Dryade : A new approach for discovering closed frequent trees in heterogeneous tree databases. *In ICDM* icd (2004), p. 543–546. ISBN 0-7695-2142-8.

A. Termier, M.-C. Rousset, M. Sebag, K. Ohara, T. Washio et H. Motoda : Dryade-parent, an efficient and robust closed attribute tree mining algorithm. *IEEE Trans. Knowl. Data Eng.*, 20(3):300–320, 2008.

M. H. Todd : Computer-aided organic synthesis. *Chem Soc Rev.*, 34(3):247–66, 2005.

W. Tutte : *Graph Theory*. Cambridge University Press, 2001.

I. Ugi, J. Bauer, R. Baumgartner, E. Fontain, D. Forstmeyer et S. Lohberger : Computer assistance in the design of syntheses and a new generation of computer programs for the solution of chemical problems by molecular logic. *Pure and Applied Chemistry*, 60 (11):1573–1586, 1988.

I. Ugi, J. Bauer, C. Blomberger, J. Brandt, A. Dietz, E. Fontain, B. Gruber, A. v. Scholley-Pfab, A. Senff et N. Stein : Models, concepts, theories, and formal languages in chemistry and their use as a basis for computer assistance in chemistry. *Journal of Chemical Information and Computer Sciences*, 34(1):3–16, 1994. URL `http ://pubs.acs.org/doi/abs/10.1021/ci00017a001`.

J. R. Ullmann : An algorithm for subgraph isomorphism. *Journal of the ACM*, 23(1):31–42, 1976. ISSN 0004-5411.

M. van Leeuwen, J. Vreeken et A. Siebes : Compression picks item sets that matter. *In* J. Fürnkranz, T. Scheffer et M. Spiliopoulou, éds : *PKDD*, vol. 4213 de *Lecture Notes in Computer Science*, p. 585–592. Springer, 2006. ISBN 3-540-45374-1.

C. J. van Rijsbergen : *Information Retrieval*. Butterworths, 2 édn, 1979.

N. Vanetik, S. E. Shimony et E. Gudes : Support measures for graph data. *Data Min. Knowl. Discov.*, 13(2):243–260, 2006.

V. N. Vapnik : *Statistical learning theory*. Wiley, New York, 1998.

P. Vismara, P. Jambaud, C. Laurenço et J. Quinqueton : Resyn : objets, classification et raisonnement distribué en chimie organique. *In* R. Ducournau, J. Euzenat et A. Napoli, éds : *Langages à objets : état et perspectives de la recherche*, vol. 19, p. 397–419. Collection didactique de l'INRIA, 1998.

P. Vismara : *Reconnaissance et représentation d'éléments structuraux pour la description d'objets complexes. Application à l'élaboration de stratégies de synthèse en chimie organique.* Thèse d'informatique, Université des Sciences et Techniques du Languedoc, Montpellier, 1995.

P. Vismara et C. Laurenço : An abstract representation for molecular graphs. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 51:343–366, 2000.

P. Vismara, J.-C. Régin, J. Quinqueton, M. Py, C. Laurenço et L. Lapied : Resyn – un système d'aide à la conception de plans de synthèse en chimie organique. *In Actes des 12èmes Journeés Internationales Intelligence Artificielle, Systèmes Experts, Langage Naturel, Avignon*, p. 305–318, 1992.

G. E. Vladutz : Do we still need a classification of reactions ? *In* P. Willet, éd. : *Modern Approaches to Chemical Reaction Searching*, p. 202–220. Gower Publishing, 1986.

G. E. Vleduts : Concerning one system of classification and codification of organic reactions. *Inf. Stor. Retr.*, 1:117–146, 1963.

C. Wang, W. Wang, J. Pei, Y. Zhu et B. Shi : Scalable mining of large disk-based graph databases. *In* Kim *et al.* (2004), p. 316–325. ISBN 1-58113-888-1.

C. Wang, Y. Zhu, T. Wu, W. Wang et B. Shi : Constraint-based graph mining in large database. *In* Y. Zhang, K. Tanaka, J. X. Yu, S. Wang et M. Li, éds : *APWeb*, vol. 3399 de *Lecture Notes in Computer Science*, p. 133–144. Springer, 2005a. ISBN 3-540-25207-X.

J. Wang, J. Han, Y. Lu et P. Tzvetkov : Tfp : An efficient algorithm for mining top-k frequent closed itemsets. *IEEE Trans. Knowl. Data Eng.*, 17(5):652–664, 2005b.

K. Wang, Y. Jiang, J. X. Yu, G. Dong et J. Han : Pushing aggregate constraints by divide-and-approximate. *In* U. Dayal, K. Ramamritham et T. M. Vijayaraman, éds : *ICDE*, p. 291–302. IEEE Computer Society, 2003. ISBN 0-7803-7665-X.

K. Wang et H. Liu : Discovering typical structures of documents : A road map approach. *In SIGIR*, p. 146–154. ACM, 1998.

K. Wang et H. Liu : Discovering structural association of semistructured data. *IEEE Trans. Knowl. Data Eng.*, 12(3):353–371, 2000.

K. Wang, L. Wang, Q. Yuan, S. Luo, J. Yao, S. Yuan, C. Zheng et J. Brandt : Construction of a generic reaction knowledge base by reaction data mining. *Journal of Molecular Graphics and Modelling*, 19(5):427–433, October 2001.

T. Washio, E. Suzuki, K. M. Ting et A. Inokuchi, éds. *Advances in Knowledge Discovery and Data Mining, 12th Pacific-Asia Conference, PAKDD 2008, Osaka, Japan, May 20-23, 2008 Proceedings*, vol. 5012 de *Lecture Notes in Computer Science*, 2008. Springer. ISBN 978-3-540-68124-3.

D. Weininger : Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.

C. Wilcox et R. Levinson : A self-organized knowledge base for recall, design, and discovery in organic chemistry. *In* T. Pierce et B. Hohne, éds : *Artificial Intelligence Applications in Chemistry*, p. 209–230. ACS Symposium Series 306, New-York, 1986.

R. Wille : Restructuring lattice theory : An approach based on hierarchies of concepts. *Ivan Rival, (Ed.), Ordered Sets*, p. 445–470, 1982.

R. Wille : Knowledge acquisition by methods of formal concept analysis. *In* E. Diday, éd. : *Proceedings of the Conference on Data Analysis, Learning Symbolic and Numeric Knowledge, Juan-Les-Pins, France*, p. 365–380. Nova Science Publisher Inc., New-York, 1989.

W. Wipke : Computer–assisted three–dimensional synthetic analysis. *In* W. Wipke, S. Heller, R. Feldman et E. Hyde, éds : *Computer Representation and Manipulation of Chemical Information*, p. 147–174. Wiley Interscience, New-York, 1974.

M. Wörlein, T. Meinl, I. Fischer et M. Philippsen : A quantitative comparison of the subgraph miners mofa, gspan, ffsm, and gaston. *In* Jorge *et al.* (2005), p. 392–403. ISBN 3-540-29244-6.

Y. Xiao, J.-F. Yao, Z. Li et M. H. Dunham : Efficient data mining for maximal frequent subtrees. *In ICDM* icd (2003), p. 379–386. ISBN 0-7695-1978-4.

D. Xin, H. Cheng, X. Yan et J. Han : Extracting redundancy-aware top-k patterns. *In* T. Eliassi-Rad, L. H. Ungar, M. Craven et D. Gunopulos, éds : *KDD*, p. 444–453. ACM, 2006. ISBN 1-59593-339-5.

D. Xin, J. Han, X. Yan et H. Cheng : Mining compressed frequent-pattern sets. *In* K. Böhm, C. S. Jensen, L. M. Haas, M. L. Kersten, P.-Å. Larson et B. C. Ooi, éds : *Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, August 30 - September 2, 2005*, p. 709–720. ACM, 2005. ISBN 1-59593-154-6, 1-59593-177-5.

X. Yan, H. Cheng, J. Han et P. S. Yu : Mining significant graph patterns by leap search. *In* J. T.-L. Wang, éd. : *SIGMOD Conference*, p. 433–444. ACM, 2008. ISBN 978-1-60558-102-6.

X. Yan et J. Han : gspan : Graph-based substructure pattern mining. *In Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), 9-12 December 2002, Maebashi City, Japan* icd (2002), p. 721–724. ISBN 0-7695-1754-4.

X. Yan, J. Han et R. Afshar : Clospan : Mining closed sequential patterns in large databases. *In* D. Barbará et C. Kamath, éds : *SDM*. SIAM, 2003. ISBN 0-89871-545-8.

X. Yan, P. S. Yu et J. Han : Graph indexing : A frequent structure-based approach. *In* G. Weikum, A. C. König et S. Dessloch, éds : *SIGMOD Conference*, p. 335–346. ACM, 2004. ISBN 1-58113-859-8.

X. Yan, P. S. Yu et J. Han : Graph indexing based on discriminative frequent structure analysis. *ACM Trans. Database Syst.*, 30(4):960–993, 2005a.

X. Yan, X. J. Zhou et J. Han : Mining closed relational graphs with connectivity constraints. *In* R. Grossman, R. J. Bayardo et K. P. Bennett, éds : *KDD*, p. 324–333. ACM, 2005b. ISBN 1-59593-135-X.

K. Yoshida, H. Motoda et N. Indurkhya : Graph-based induction as a unified learning framework. *Journal of Applied Intelligence*, 4(4):297–328, 1994.

K. Yoshida et H. Motoda : Clip : Concept learning from inference patterns. *Artif. Intell.*, 75(1):63–92, 1995.

M. Zaki et C.-J. Hsiao : Charm : An efficient algorithm for closed itemset mining. *In* R. Grossman, J. Han, V. Kumar, H. Mannila et R. Motwani, éds : *Second SIAM International Conference on Data Mining, Arlington*, 2002.

M. J. Zaki : Scalable algorithms for association mining. *IEEE T. Knowl. Data. En.*, 12 (3):372–390, 2000.

M. J. Zaki : Spade : An efficient algorithm for mining frequent sequences. *Mach. Learn.*, 42 (1/2):31–60, 2001.

M. J. Zaki : Efficiently mining frequent trees in a forest. *In KDD*, p. 71–80. ACM, 2002. ISBN 1-58113-567-X.

M. J. Zaki : Efficiently mining frequent embedded unordered trees. *Fundam. Inform.*, 66 (1-2):33–52, 2005a.

M. J. Zaki : Efficiently mining frequent trees in a forest : Algorithms and applications. *IEEE T. Knowl. Data. En.*, 17(8):1021–1035, 2005b.

M. J. Zaki, N. Parimi, N. De, F. Gao, B. Phoophakdee, J. Urban, V. Chaoji, M. A. Hasan et S. Salem : Towards generic pattern mining. *In* B. Ganter et R. Godin, éds : *Formal Concept Analysis, Third International Conference, ICFCA 2005, Lens, France, February 14-18, 2005, Proceedings*, vol. 3403 de *Lecture Notes in Computer Science*, p. 1–20. Springer, 2005. ISBN 3-540-24525-1.

M. J. Zaki, S. Parthasarathy, M. Ogihara et W. Li : New algorithms for fast discovery of association rules. *In KDD*, p. 283–286, 1997.

Z. Zeng, J. Wang et L. Zhou : Efficient mining of minimal distinguishing subgraph patterns from graph databases. *In* Washio *et al.* (2008), p. 1062–1068. ISBN 978-3-540-68124-3.

Q.-Y. Zhang et J. Aires-de Sousa : Structure-based classification of chemical reactions without assignment of reaction centers. *Journal of Chemical Information and Modeling*, 45(6):1775–1783, 2005. ISSN 1549-9596.

A. Zhou, W. Jin, S. Zhou et Z. Tian : Incremental mining of schema for semistructured data. *In* N. Zhong et L. Zhou, éds : *PAKDD*, vol. 1574 de *Lecture Notes in Computer Science*, p. 159–168. Springer, 1999. ISBN 3-540-65866-1.

F. Zhu, X. Yan, J. Han et P. S. Yu : gprune : A constraint pushing framework for graph pattern mining. *In* Z.-H. Zhou, H. Li et Q. Yang, éds : *PAKDD*, vol. 4426 de *Lecture Notes in Computer Science*, p. 388–400. Springer, 2007. ISBN 978-3-540-71700-3.

Annexe A

Principales notations et acronymes

Symbole ou acronyme	Signification
$\|E\|$, $\|M\|$	Cardinal de l'ensemble E, longueur du motif M.
$\overline{x}$	Moyenne de la grandeur x sur un ensemble de motifs (par défaut sur $\mathcal{F}$).
$\mathcal{D}$	Ensemble des données.
$\mathcal{F}$	Ensemble des motifs fréquents.
$\mathcal{F}_l$	Ensemble des motifs fréquents de longueur l.
$(\mathcal{M}, \leq_{\mathcal{M}})$	Ordre des motifs.
$\subseteq_S$	Relation d'inclusion entre schémas de réactions.
$\subseteq_G$	Relation d'inclusion entre graphes (i.e. de sous-graphe partiel isomorphe)
E^C	Sous-ensemble complémentaire du sous-ensemble E (par rapport à un ensemble de référence)
BdR	Base de données de réactions chimiques
Schéma CMS	Schéma caractéristique de méthode de synthèse

Fig. A.1: Notation

Annexe B

Liste des MPIs fréquents pour le jeu de données Mushroom

La figure B.2 représente les 19 MPIs fréquents trouvés dans le jeu MUSHROOM, ainsi que leur score, fréquence et longueur associés. La signification des attributs est précisée sur la figure B.1.

Attr.	Signification	Valeurs et leurs significations
0	Toxicité	p = vénéneux, e = comestible
1	Forme du chapeau	b = cloche, c = conique, x = convexe, f = plat, k = bosselée, s = creuse
2	Surface du chapeau	f = fibreuse, g = rayée, y = écailleuse, s = lisse
3	Couleur du chapeau	n = brune, b = chamois, c = cannelle, g = grise, r = verte, p = rose, u = pourpre, e = red, w = blanche, y = jaune
4	Chapeau abîmé ?	t = oui, f = non
5	Odeur	a = amande, l = anisée, c = créosote, y = poissonneuse, f = fétide, m = moisie, n = aucune, p = âcre, s = épicée
6	Fixation des lamelles	a = attachées, d = tombantes, f = libres, n = encochées
7	Espace entre lamelles	c = proches, w = touffues, d = distantes
8	Épaisseur des lamelles	b = large, n = étroite
9	Couleur des lamelles	k = noire, n = brune, b = chamois, h = chocolat, g = grise, r = verte, o = orange, p = rose, u = pourpre, e = rouge, w = blanche, y = jaune
10	Forme de la tige	e = élargie, t = fuselée
11	Racine de la tige	b = bulbeuse, c = massue, u = coupe, e = égale, z = rhizomorphe, r = enracinée, ? = manquante
12	Surface de la tige au dessus de l'anneau	f = fibreuse, y = écailleuse, k = soyeuse, s = lisse
13	Surface de la tige en dessous de l'anneau	f = fibreuse, y = écailleuse, k = soyeuse, s = lisse
14	Couleur de la tige au dessus de l'anneau	n = brune, b = chamois, c = cannelle, g = grise, o = orange, p = rose, e = rouge, w = blanche, y = jaune
15	Couleur de la tige en dessous de l'anneau	n = brune, b = chamois, c = cannelle, g = grise, o = orange, p =rose, e = rouge, w = blanche, y = jaune
16	Type du voile	p = partiel, u = universel
17	Couleur du voile	n = brune, o = orange, w = blanche, y = jaune
18	Nombre d'anneaux	n = aucun, o = un, t = deux
19	Type d'anneau	c = toile d'araignée, e = évanescente, f = éclatant, l = grand, n = aucun, p = pendant, s = gainé, z = secteur
20	couleur des spores	k = noire, n = brune, b = chamois, h = chocolat, r = verte, o = orange, u = pourpre, w = blanche, y = jaune
21	Population	a = abondante, c = en bouquet, n = nombreuse, s = éparpillée, v = parsemé, y = solitaire
22	Habitat	g = herbes, l = feuilles, m = prairie, p = sentiers, u = ville, w = détritus, d = bois

FIG. B.1: Signification des attributs pour le jeu MUSHROOM.

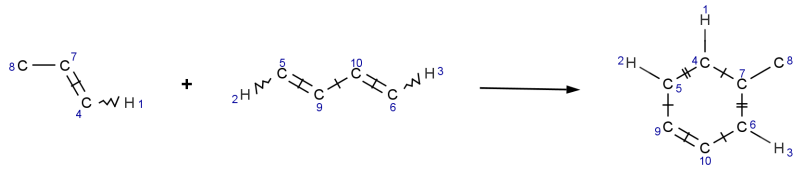
Attributs (multi-valués)																						Score	Fréq.	Long.
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22			
						f	c									p	w	o					3.86	6272	5
e				t		f	c	b				s	s			p	w	o	p				3.03	2240	11
p				f		f	c	n	b	t	?					p	w	o	e	w	v		2.98	1728	14
e				t	n	f	c	b		t	b	s	s			p	w	o	p				2.98	1728	14
e					n	f		b		t						p	w	o					2.46	2496	8
e						f		b				s				p	w	o					2.43	2816	7
p				f	f	f	c	b		e	b	k	k			p	w	o	l	h			2.39	1296	15
e					n	f		b		t		s	s			p	w	o					2.36	1920	10
				f		f				t						p	w	o	e				2.15	2496	7
p					f	f	c	b			b					p	w	o		h			1.95	1584	10
e				f	n	f	w	b		t	e			w	w	p	w	o	e				1.32	768	14
e				f	n	f	w	b						w	w	p	w						1.30	1056	10
e				f	n	f					e			w	w	p	w	o					1.06	864	10
e				t		f	c	b		e	c	s	s		w	p	w	o	p				0.95	512	15
e					n	f		b		e	?					p	w	t		w			0.59	480	10
e					n			b		e	?					p							0.50	672	6
					n	f		b		e						p	w	t					0.49	440	9
e				f	n			b		e	?					p			p				0.47	480	8
e				f	n					e						p			p				0.44	592	6

FIG. B.2: Les MPIs fréquents pour le jeu MUSHROOM et la fonction d'aire s_a (cf figure B.1 pour l'interprétation des attributs et de leurs valeurs).

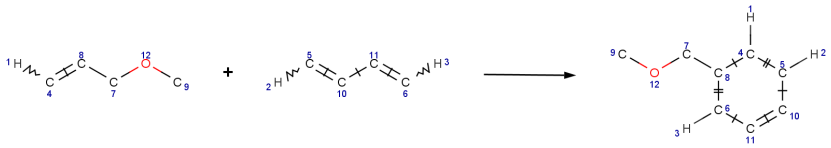
Annexe C

Extraits de résultats expérimentaux

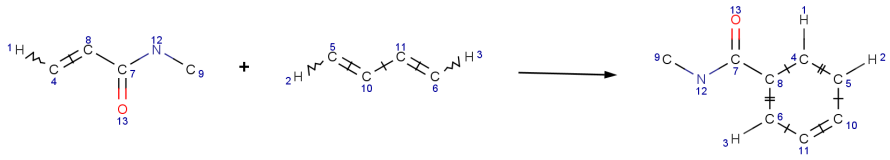
C.1 Schémas de réactions les plus informatifs (cf chapitre 5)



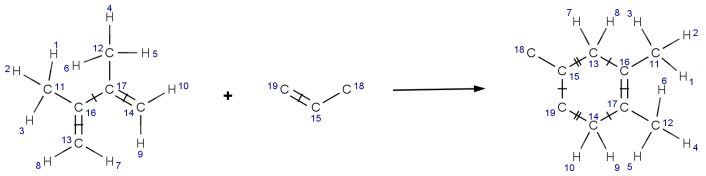
(a) MPI n° 1



(b) MPI n° 8

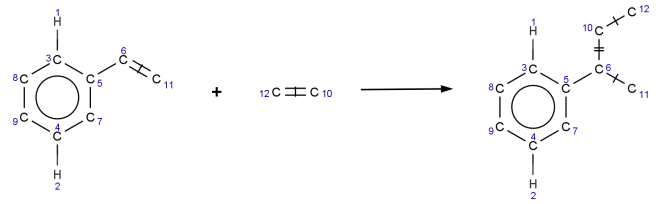


(c) MPI n° 13

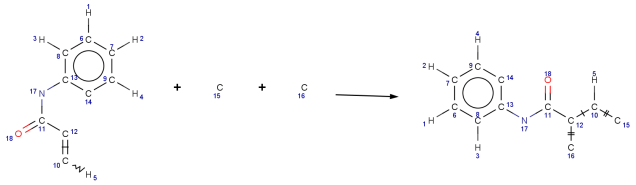


(d) MPI n° 31

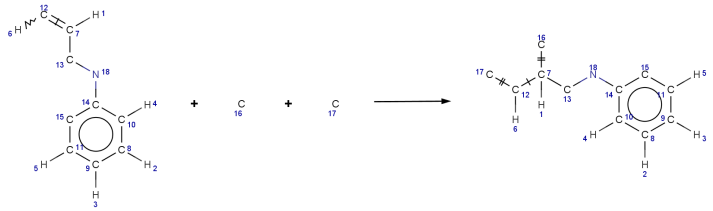
FIG. C.1: MPI sélectionnés du 1er au 4ème



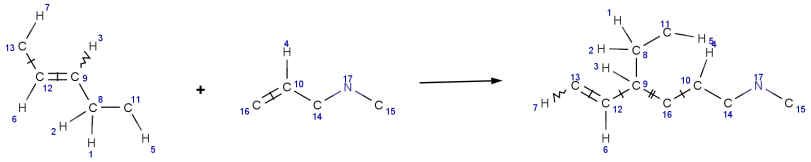
(a) MPI n° 33



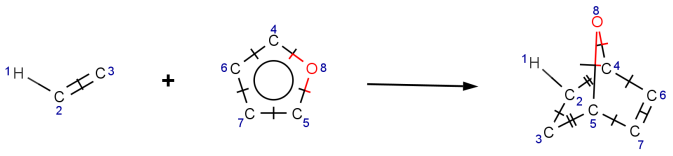
(b) MPI n° 34



(c) MPI n° 41

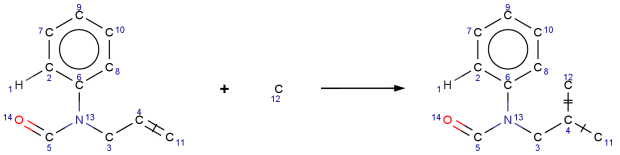


(d) MPI n° 65

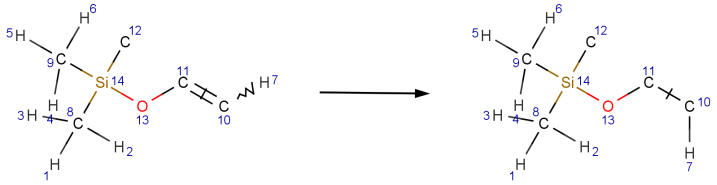


(e) MPI n° 74

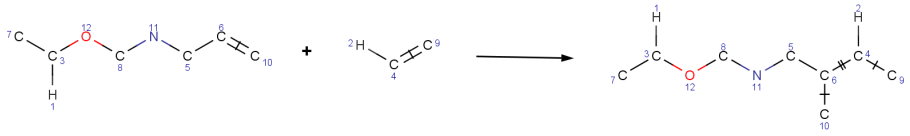
FIG. C.2: MPI sélectionnés du 5ème au 9ème



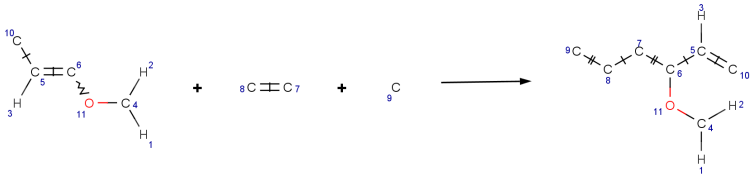
(a) MPI n° 79



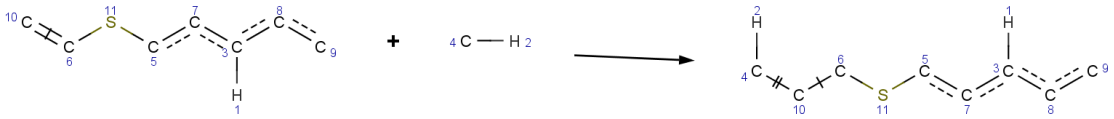
(b) MPI n° 114



(c) MPI n° 137

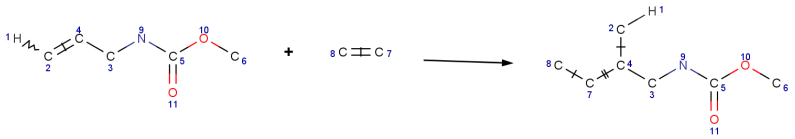


(d) MPI n° 143

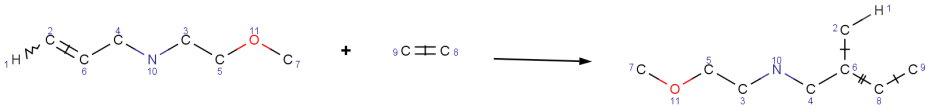


(e) MPI n° 144

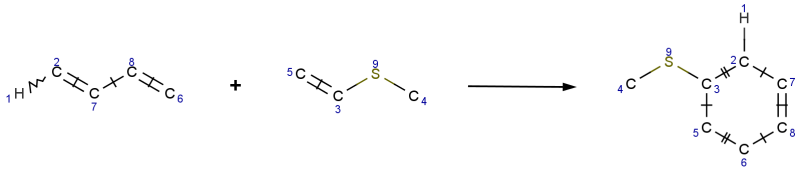
FIG. C.3: MPI sélectionnés du 10ème au 14ème



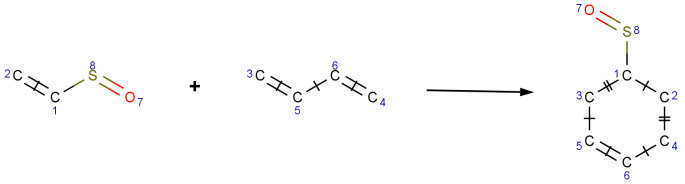
(a) MPI n° 146



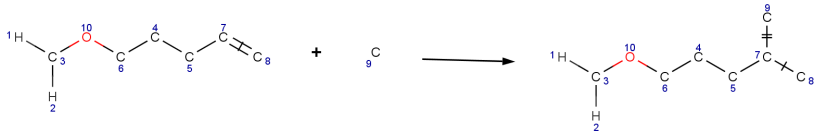
(b) MPI n° 158



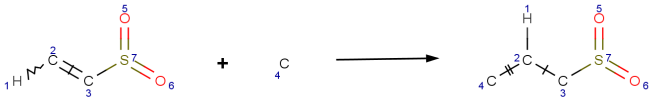
(c) MPI n° 160



(d) MPI n° 170



(e) MPI n° 182



(f) MPI n° 198

FIG. C.4: MPI sélectionnés du 15ème au 20ème

C.2 Formabilité des liaisons (cf chapitre 7)

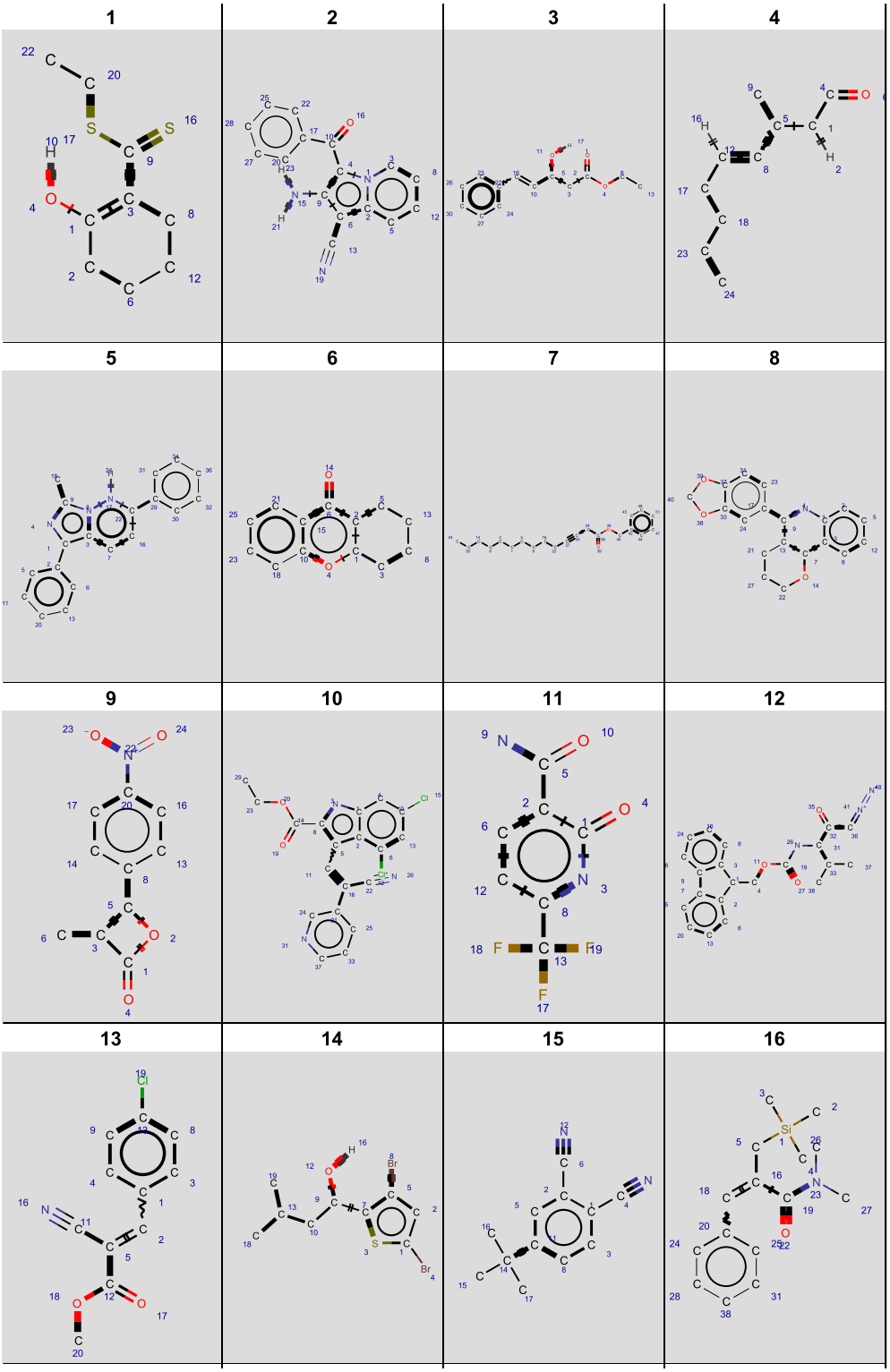


FIG. C.5: Exemples de résultats produits par `GemsBond`

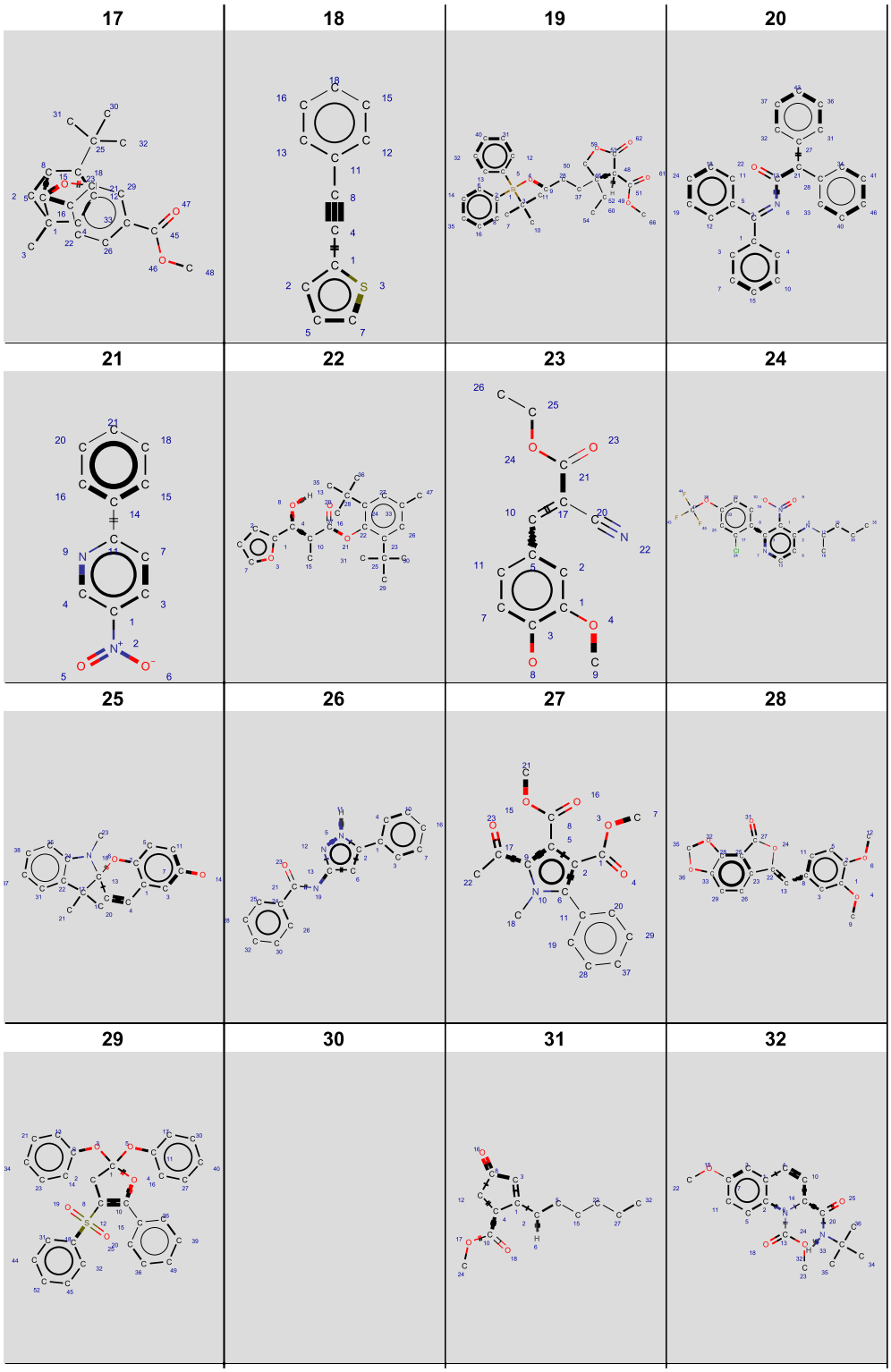


FIG. C.6: Exemples de résultats produits par `GemsBond`

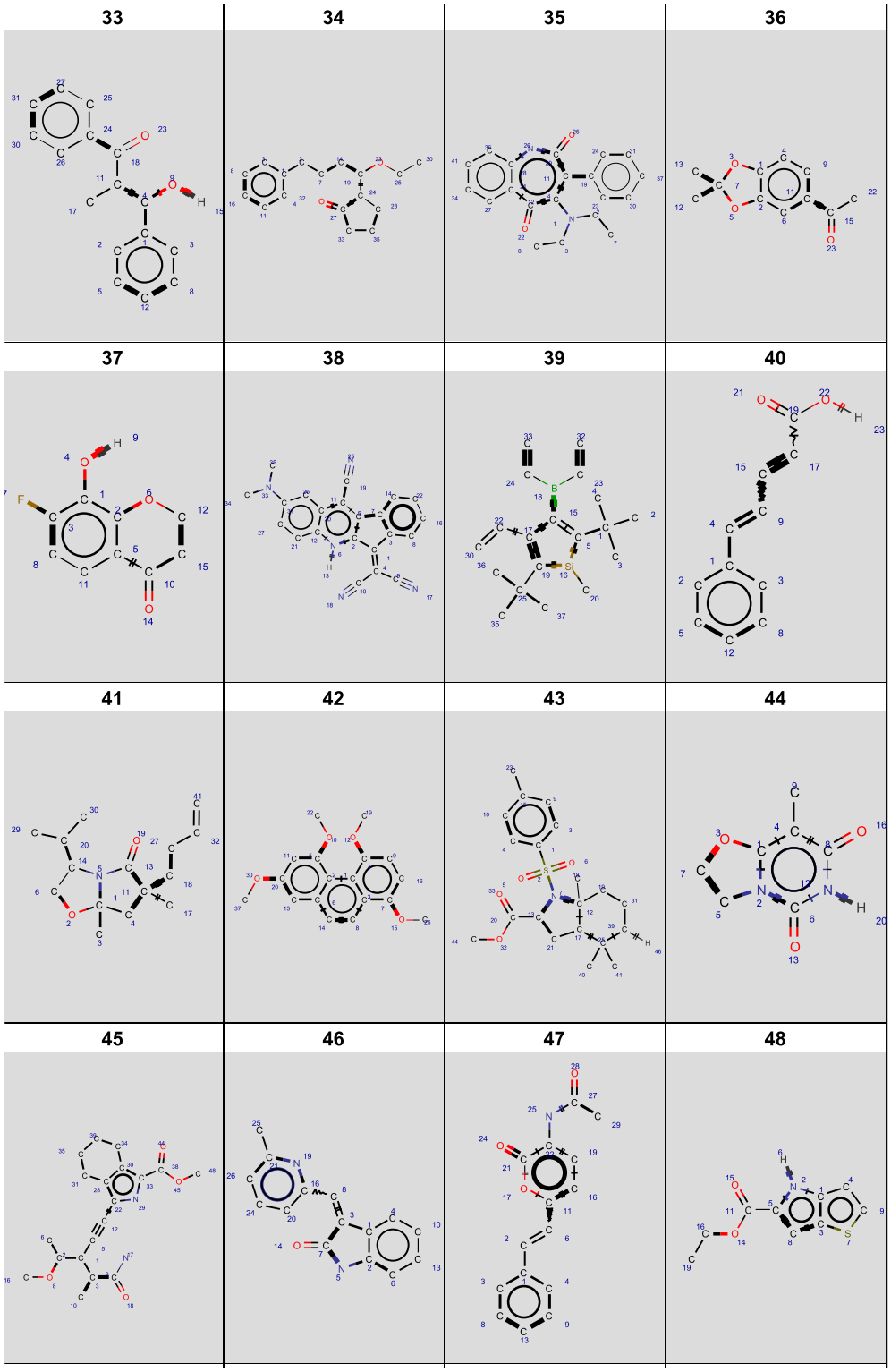


FIG. C.7: Exemples de résultats produits par `GemsBond`

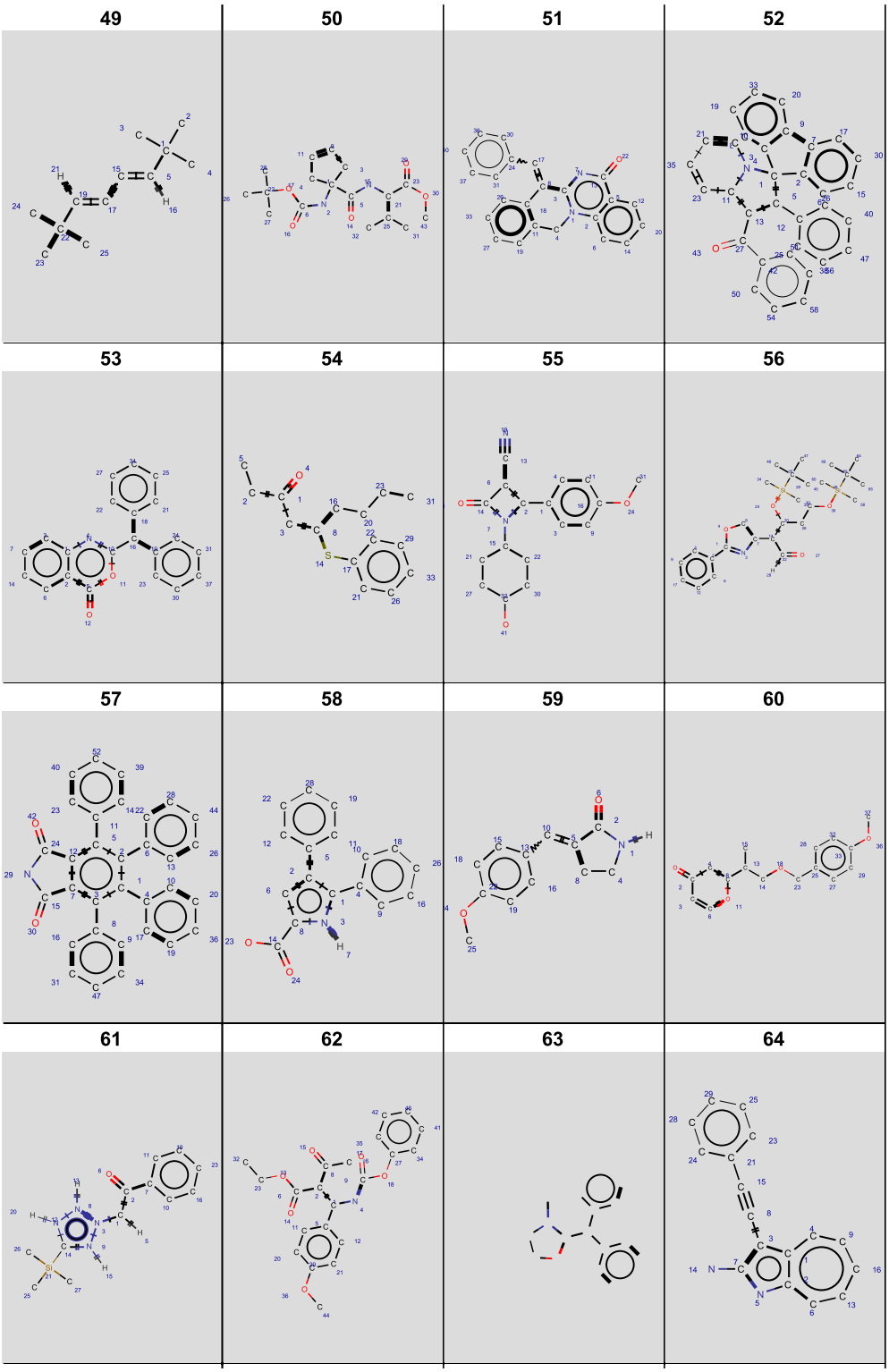


FIG. C.8: Exemples de résultats produits par `GemsBond`

Annexe D

Expertise des résultats produits par GemsBond

L'analyse des résultats produits par `GemsBond` *(cf chap. 7) réalisée par Gilles Niel, chargé de recherches au CNRS et expert de la synthèse organique, a donné lieu à la rédaction en anglais d'un rapport d'expertise dont le texte est donné ci-dessous dans son intégralité.*

As it was previously discussed, `GemsBond` produces for an input bond b its confidence $\text{conf}_G(b)$, the explanatory environment or explanation E_{max} whose confidence $\text{conf}(E_{max})$ is equal to $\text{conf}_G(b)$, and the frequency $\text{freq}(E_{max})$ of this environment. The explanation is thus the structural environment relevant to justify the formability level of bond b computed by `GemsBond` . As a chemical graph, an explanation can be directly visualized and interpreted by an organic chemist. Thus in this section, we will examine to which extent the information produced by the various explanations is chemically relevant. We will chiefly compare the explanations with either the structural objectives of synthetic methods or the retrons of the transformations corresponding to these methods while keeping in mind that `GemsBond` has no formal knowledge about chemical transformations or retrons. All along this discussion, only the formed carbon-carbon bonds of any type will be taken into account since they were selected during the early phase of the data-mining process. A first question to be solved is to determine from which confidence threshold a bond can be considered as formable enough by organic chemists. To tackle this question, compound **1** on D.1 was chosen as a prototypical example because it contains cyclic and non-cyclic parts as well as two remote functions[64] without any influence on each other. Several noteworthy points can be deduced from the results given in D.2 :

1. Most bonds of compound **1** display confidence values lower than 0.28. None of these bonds contains any functional atom[65], except for the double bond $(C21, C22)$. The explanations related to these bonds are small-sized because `GemsBond` determines very quickly that a larger environment will bring no further information to discriminate between such bonds. They have low confidences as well as high frequencies and from a

[64] We define a *function* as a connected molecular substructure which comprises exclusively carbon-carbon multiple (including aromatic) bonds and/or carbon-heteroatom bonds and/or heteroatom-heteroatom bonds.

[65] An atom is *functional* if it belongs to a function.

retrosynthetic point of view, a domain expert may consider that all these non-functional bonds should be preserved rather than disconnected. On the other hand it may be surprising to observe a low confidence for the double bond $(C21, C22)$, which is a very common structural pattern in organic synthesis. First this double bond is far from any other function present in the molecule and such an isolated double bond is generally more difficult to prepare than an activated double bond. Secondly, though double bonds are more frequently formed than single bonds, they are much less present than single bonds. Thus a direct comparison between their respective confidences is equivocal. Thirdly filtering the dataset during the selection operation with a minimal yield value of 90% introduces a bias in the results. We verified this third point by querying the ChemInform and RefLib databases through using the graph substructure of the explanation related to this double bond $(C21, C22)$ as query substructure. The topology of the double bond was fixed in a chain. Without any further constraint the proportion of hits obtained with a 90% yield is 3.7%. If the double bond is isolated, i.e. without any functional atom included for instance in an electron-withdrawing group, the the proportion of hits obtained with a 90% yield is 1.1%.

2. Three bonds of compound **1** display confidence values ranging between 0.60 and 0.78. First, bond $(C1, C17)$ shows the highest confidence while its related explanation contains more specific information, e.g. the oxygen atom $O35$, than the explanations related to bonds $(C5, C17)$ and $(C1, C5)$, which have lower confidences. `GemsBond` not only discriminates the environments of the three bonds of the cyclopropyl moiety but also assigns them higher confidences than those of non-functional bonds. This result is in agreement with the domain knowledge, especially because small-sized ring bonds are more often formed than aliphatic chain bonds. Bonds $(C16, C20)$ and $(C23, C24)$ display a confidence equal to 0.70 and 0.60, respectively, and are both located in a β-position to a double bond. We observe that bonds located in a β-position to a double bond, even non-functional ones, generally have a confidence higher than 0.35. That is, for instance, the case of bond $(C20, C34)$ whose related explanation is rather simple. We also notice that the higher the number of atoms or branched atoms present in a explanation, the higher is the confidence of the corresponding bond and the lower is the related frequency. This is illustrated by the difference between the confidences of the two bonds $(C16, C20)$ and $(C23, C24)$.
3. The ether function has a weak but significant influence on the confidences of the bonds $(C5, C6)$ and $(C6, C7)$ located in α-position. This α-effect, i.e. the increase of the confidence computed for a bond located in α-position to a functional atom, has been generally observed as discussed later in this section.

These results prompted us to propose a first ranking for formable bonds. Single bonds having confidences higher than 0.60 can be considered as easily formable while those having confidences lower than 0.30 can be considered as hardly formable. Single bonds having confidences ranging from 0.30 to 0.60 display should be moderately formable. These assessments may be refined by examining the whole results produced by our method. For each of the studied 7537 reaction products, at least one bond per product is assigned the best confidence. D.3 displays the distribution of reaction products as a function of the best confidence. We observe that only 1.7% of reaction products show a best confidence lower than 0.40 while 85.8% show a best confidence higher than 0.60. Furthermore `GemsBond` generated 312 018 explanations that correspond to the total number of bonds included in the 7537 reaction products and only 5% of these 312 018 bonds have been assigned a confidence higher than 0.60.

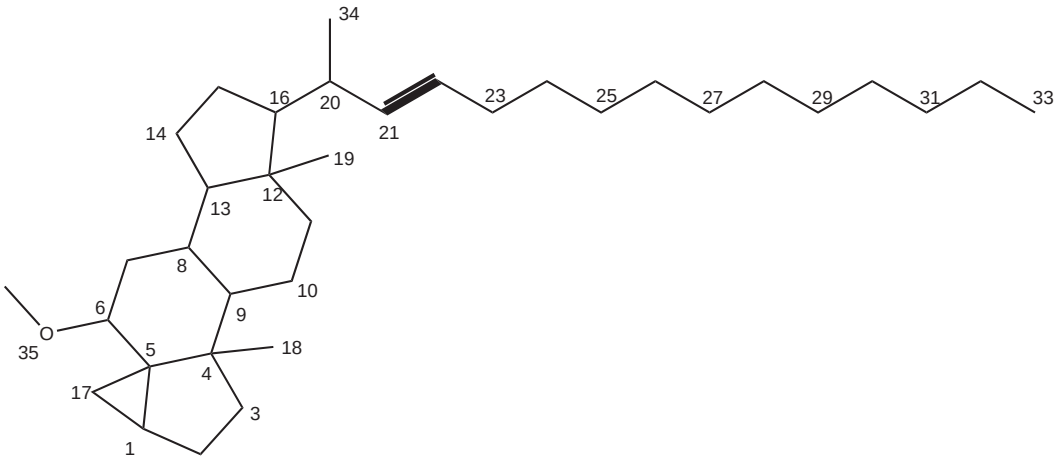


FIG. D.1: Compound **1**. Bold bonds specify the bonds formed by the reaction the molecule is a product of.

Thus a confidence threshold of 0.60 has the advantage of discriminating at least one formable bond per target molecule in 85.8% of cases among a bond subset that only represents 5% of all computed bonds.

With this confidence threshold in mind, let us to examine formable bonds to get further insights into their structural environments by studying their related explanations. The main observation lies in variable α- or β-effects induced by the presence of a functional atom in α- respectively β-position to this bond. This is exemplified in D.4 and D.5 for some usual isolated functions. The alcohol function in compound **2**, as well as the ether function in compound **3**, has a weak influence on the formability of α-bonds and no effect on that of β-bonds (D.4). The influence of the amine function in compound **4**, is stronger on the formability of α-bonds which display confidences ranging from 0.41 to 0.44. Compounds **5** and **6**, bearing a thioether and a sulfone function, respectively, show an analogous behavior with slight differences on confidences related to their α-bonds. The halogen functions present in compounds **7**, **8** and **9** lead to medium-range α- and β-effects. If the chloro- and the bromo- compounds present similar confidences, the confidence of the α-bond of compound **9** (conf = 0.15) is markedly lower than those corresponding to α-bonds of compounds **7** and **8**. Moreover the explanation related to this α-bond does not display any iodine atom and is identical to the explanation related to the $C32 - C33$ of compound **1**. In effect such primary iodo compounds are usually prepared by radical or nucleophilic substitution or by addition on a double bond, both processes that do not form any α-bond.

In D.5, more examples are given to study how confidence is affected by functional groups containing multiple bonds. In most cases a bond located in β-position to a multiple bond obtains a higher confidence than most bonds located in α-position. For instance, all compounds **10-13** display the same explanation for the β-bond whatever the nature of the carbonyl- or carboxyl- group. Among these groups, the aldehyde function is the one and only function to bring about a confidence increase of its α-bond, a fact consistent with the domain knowledge because this function is frequently introduced through a one carbon chain elongation. If we consider now the aromatic ring of compound **14**, firstly a significant β-effect (conf = 0.50) is observed, secondly all aromatic bonds show a very low confidence. This seems

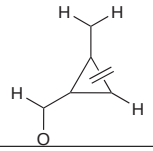
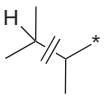
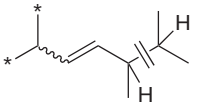
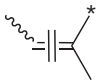
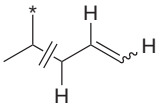
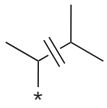
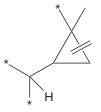
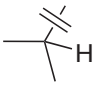
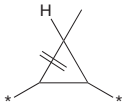
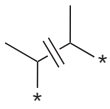
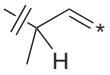
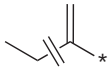
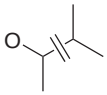
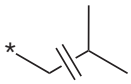
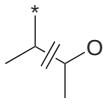
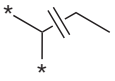
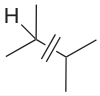
Bond(s) b	Conf. $\text{conf}_G(b)$	Explanation E_{max}	Freq. of E_{max}	Bond b	Conf. $\text{conf}_G(b)$	Explanation E_{max}	Freq. of E_{max}
$C1, C17$	0.78		12	$C7, C8$ $C15, C16$	0.27		1182
$C16, C20$	0.70		109	$C21, C22$	0.25		670
$C23, C24$	0.60		554	$C1, C2$ $C3, C4$ $C4, C5$ $C11, C12$ $C12, C13$ $C13, C14$	0.24		1770
$C5, C17$	0.46		75	$C20, C21$	0.21		1838
$C1, C5$	0.39		111	$C2, C3$ $C10, C11$ $C14, 1C5$ $C24, C25$ ... $C31, C32$	0.20		2535
$C20, C34$	0.35		1181	$C22, C23$	0.17		691
$C5, C6$	0.32		411	$C4, C18$ $C12, C19$	0.16		2479
$C6, C7$	0.29		884	$C32, C33$	0.15		3533
$C4, C9$ $C8, C9$ $C8, C13$ $C12, C16$	0.28		506				

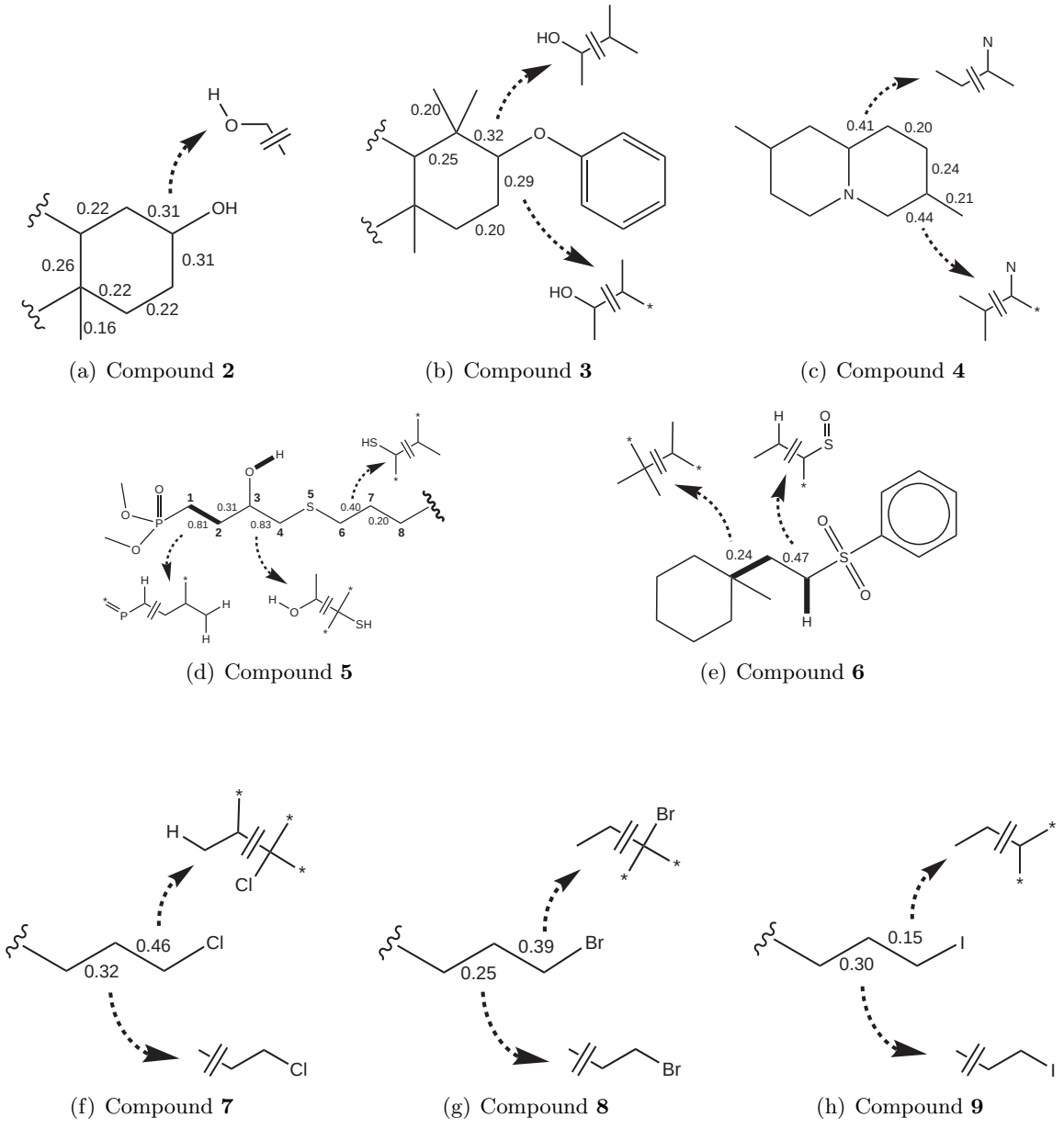
FIG. D.2: Bonds of compound **1** sorted in decreasing order of confidence. In the explanation columns, the bond the environment refers to is displayed as a formed bond (i.e with two parallel crossing strokes) ; a wiggly stroke means the bond is of undefined stereochemistry ; the asterisk denotes an atom of any type.

Best confidence	Reaction products	
	Number	Ratio
$c < 0.2$	45	0.6%
$0.2 \leq c < 0.4$	85	1.1%
$0.4 \leq c < 0.6$	946	12.6%
$0.6 \leq c < 0.8$	1727	22.9%
$0.8 \leq c < 1$	2542	33.7%
$c = 1$	2192	29.1%

FIG. D.3: Reaction product distribution as a function of the best confidence

logical because aromatic bonds are very rarely formed during a synthesis. The imine, nitrile, and acetylenic functions of derivatives 15, 16 and 17, respectively, show medium-range effects on both α- and β-bonds. We observe that the β-bond confidences of 15 varies appreciably depending on the position of the imine nitrogen atom. The two bonds located in β-position to the nitrile function of 16 have different environments and consequently display different confidences (conf = 0.51, conf = 0.64). The confidence of the α-bond to the nitrile function is also medium-range revealing a partial formability character as in the case of aldehyde 10. Finally, the acetylenic function of 17 has a higher influence on the confidence of the α-bond than on the one of the β-bond ; this information is consistent with the frequent use of synthetic methods able to form such carbon-carbon bonds. The acetylenic bond, which is usually rarely formed, shows here a low confidence as in the case of the aromatic bonds of 14. More generally, the explanations related to bonds in compounds **2-17** provide interesting pieces of information because they reveal the presence of a function responsible for either an α- or a β-effect. But no γ-effect is detected for these compounds. On the other hand, except for the easily formable bonds located in β-position to double bonds (conf $\geq 0,60$), most formable bonds of compounds **2-17** display medium-range confidence values.

If we examine now the compounds possessing at least one bond that show a confidence higher than 0.60, i.e. about 86% of studied compounds, we notice that formability of such a bond depends very often on conjugated α- and β-effects. Some representative examples are given on D.6. In compound **18**, bond $(C3, C4)$ shows a confidence equal to 0.89, a markedly higher value than previously observed for a bond in α-position to an alcohol function or in β-position to an alkene function - in compounds **2** and **5** in D.4. The explanation related to this bond includes actually both alcohol and alkene functions and involves the retron of the transformation corresponding to the addition of an allyl organometallic species to a carbonyl derivative. We observe comparable values for bonds $(C2 - C3)$ of compounds **19** and **20** whose related explanations involve the retrons of the transformations corresponding to the addition of an ester enolate to an aldehyde and to the addition of a ketone enolate to an imine, respectively. In the case of compound **21**, three bonds display a confidence higher than 0.60 : $(C2, C3)$, $(C1, C3')$, and $(C2', C3')$. The highest confidence displayed by bond $(C1, C3')$ is not surprising because to connect a carbonyl synthon to an aromatic one is a very usual process. Actually the related explanation includes both carbonyl and aromatic functions. From an organic synthesis point of view, it is quite logical to retrieve double bond $(C2, C3)$ among the formable bonds since it may result from a classic olefination reaction. More surprising is the computed confidence for the bond $(C1, C1')$ of compound **22** because no α- or β-effect can justify such a high value (conf = 1.00) by comparison with the corresponding bond of 18 and 19 (conf = 0.18). The related explanation contains both alkene and phenyl functions and

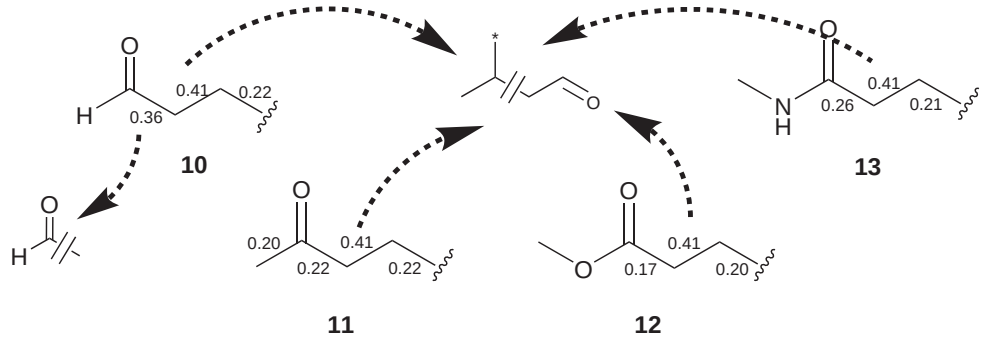


(a) Compound **2** (b) Compound **3** (c) Compound **4**

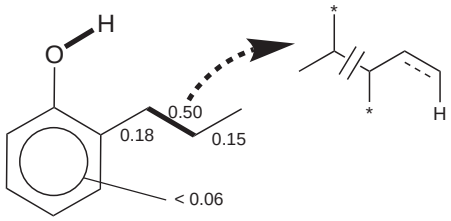
(d) Compound **5** (e) Compound **6**

(f) Compound **7** (g) Compound **8** (h) Compound **9**

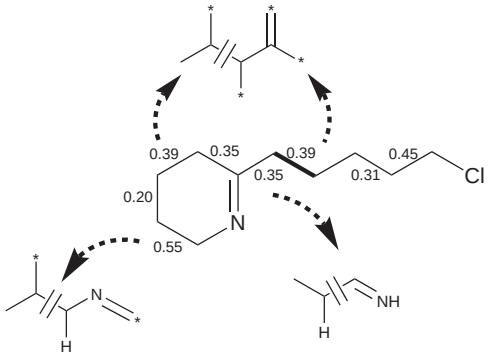
FIG. D.4: Meaningful confidences and explanations related to compounds **2-9**.



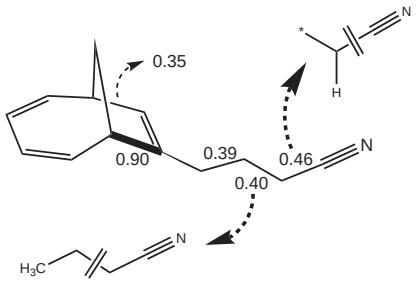
(a) Compounds **10** to **13**



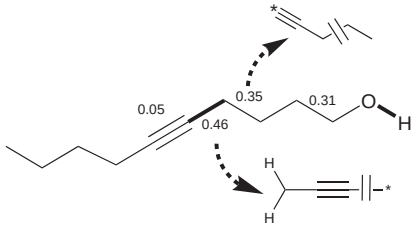
(b) Compound **14**



(c) Compound **15**

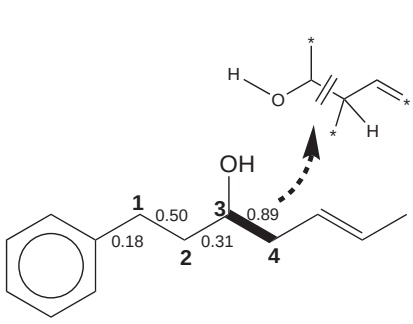


(d) Compound **16**

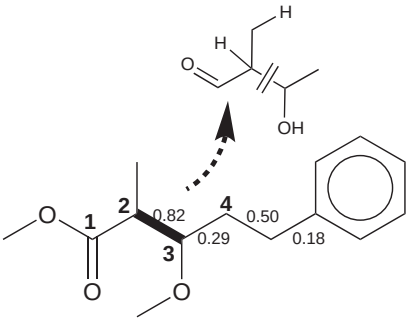


(e) Compound **17**

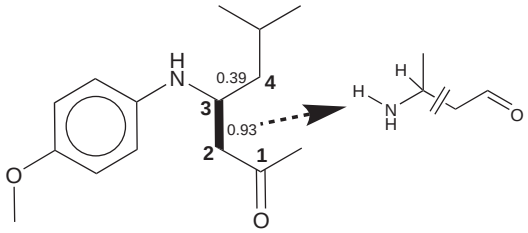
Fig. D.5: Meaningful confidences and explanations related to compounds **10-17**. Dotted lines represent aromatic bonds. Bold bonds specify the bonds formed by the reaction the molecule is a product of.



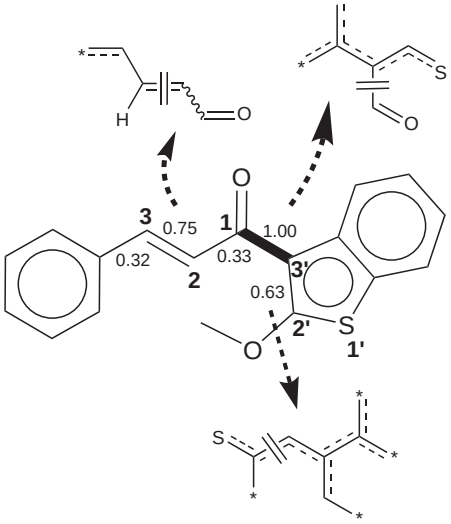
(a) Compound **18**



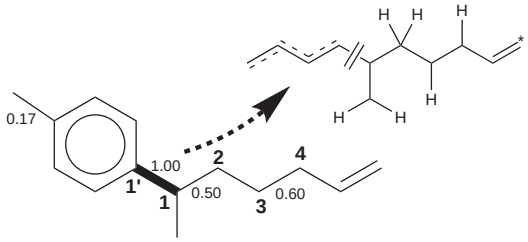
(b) Compound **19**



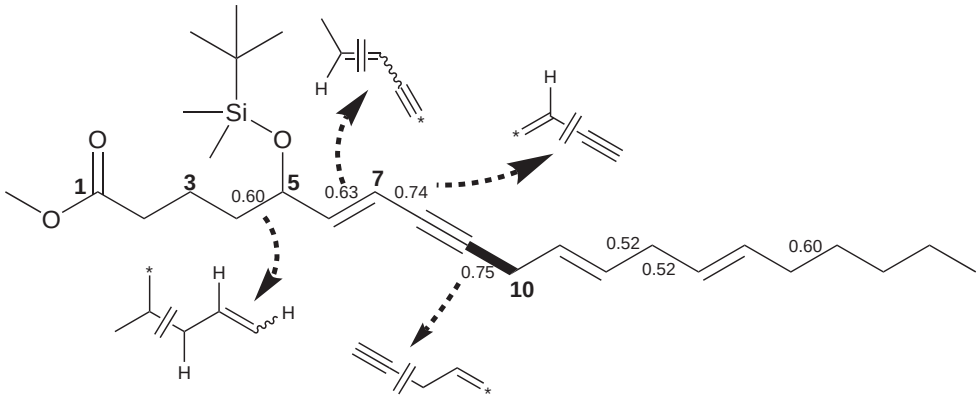
(c) Compound **20**



(d) Compound **21**



(e) Compound **22**



(f) Compound **23**

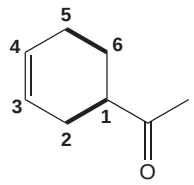
FIG. D.6: Meaningful confidences and explanations related to compounds **18-23**. Dotted lines represent aromatic bonds. Bold bonds specify the bonds formed by the reaction the molecule is a product of.

an especially high number of atoms and bonds and thus has a low frequency (freq = 8). By comparison bonds $(C1, C2)$ and $(C3, C4)$ display lower confidences (conf = 0.50, conf = 0.60) but their related explanations (freq = 866, freq = 554) contain a single function. Finally, five distinct bonds of compound **23** display a confidence higher than 0.60 and can thus be considered as formable bonds. The two best confidences (conf = 0.74, conf = 0.75) concern respectively the bonds $(C7, C8)$ and $(C9, C10)$ incident to the alkyne function. These two bonds display higher confidences than the one observed for the analogous bond of (conf = 0.46) because of the inclusion of the neighboring double bond in both explanations. In words of organic synthesis these confidences are in agreement with known $Csp-Csp^2$ cross-coupling methods.

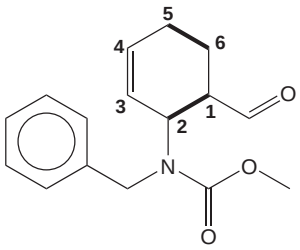
Among the synthetic methods enabling the formation of six-membered rings, the Diels-Alder reaction is probably the most famous one. We studied therefore some reaction products 24-28 resulting from this synthetic method (D.7) and especially the confidences of bonds $(C1, C2)$ and $(C5, C6)$, which are logically formed during the reaction.

We noticed in each case high confidences for these two bonds. The explanations related to the formation of bond $(C2, C2)$ include, for each studied compound, both alkene and carbonyl functions that constitute a part of the expected favorable environment for a Diels-Alder reaction. With respect to bond $(C5, C6)$, the explanations in the case of compounds **24-26** are identical to the one that was already computed for a bond located in a β-position to a double bond, see compounds **5**, **22**, **23** or the bonds $(C2, C7)$ and $(C5, C7)$ of compounds **26** and **27**. Interestingly, in the case of compounds **27** and **28**, the explanations include the retron of the Diels-Alder transformation and especially the six-membered ring. Let us remind that `GemsBond` has no a priori chemical knowledge about chemical transformations and synthetic methods and that it computes a confidence for each bond without considering the simultaneous formation of two bonds as in the case of cycloaddition reaction products. This point must be considered in future developments of our method that already discriminates the two formable bonds involved in this cycloaddition.

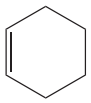
In short `GemsBond` recognizes at least one formable bond in 86% of studied bonds. It produces an explanation including in many cases the retron of a known transformation as well as the function(s) that constitute a necessary environment for the bond to be formed. This study only considers carbon-carbon formed bonds and should be extended to carbon-heteroatom bonds and/or heteroatom-heteroatom bonds in order to take into account the vast domain of heterocyclic compounds.



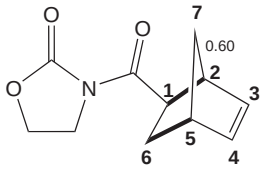
(a) Compound **24**



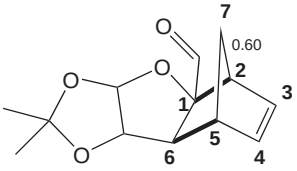
(b) Compound **25**



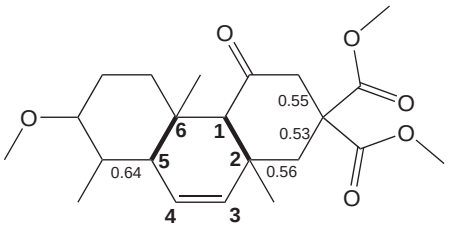
(c) Diels-Alder reaction retron



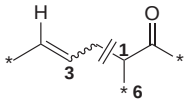
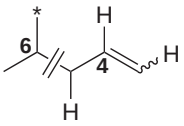
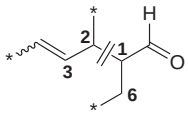
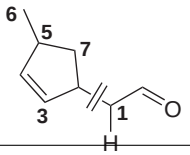
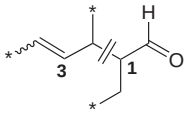
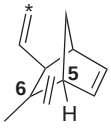
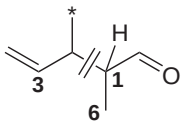
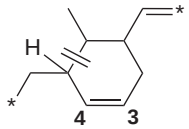
(d) Compound **26**



(e) Compound **27**



(f) Compound **28**

Compound number	$(C1, C2)$ bond Confidence	Explanation	$(C5, C6)$ bond Confidence	Explanation
24	0.69		0.60	
25	1.00		as above	
26	1.00		as above	
27	1.00		0.97	
28	0.81		0.87	

(g) Explanations of the different bonds $(C1, C2)$ and $(C5, C6)$

FIG. D.7: Meaningful confidences and explanations related to compounds **24-28**. Dotted lines represent aromatic bonds. Bold bonds specify the bonds formed by the reaction the molecule is a product of.

Résumé

Des millions de réactions chimiques sont décrites dans des bases de données sous la forme de transformations de graphes moléculaires. Cette thèse propose différentes méthodes de fouille de donnés pour extraire des motifs pertinents contenus dans ces graphes et ainsi aider les chimistes à améliorer leurs connaissances des réactions chimiques et des molécules. Ainsi on commence par montrer comment le problème central de la recherche des schémas de réactions fréquents peut se résoudre à l'aide de méthodes existantes de recherche de sous-graphes fréquents. L'introduction du modèle général des motifs les plus informatifs permet ensuite de restreindre l'analyse de ces motifs fréquents à un nombre réduit de motifs peu redondants et représentatifs des données. Si l'application du modèle aux bases de réactions permet d'identifier de grandes familles de réactions, le modèle est inadapté pour extraire les schémas caractéristiques de méthodes de synthèse (schémas CMS) dont la fréquence est trop faible. Afin de surmonter cet obstacle, est ensuite introduite une méthode de recherche heuristique fondée sur une contrainte d'intervalle entre graphes et adaptée à l'extraction de motifs de très faible fréquence. Cette méthode permet ainsi de déterminer à partir d'exemples de réactions et sous certaines conditions le schéma CMS sous-jacent à une réaction donnée. La même approche est ensuite utilisée pour traiter le problème de la classification supervisée de sommets ou d'arêtes fondée sur leurs environnements puis exploitée pour évaluer la formabilité des liaisons d'une molécule. Les résultats produits ont pu être analysés par des experts de la synthèse organique et sont très encourageants.

Mots-clés: Fouille de données, fouille de graphes, extraction sélective de motifs dans des données, recherche des motifs fréquents, classification supervisée, bases de données de réactions chimiques, chémoinformatique.

Abstract

Millions of chemical reactions are described in databases as transformations of molecular graphs. This thesis proposes different data-mining methods to extract relevant patterns included in those graphs and therefore to help chemists in improving knowledge about chemical reactions and molecules. One first shows how the central problem of searching frequent reaction patterns can be solved using existing graph-mining methods. Introducing the general model of most informative patterns then allows experts to reduce the analysis of these frequent patterns to a very small set of non-redundant patterns characteristic of data. If the application of this model to reaction database identifies large and characteristic families of reactions, the model doesn't allow in practice the extraction of reaction patterns characteristic of synthesis methods (abbr. CSM patterns) as their frequencies are far too low. In order to overcome this problem, is introduced a heuristic search algorithm based on a graph interval constraint and able to extract patterns with very low frequency. Thus this method determines from examples of chemical reactions and under some conditions the CSM pattern underlying a given input reaction. The same approach is then used to address the problem of supervised classification of vertices or edges based on their environment and then applied to evaluate formability of bonds in molecules. Experimental results have been analyzed by experts and are very encouraging.