Méthodes combinatoires de reconstruction de réseaux phylogénétiques

Philippe Gambette

ACADÉMIE DE MONTPELLIER
UNIVERSITÉ **M**ONTPELLIER **II**
Sciences et **T**echniques du **L**anguedoc

THÈSE

présentée au Laboratoire d'Informatique de Robotique
et de Microélectronique de Montpellier pour
obtenir le diplôme de doctorat

Spécialité	:	**Informatique**
Formation Doctorale	:	**Informatique**
École Doctorale	:	**Information, Structures, Systèmes**

Méthodes combinatoires de reconstruction de réseaux phylogénétiques
Combinatorial Methods for Phylogenetic Network Reconstruction

par

Philippe GAMBETTE

Soutenue le 30 novembre 2010, devant le jury composé de :

Directeur de thèse
M. Christophe PAUL, Directeur de Recherche ..... CNRS, LIRMM

Co-Directeur de thèse
M. Vincent BERRY, Professeur ..... Université Montpellier 2, LIRMM

Rapporteurs
M. Guillaume FERTIN, Professeur ..... Université de Nantes, LINA
M. Vincent MOULTON, Professeur ..... University of East Anglia

Présidente du jury
Mme Violaine PRINCE, Professeur ..... Université Montpellier 2, LIRMM

Examinateurs
M. Alain GUÉNOCHE, Directeur de Recherche ..... CNRS, IML
M. Eric TANNIER, Chargé de Recherche ..... INRIA, LBBE

Table des matières

ACADÉMIE DE MONTPELLIER
UNIVERSITÉ **M**ONTPELLIER **II**
Sciences et **T**echniques du **L**anguedoc

THÈSE

présentée au Laboratoire d'Informatique de Robotique
et de Microélectronique de Montpellier pour
obtenir le diplôme de doctorat

Spécialité	:	**Informatique**
Formation Doctorale	:	**Informatique**
École Doctorale	:	**Information, Structures, Systèmes**

Méthodes combinatoires de reconstruction de réseaux phylogénétiques
Combinatorial Methods for Phylogenetic Network Reconstruction

par

Philippe GAMBETTE

Soutenue le 30 novembre 2010, devant le jury composé de :

Directeur de thèse
M. Christophe PAUL, Directeur de Recherche ..... CNRS, LIRMM

Co-Directeur de thèse
M. Vincent BERRY, Professeur ..... Université Montpellier 2, LIRMM

Rapporteurs
M. Guillaume FERTIN, Professeur ..... Université de Nantes, LINA
M. Vincent MOULTON, Professeur ..... University of East Anglia

Présidente du jury
Mme Violaine PRINCE, Professeur ..... Université Montpellier 2, LIRMM

Examinateurs
M. Alain GUÉNOCHE, Directeur de Recherche ..... CNRS, IML
M. Eric TANNIER, Chargé de Recherche ..... INRIA, LBBE

Remerciements

Merci à mes directeurs pour ces trois années de thèse ! Grâce à Vincent et Christophe, j'ai pu compter sur une véritable équipe de co-direction complémentaire sur les domaines scientifiques, habituée au travail interdisciplinaire. Ils m'ont apporté des pistes, des outils, des techniques, mais aussi de la sérénité dans les moments de doute, l'indispensable soutien financier pour la valorisation des résultats et surtout une grande liberté de recherche et de collaborations, tout en restant très présents et disponibles pour nos travaux en commun.

Je remercie Guillaume Fertin et Vincent Moulton d'avoir accepté d'évaluer cette thèse, Alain Guénoche et Eric Tannier qui ont bien voulu être examinateurs, leur expertise en tant que références dans la communauté bioinformatique est très précieuse. Merci aussi à Violaine Prince, dont j'ai pu découvrir et apprécier pendant mon doctorat les talents de linguiste-informaticienne, compositrice, chanteuse, et présidente de jury, d'avoir également accepté de faire partie de mon jury de thèse.

Mes rencontres avec Olivier Gascuel et Michel Habib, en stage de recherche, sont à l'origine de cette thèse au LIRMM. J'ai bénéficié des meilleures conditions pour découvrir le monde de la recherche et y entrer, grâce à leurs qualités humaines et scientifiques, que j'ai retrouvées chez Vincent et Christophe.

Tous mes coauteurs m'ont énormément apporté, en partageant autant leurs techniques et leurs connaissances que leur enthousiasme et leur dynamisme à des moments clés. Merci à Daniel, Stéphane, Vincent, Christophe, Regula, Christophe, Kathi, Jean, Delphine, Hyeran, Melissa, Elsa et Constance, avec qui j'ai eu la chance de travailler. C'était aussi un privilège inouï de faire partie des équipes AlGCo et MAB du LIRMM, où tant de talents et d'humour sont réunis. Séminaires, repas et pauses café m'ont permis d'apprécier régulièrement ceux de Stéphane, d'Émeric, Daniel, Philippe, Benjamin, Alexandre, Stéphan et Marie-Catherine, et d'Anne-Muriel, Laurent, Gilles, Annie, François, Jean-François, Vincent, Alban et Éric.

Je remercie également les doctorants du LIRMM pour les bons moments partagés pendant ces trois ans, et leur participation à ma longue quête de l'exhaustivité du trombinoscope des doctorants. Je citerai particulièrement Lisa et Khalil avec qui nous avons relancé le SéminDoc. Grâce à Paola et Cécile, les préparations de projets portés au sein de l'asso Contact ont été aussi réussies que conviviales. Et c'est aussi à Paola que je dois la motivation initiale pour mon engagement de représentation des étudiants et des doctorants, à l'origine de nouveaux intérêts et de compétences que je n'aurais pas imaginé développer pendant cette thèse, avec le soutien de la Présidente de l'Université et de son équipe. Les doctorants et membres actifs de l'asso Contact, dont Cathy sa directrice, m'ont accompa-

gné dans cette aventure, et certains même au-delà (que d'erreurs et de coquilles corrigées dans ce manuscrit grâce à Pascale !).

L'appartenance à deux équipes de recherche était très enrichissante en termes de contacts scientifiques et amicaux avec des jeunes chercheurs, je regrette de n'avoir pas pu les approfondir avec tous, mais j'ai pu profiter de la présence de Binh, Jean, Anthony et Kevin chez VAG-AlGCo, et de Sam, Jean-Baka, Jean-Philippe, Sylvain, Matthieu, Mathieu, Fabio, Nicolas, Pierre, Celine et Raluca chez MAB. Et bien sûr des trois compères du bureau d'à côté : Jean-Rémy, Floréal et Benoît, qui ont supporté mes irruptions avec le sourire et toujours des solutions à mes questions, parfois avant même que je les formule ! Je n'oublie pas les collègues doctorants qui m'ont fait apprécier la vie au labo avant ma thèse, au LIRMM, au ZBIT et au LIAFA, et donné plein d'outils utiles pour la suite : Denis, Alexis et François, les deux Tobias, Christian et Daniel, Marie, Mathilde, Laura, Vincent, Michaël et Mathias. Et les amis qui m'ont fait sortir la tête de ma recherche, malgré la distance : Yun, Lisa, Pierre, Maxime, Anne-Cécile, Noémie, Alice, Valentin, Céline, Nicolas, Marc, Matthieu, Guyslain, Yiota, Sarah, Marcellin, Arnaud, Julian, Anne et Ahmed.

Merci au personnel administratif qui s'est toujours montré présent et disponible pour accompagner mon entrée dans le monde de la recherche, Marine Gaudefroy-Bergmann à Tübingen, Noëlle Delgado à Paris, Pascale Decomble, Isabelle Gouat, Elisabeth Greverie, Caroline Imbert, Bernadette Lacan, Cécile Lukasik, Laetitia Megual, Elisabeth Petiot, Martine Péridier, Nadine Tilloy et Caroline Ycre à Montpellier. En soutenant des projets et missions, l'école doctorale I2S a fait plus qu'assurer ma formation doctorale, j'en remercie les responsables Christophe Dony et Marc Herzlich.

Le département informatique de la Faculté des Sciences de l'Université Montpellier 2 m'a offert un premier contact direct avec l'enseignement face aux étudiants. J'ai pu compter sur Philippe Janssen, co-bureau et tuteur de monitorat pour répondre à toutes mes questions sur divers aspects de l'enseignement. Ce fut très agréable de travailler à ses côtés et bénéficier de son expérience, comme avec Anne-Muriel, Sèverine, Stéphane, Thérèse, Michel, Pierre et Jean-François, et l'équipe de RezUFR.

Ma pratique de l'informatique est passée par l'apprentissage de divers langages, et je retourne aux sources pour remercier respectivement et chronologiquement ma maman, Patrick Sensi, Emmanuel Monnet, Franck Taïeb, Daniel Huson, et Pierre Pompidor de m'avoir appris ou permis d'apprendre le Basic Casio 6500 G, Pascal Delphi, HTML, CaML, Java, et Python.

Et en dehors de l'informatique, pendant ces dernières années, c'est Delphine qui m'a le plus appris. Merci pour toutes ces découvertes, que la longueur d'une thèse ne suffirait pas à rappeler, et dont ce paragraphe peine à décrire l'intensité et la diversité.

Merci enfin à ma famille pour son soutien et ses encouragements depuis toujours, et les conditions de travail idéales qu'elle a su m'offrir.

Préambule

Introduction

Les arbres phylogénétiques

L'utilisation des arbres et des réseaux comme moyens de classification remonte bien avant la formalisation mathématique de ces objets. Dès l'Antiquité, après les tentatives de classification des animaux par Aristote et des végétaux par Théophraste, Porphyre a proposé une classification arborée des "substances", divisées en "esprits" et "corps", ces derniers étant divisés en "minéraux" et "êtres vivants", etc. (cf. figure 0.1).

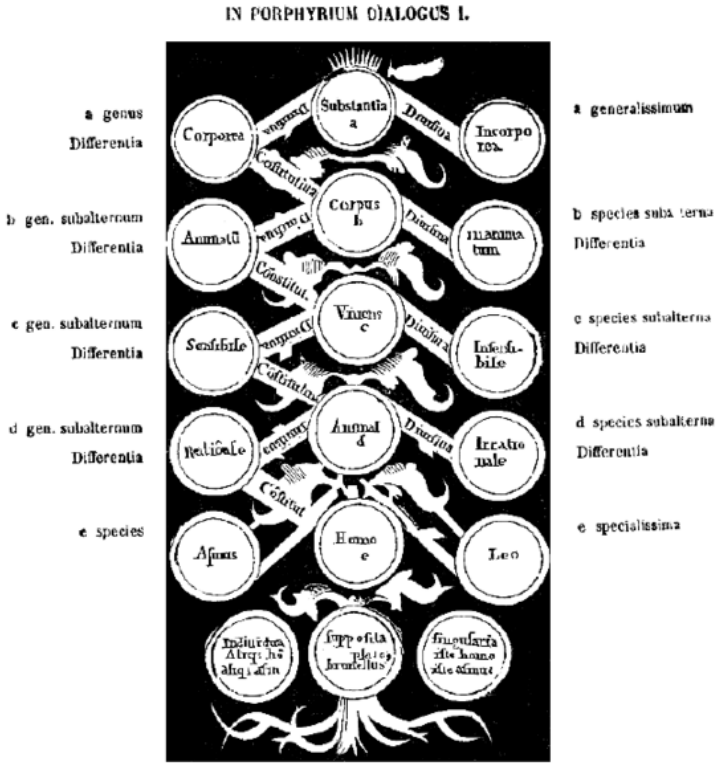
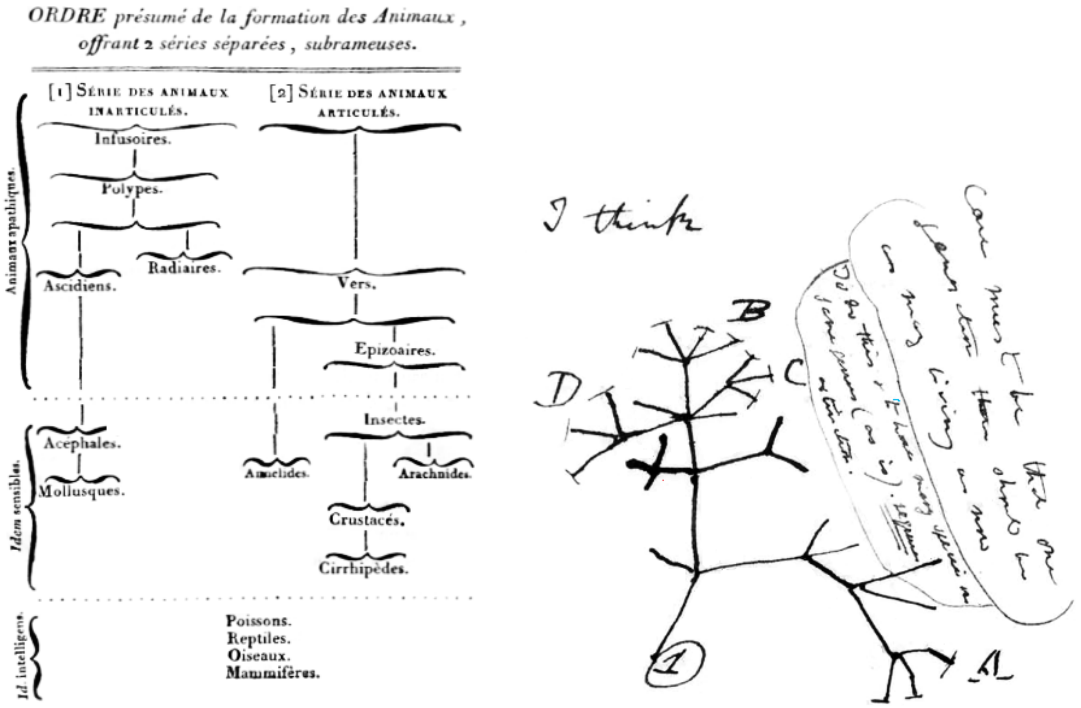


FIGURE 0.1 : L'arbre de Porphyre (III^e siècle), dans la traduction latine de son *Isagogè* par Boèce (VI^e siècle).

L'intérêt pour l'Histoire Naturelle et la classification du vivant se développe particulièrement à l'âge classique (XVII^e-XVIII^e siècles) [Foucault, 1966], et l'arbre y est utilisé comme moyen d'organiser les espèces en fonction de caractères communs comme en figure 0.2(a), bien avant la proposition de Darwin de l'utiliser comme modèle de l'évolution selon les mécanismes de la sélection naturelle [Ragan, 2009], illustrée en figure 0.2(b). Ces



(a) (b)

FIGURE 0.2 : Un arbre de Lamarck (a) extrait de l'*Histoire naturelle des animaux sans vertèbres* (1815), et l'arbre "I think" (b) du *Carnet B* de Darwin (1837).

arbres ont une racine correspondant au concept le plus général dans le cas d'une **hiérarchie** de concepts, ou à un ancêtre commun à toutes les espèces dans le cas de l'**arbre de la vie**.

Chacun de ces concepts est usuellement appelé **taxon** dans le domaine de la classification, où il correspond à un groupe d'organismes partageant certaines propriétés communes. Dans la suite, nous utiliserons ce terme pour désigner plus généralement tout concept ou individu représenté par un embranchement ou une feuille de l'arbre, que ce soit une espèce, une famille, un gène, un individu, etc.

La reconstruction d'arbres évolutifs, objectif de la **Phylogénie**, a plusieurs intérêts : historiques (découvrir le passé et dater les étapes de l'évolution des êtres vivants), épidémiologiques (identifier la souche d'un virus pour étudier sa transmission), médicaux (déter-

miner le traitement à utiliser pour combattre un pathogène en se référant aux traitements connus pour des pathogènes proches dans l'arbre), écologiques (suivre et protéger la biodiversité)... Il est important de garder à l'esprit que ces arbres ne constituent qu'un modèle mathématique permettant de décrire l'évolution de la façon la plus utile pour le problème qui nous intéresse. Dans le cas d'un **arbre des espèces** par exemple, chacune de ces espèces correspond à une population d'individus, et chaque embranchement correspond à une **spéciation**, c'est-à-dire une séparation en deux nouvelles espèces distinctes, et donc en deux nouvelles populations. Ce modèle illustré en figure 0.3(a) représente en fait de manière simplifiée la **tokogénie** de la figure 0.3(b), c'est-à-dire l'ensemble des relations de parenté entre les individus et leurs ancêtres [Hennig, 1966]. Dans le cas où chaque individu a un seul parent (par exemple dans le cas de la reproduction par division binaire), cette tokogénie a également une structure d'arbre, qui nous permet de retracer l'histoire des gènes d'individus actuels (l'**arbre des gènes**).

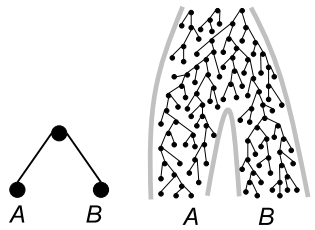


FIGURE 0.3 : Un arbre modélisant une spéciation d'une espèce se divisant en deux sous-espèces A et B (a), et la tokogénie correspondante (b) dans le cas de reproduction asexuée sans échange de matériel génétique, où chaque point représente un individu et les deux points liés en dessous de lui sont les deux clones auxquels il donne naissance.

Toutefois ces relations de parenté se compliquent dans le cas de la reproduction sexuée où un individu a deux parents de la même espèce : la tokogénie n'est alors plus un arbre mais un **réseau**, appelé **graphe de recombinaison ancestral** ou **pedigree**, illustré en figure 0.4(i), qui décrit les relations de parenté entre individus et illustre la **recombinaison** du matériel génétique. Cela n'empêche pas de choisir un modèle d'arbre pour représenter l'aspect global de ce réseau du point de vue des espèces.

Les réseaux phylogénétiques

Il arrive cependant que certains transferts de matériel génétique aient lieu entre individus de deux espèces coexistantes différentes. Même si les hybrides sont généralement stériles pour les mammifères, il est fréquent que des poissons [Hubbs, 1955], et surtout des plantes [Grant, 1971], de deux espèces différentes, aient une descendance commune fertile, qui donne naissance à une nouvelle espèce par **hybridation**.

Les transferts de matériel génétique entre individus d'espèces coexistantes sont également causés par d'autres mécanismes biologiques, en particulier le **transfert horizontal** chez les bactéries : soit par transmission directe d'une bactérie à une autre (**conjugaison**), soit par l'intermédiaire de l'environnement (**transformation**) ou d'un virus (**transduction**).

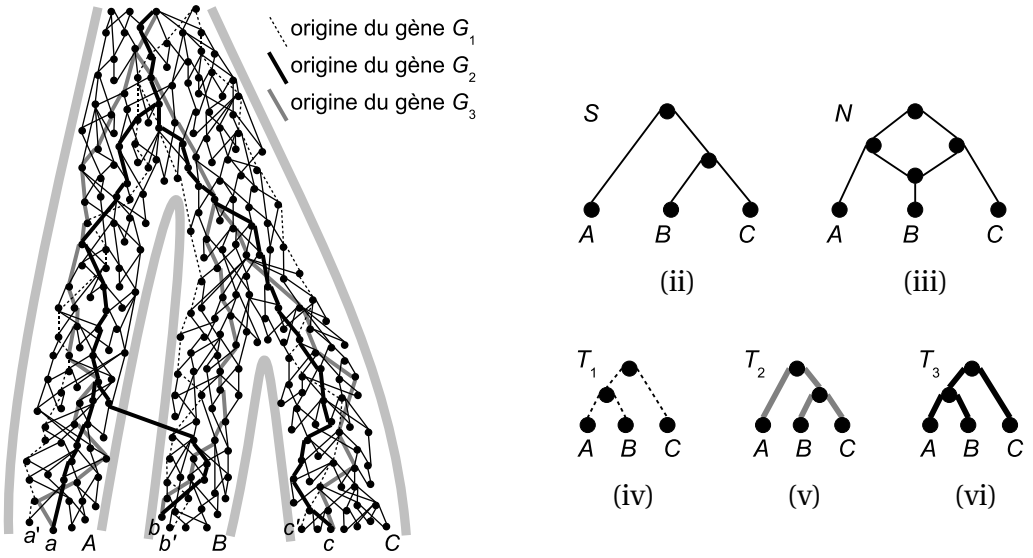


FIGURE 0.4 : Un graphe de recombinaison ancestral (tokogénie) où chaque point représente un individu et les deux points liés au-dessus de lui sont ses deux parents (i). On peut représenter cette histoire évolutive de manière simplifiée par l'arbre des espèces S (ii) ou par le réseau phylogénétique N (iii). En reconstituant l'histoire d'un gène G_2 (en gras gris), présent chez les individus a', b' et c' des espèces A, B et C, il est possible que l'on obtienne la même configuration que l'arbre des espèces S avec l'arbre T_2 (v). Cependant, l'hybridation montrée dans le graphe de recombinaison ancestral peut induire, pour un autre gène G_3 (en gras noir), une topologie T_3 différente (vi). Un autre événement biologique, le **tri de lignées**, c'est-à-dire la coexistence de deux versions d'un gène dans une espèce, peut lui aussi conduire à ce que l'histoire d'un gène G_1 (en pointillés) corresponde à un arbre T_1 lui aussi différent de S (iv).

Quel que soit le processus biologique impliqué, il aboutit à la transmission d'un gène, ou d'un ensemble de gènes, d'une branche à l'autre de l'arbre des espèces. La question se pose alors de savoir si l'on doit continuer à considérer l'arbre comme modèle de l'évolution des espèces, comme en figure 0.4(ii) [1], ou lui préférer un **réseau phylogénétique**, illustré

1. Des définitions moins évidentes de l'arbre des espèces correspondant à une tokogénie donnée, fondées sur des critères combinatoires, ont été récemment proposées par Dress *et al.* [2010].

en figure 0.4(iii). Quel que soit le modèle choisi pour la phylogénie, en particulier si c'est un arbre, il faudra garder à l'esprit, que les arbres de gènes peuvent être incompatibles avec l'arbre des espèces, comme montré en figure 0.4(iv-vi). Le modèle de réseau, plus riche, permet de stocker et visualiser ces informations d'incompatibilité.

Ainsi, l'arbre comme modèle d'évolution peut, dans certains cas, être remplacé par un réseau. La même question se pose pour l'arbre comme modèle de classification. En effet, lorsqu'on essaie de classer des individus en fonction d'états partagés pour des **caractères** [Darlu et Tassy, 1993], ces caractères ne sont pas toujours compatibles avec le modèle d'arbre, d'une part pour les raisons biologiques évoquées ci-dessus (un individu issu d'une hybridation aura des caractères des deux espèces de ses parents), mais également pour d'autres raisons biologiques, comme la perte d'un gène, ou les **homoplasies** (par **convergence évolutive**, c'est-à-dire que deux gènes évoluent indépendamment de la même manière, ou par **réversion**, c'est-à-dire que le gène revient à un état précédent).

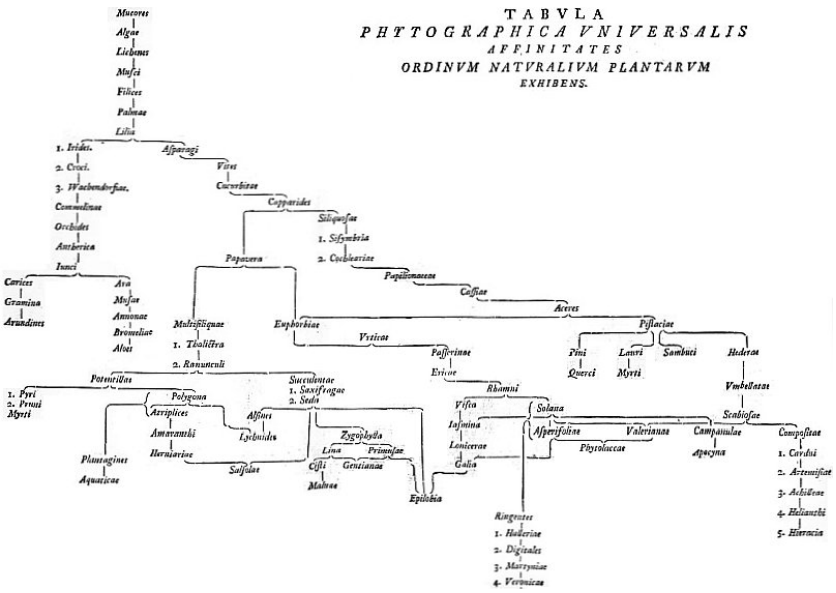
Pour représenter ces phénomènes dans une classification, il faut autoriser des groupes de taxons non seulement inclus ou disjoints (comme c'est le cas dans un arbre), mais aussi chevauchants, ce qui donne naissance à plusieurs types de réseaux phylogénétiques, comme ceux de la figure 0.5 [Ragan, 2009], ou encore les diverses généralisations des hiérarchies étudiées par la communauté des chercheurs en classification depuis les années 80 (hiérarchies faibles, pyramides, quasi-hiérarchies, etc.). Même si l'on suppose que ces réseaux reflètent des relations de parenté, on ne peut interpréter tous leurs embranchements comme des événements biologiques. Ce sont donc des **réseaux phylogénétiques abstraits**, que la terminologie distingue des **réseaux phylogénétiques explicites**, associés à un modèle d'évolution biologique précis.

Problématiques

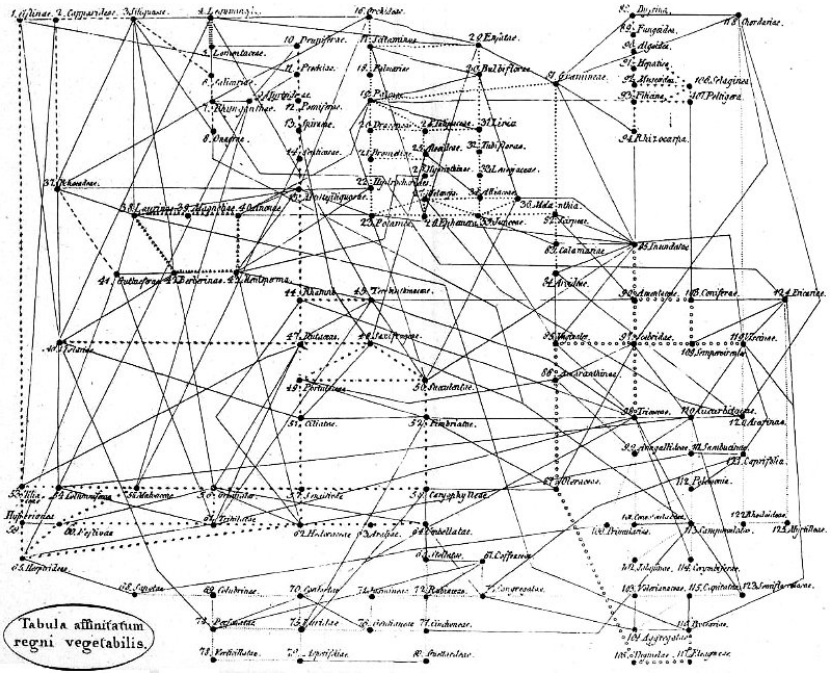
Les réseaux phylogénétiques et leurs méthodes de reconstruction constituent une thématique de recherche assez récente, qui a vraiment pris son essor à partir des années 2000, comme illustré en figure 0.6.

Un foisonnement d'idées nouvelles a donné lieu à de nombreuses approches de nature très diverse : **géométriques** (fondées sur l'étude des distances entre feuilles de l'arbre), **statistiques** (qui nécessitent de définir un modèle d'évolution), **combinatoires** (qui étudient des ensembles finis et discrets d'éléments du réseau)...

C'est sur les méthodes combinatoires que nous allons nous concentrer. Rappelons que ces méthodes, et notamment celles qui font appel à des outils de théorie ou d'algorithmique des graphes, sont souvent utilisées pour résoudre des problèmes bioinformatiques (voir par exemple [Setubal et Meidanis, 1997; Junker et Schreiber, 2008; Fertin *et al.*, 2009]). En phylogénie, comme elles prennent en entrée des collections d'éléments (par exemple, des arbres) qui doivent être présents dans le réseau en sortie, les volumes de données à traiter sont bien moins importants que pour les méthodes qui fonctionnent directement à



(a)



(b)

FIGURE 0.5 : Deux réseaux phylogénétiques, le premier extrait d'*Ordines naturales plantarum commentatio botanica* de Johann Rühling, publié en 1774 (a), et le second extrait de *Tabula affinitatum regni vegetabilis* d'August Johann Georg Carl Batsch, publié en 1802 (b).

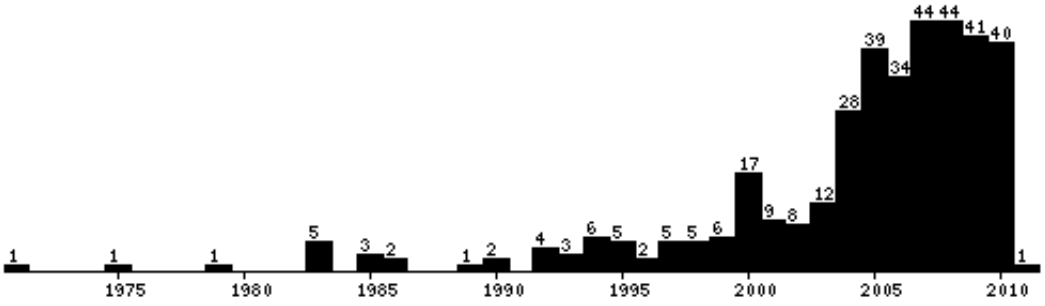


FIGURE 0.6 : Évolution du nombre de publications sur la théorie des réseaux phylogénétiques [Gambette, 2010].

partir des séquences du génome. Dans un contexte d'explosion du nombre des séquences disponibles, l'approche qui consiste à commencer par calculer un arbre, pour chaque gène des espèces dont on veut retracer la phylogénie, puis à construire le réseau à partir de ces arbres nous a paru plus efficace et rapide que l'approche qui tenterait de reconstruire directement le réseau à partir de l'ensemble des données de séquences correspondantes. En outre, de plus en plus de bases de données d'arbres de gènes sont disponibles, avec en particulier la reconstruction des premiers **phylomes** [Huerta-Cepas *et al.*, 2007], constitués des arbres de tous les gènes d'une espèce donnée au sein d'un ensemble de taxons de référence.

Au début de ce travail de thèse, plusieurs méthodes combinatoires existaient déjà, qui, à partir de divers types de données en entrée, fournissaient divers types de réseaux phylogénétiques en sortie. Parfois développées de manière indépendante par des équipes de recherche différentes, les relations entre ces méthodes et les sous-classes de réseaux qu'elles reconstruisent n'étaient pas toujours évidentes. S'imposait alors la nécessité de *trouver des liens entre ces méthodes*, et de *clarifier les propriétés et relations des éléments mathématiques étudiés*.

Un autre objectif naturel était de *développer de nouvelles méthodes combinatoires* : d'une part des *méthodes théoriques* qui aident à comprendre les propriétés mathématiques des objets étudiés, d'autre part des *méthodes pratiques* qui ont recours à l'ensemble des outils existants en combinatoire et plus spécifiquement en théorie des graphes, pour y puiser de quoi résoudre efficacement les problèmes de reconstruction de réseaux phylogénétiques.

Enfin, il s'agissait également de *discuter de la pertinence et de la fiabilité de ces méthodes* en précisant *leurs conditions d'utilisation et leurs limites*, et en les *confrontant à des données réelles*.

Plan de la thèse

Le fil directeur suivi dans cette thèse conduit de la présentation des objets utilisés, à l'examen des méthodes et des algorithmes qui les manipulent et, finalement, à l'implémentation et l'utilisation pratique de ces méthodes. Ainsi, l'état de l'art de ces trois étapes sera distribué dans chacun des chapitres correspondants. Les résultats existants seront clairement identifiés par les références afférentes. En l'absence de mention contraire, les théorèmes, lemmes, propositions et corollaires de ce manuscrit, accompagnés de démonstrations, sont des contributions de cette thèse. Plus précisément, les contributions principales apportées par ce travail s'articulent de la manière suivante.

Nous commençons par préciser au début du chapitre 1 le ***vocabulaire*** de la théorie des graphes qui nous sera utile tout au long de cette thèse, puis nous définissons les deux structures au coeur de notre étude que sont les arbres et réseaux phylogénétiques, ainsi que les divers objets combinatoires qu'on peut en extraire. Cette synthèse nous permet de faire émerger quelques premières propriétés intéressantes des objets considérés. Nous discutons également de leur contexte d'utilisation.

Puis, en section 1.4, nous montrons quelques propriétés de ***structure des réseaux de niveau*** k [Gambette *et al.*, 2009]. Nous introduisons une nouvelle classe de réseaux phylogénétiques, les réseaux non enracinés de niveau k, sur lesquels nous donnons également quelques propriétés [Gambette *et al.*, 2010]. Nous prouvons alors des relations d'inclusion ou d'équivalence entre des classes de réseaux phylogénétiques explicites et abstraits [Gambette et Huber, 2010; Gambette *et al.*, 2010], avant de synthétiser l'ensemble des relations connues entre sous-classes de réseaux phylogénétiques.

Le chapitre 2 est consacré à l'***algorithmique de la reconstruction des réseaux phylogénétiques***. Après un bref état de l'art des méthodes combinatoires de reconstruction de réseaux, nous justifions notre intérêt pour la reconstruction de réseaux phylogénétiques explicites à partir de clades, triplets et quadruplets. Nous résumons tout d'abord les résultats existants sur les triplets, puis montrons comment en généraliser certains dans un contexte non enraciné pour une reconstruction à partir de quadruplets [Gambette *et al.*, 2010]. Ces premiers algorithmes de reconstruction de réseaux phylogénétiques explicites à partir de quadruplets ayant pour l'instant principalement un intérêt théorique, nous présentons également une méthode pratique de reconstruction d'un réseau à une couche de réticulation à partir de clades [Huson *et al.*, 2009]. En effet, bien que fondée sur la résolution de deux problèmes NP-complets, elle propose pour les résoudre des algorithmes exacts très efficaces en pratique.

Le chapitre 3 présente des ***limites d'utilisation des méthodes combinatoires***. En évoquant tout d'abord le problème du bruit et du silence dans les données, nous montrons comment prendre en compte les données erronées dans les méthodes combinatoires [Gambette *et al.*, 2008], et rappelons les méthodes existantes pour inférer des données manquantes dans le cas de la reconstruction à partir de clades. Nous donnons également des éléments concrets sur l'explosion combinatoire en fonction du niveau, pour les

réseaux phylogénétiques de niveau borné [Gambette *et al.*, 2009]. Dernière limite des méthodes combinatoires : leur fiabilité. Nous montrons que même avec un ensemble complet et correct de données en entrées, il peut y avoir une incertitude sur les résultats fournis par les algorithmes de reconstruction à partir de triplets car plusieurs réseaux tout aussi parcimonieux sont solutions [Gambette et Huber, 2010].

Nous confrontons finalement au chapitre 4 les méthodes combinatoires aux ***données réelles***, en montrant tout d'abord comment choisir la méthode de reconstruction appropriée en fonction des données, grâce à une bibliographie interactive en ligne [Gambette, 2010]. Puis nous abordons le problème de la sélection des données, qui pose de nouvelles questions algorithmiques. Nous proposons de le résoudre par une approche semi-automatique basée sur un nouvel outil de visualisation, le nuage arboré [Gambette et Véronis, 2010]. Nous donnons alors des exemples de réseaux obtenus par les méthodes combinatoires de reconstruction en présentant au passage les outils logiciels utilisés.

En conclusion, nous évoquons quelques problèmes ouverts, et montrons comment intégrer au mieux les méthodes combinatoires, en prenant en compte leurs limites, au sein d'une démarche de reconstruction de réseaux phylogénétiques fiables.

Publications issues de cette thèse

Cette thèse a donné lieu aux publications suivantes, publiées, à paraître, en cours d'acceptation, ou en préparation. Nous évoquons brièvement à la fin de l'annexe d'autres travaux de recherche menés pendant cette thèse, mais sur des thématiques distinctes, qui ont donné lieu à des publications.

Les propriétés de la section 1.4.3 sur la structure des réseaux de niveau k, et l'analyse de la section 3.2 sur l'explosion de complexité en fonction du niveau ont été présentées à la conférence CPM en 2009 :

- [Gambette *et al.*, 2009] Philippe Gambette, Vincent Berry & Christophe Paul : The Structure of Level-k Phylogenetic Networks, *Proceedings of the twentieth Annual Symposium on Combinatorial Pattern Matching (CPM'09), LNCS* 5577, p. 289-300, 2010.

Les résultats sur la reconstruction de réseaux à une couche de réticulation à partir de clades souples des sections 1.4.2 et 2.3 ont fait l'objet d'un article à la conférence ISMB/ECCB 2009, publié dans Bioinformatics :

- [Huson *et al.*, 2009] Daniel Huson, Regula Rupp, Vincent Berry, Philippe Gambette & Christophe Paul : Computing Galled Networks from Real Data, *Bioinformatics* 25(12), *Proceedings of the seventeenth Annual Conference on Intelligent Systems for Molecular Biology & eighth European Conference on Computational Biology (ISMB'09)*, p. i85-i93, 2009.

Le lien entre clades et triplets de la section 1.3.2(d), le lien entre réseaux enracinés de niveau 1 et hiérarchies faibles de la section 1.5.1, ainsi que les résultats sur l'encodage des réseaux de niveau 1 et 2 de la section 3.3, sont réunis dans un article soumis :

- [Gambette et Huber, 2010] Philippe Gambette & Katharina T. Huber : A Note on Encodings of Phylogenetic Networks of Bounded Level, soumis à *Journal of Mathematical Biology*, 2010.

La définition du paramètre de niveau dans un contexte non enraciné de la section 1.4.4, les liens entre réseaux non enracinés de niveau 1 et ensembles circulaires de bipartitions de la section 1.5.2, ainsi que les résultats sur la reconstruction à partir de quadruplets de la section 2.2 sont présentés dans un article en préparation :

- [Gambette *et al.*, 2010] Philippe Gambette, Vincent Berry & Christophe Paul : Quartets and unrooted phylogenetic networks, *en préparation*, 2010.

Le logiciel d'assistance à la sélection de taxons dont le concept est présenté en section 4.1.4, fait l'objet d'un article en préparation :

- Philippe Gambette & Vincent Berry : HeurisTree, an interface for data selection before phylogenetic network reconstruction, *en préparation*, 2011.

L'ensemble de la démarche de reconstruction combinatoire de cette thèse a fait l'objet d'un exposé invité en 2009 aux Journées de la Société Française de Systématique, à paraître dans la revue Biosystema :

- Philippe Gambette : Reconstruction combinatoire de réseaux phylogénétiques, *Biosystema*, à paraître, 2010.

Première partie

Approche combinatoire des réseaux phylogénétiques

1 Arbres et réseaux comme objets combinatoires

Dans ce chapitre, nous introduisons le vocabulaire et les objets mathématiques utiles pour la reconstruction de réseaux phylogénétiques par des méthodes combinatoires. Nous présentons également des résultats nouveaux sur la structure des objets étudiés, et la façon dont on peut les mettre en relation.

1.1 Premières définitions

Pour définir formellement les arbres et les réseaux phylogénétiques, nous aurons besoin du vocabulaire classique de la théorie des graphes, qui fournira d'ailleurs de nombreux outils tout au long de cette thèse. Nous définirons formellement les arbres non enracinés comme des graphes, puis les arbres enracinés comme des graphes orientés.

1.1.1 Réseaux et graphes orientés

Commençons par définir un **réseau** (ou **graphe** simple sans boucle) $N = (V(N), E(N))$, qui est constitué d'un ensemble $V(N)$ de points appelés **sommets** ou **noeuds**, mis en relation par un ensemble $E(N) \subseteq \{\{x,y\} \mid x \neq y \in V\}$ de liens appelés **arêtes**. Quand il n'y a pas d'ambiguïté sur le réseau concerné, on notera simplement V son ensemble de sommets et E son ensemble d'arêtes. On notera xy l'arête $\{x,y\}$, et l'on dit que x est **adjacent** à y, et l'arête xy est **incidente** à x et y. On dit aussi que x et y sont **voisins**. Le **degré** $\delta(x)$ d'un sommet x est le nombre de voisins de x. Un sommet de degré 0 est appelé **sommet isolé**.

Le **sous-graphe de** N **induit** par un ensemble de sommets $V' \subseteq V(N)$ est le graphe $N[V'] = (V', \{\{x,y\} \in E(N) \mid x,y \in V'\})$. Étant donné un graphe $N = (V,E)$ et un sous-ensemble d'arêtes $E' \subseteq E$, nous notons $N - E' = (V, E - E')$ le graphe obtenu en effaçant de N les arêtes de E'. On dit alors que $N - E'$ est un **graphe partiel** de N.

Un graphe $G = (V,E)$ est un **stable** si $E = \emptyset$, c'est une **clique** si $E = \{xy \mid x,y \in V\}$. Un graphe $G = (V_1 \cup V_2, E)$ est **biparti** si $G[V_1]$ et $G[V_2]$ sont des stables. Si de plus chaque sommet de V_1 est adjacent à tous ceux de V_2, G est une **biclique**. Par extension, étant donné un réseau $N = (V,E)$, on dit qu'un sous-ensemble V' de sommets est un stable (respectivement une **clique**, une **biclique**) de N si $N[V']$ est un stable (respectivement une clique, une biclique).

Étant donné un ensemble $S = \{s_1, \ldots, s_k\}$ de sommets, le graphe $P_k = (S, \{s_i s_{i+1} \mid 1 \leq i < k\})$ est un **chemin** (d'**extrémités** s_1 et s_k, et dont les autres sommets sont les **sommets internes**) et $C_k = (S, \{s_i s_{i+1} \mid 1 \leq i < k\} \cup \{s_k s_1\})$ est un **cycle**. Par extension, étant donné un réseau $N = (V, E)$, on appelle **chemin** ou **chaîne** (respectivement **cycle**) de N tout graphe $N' = (V' \subseteq V, E' \subseteq E)$ tel que N' est un chemin (respectivement un cycle). Un réseau N est **connexe** si entre toute paire de sommets u et v de N il existe un chemin de N dont les extrémités sont u et v. Une **composante connexe** d'un réseau N est un sous-graphe induit maximal (par inclusion) de N qui est connexe. Un **arbre** est un réseau connexe sans cycle. Un **arbre couvrant** d'un réseau $N = (V, E)$ est un arbre $T = (V, E')$ qui est un graphe partiel de N.

Un graphe est **planaire** s'il peut être représenté dans le plan sans croisement d'arêtes. Dans sa représentation planaire, une **face** est un ensemble de points tel que pour toute paire de points de cet ensemble, on peut tracer une ligne de l'un à l'autre qui ne croise aucune arête, et la **face extérieure** est celle de surface infinie. Un graphe est dit **planaire extérieur** s'il a une représentation planaire telle que tous ses sommets appartiennent à la face extérieure.

Un **graphe orienté** est un graphe dans lequel une orientation est fournie à chaque arête, alors appelée **arc**. Plus formellement, un graphe orienté $G = (V(G), A(G))$ est constitué d'un ensemble $V(G)$ de sommets mis en relation par un ensemble de couples de sommets $A(G) \subseteq \{(x, y) \mid x \neq y \in V\}$. Pour l'arc (x, y), on appelle x la **source** et y la **cible**, et on dit que x et y sont **voisins**. Le **degré entrant** d'un sommet v est $\delta^+(v) = \{(x, v) \mid x \in V(G)\}$, son **degré sortant** est $\delta^-(v) = \{(v, x) \mid x \in V(G)\}$, et son degré est $\delta(v) = \delta^-(v) + \delta^+(v)$. Le **graphe non orienté sous-jacent** de G est alors $U(G) = \{V(G), \{xy \mid (x, y) \in A(G)\}\}$.

Un **chemin orienté** de s_1 à s_k dans G est un graphe orienté $G' = (\{s_1, \ldots, s_k\} \subseteq V(G), A' \subseteq A(G))$ tel que $(s_i, s_{i+1}) \in A'$ pour $1 \leq i < k$. Remarquons qu'alors $U(G')$ est un chemin d'$U(G)$. Un graphe orienté est dit **sans circuit** si pour tout arc (s_k, s_1) il ne contient pas de chemin orienté de s_1 à s_k. Un graphe orienté G est **connexe** si $U(G)$ est connexe, et **fortement connexe**, si entre toute paire de sommets x et y de G il existe un chemin orienté de x à y et de y à x.

Pour un graphe (respectivement un graphe orienté) connexe G, un **isthme** est une arête (resp. un arc) dont la suppression rend G non connexe, l'isthme est **trivial** s'il est incident à un sommet de degré 1. Une **coupe** est un ensemble d'arêtes dont la suppression rend G non connexe, c'est une **coupe minimale** si elle est minimale pour l'inclusion, c'est-à-dire qu'elle ne contient aucune autre coupe. Par exemple l'ensemble $\{e_1, e_2\}$ est une coupe du réseau N' de la figure 1.8(ii). Un **sommet d'articulation** est un sommet dont la suppression rend G non connexe. Un **bloc** S de G est un sous-ensemble de sommets de G, maximal pour l'inclusion, tel que $G[S]$ ne contient pas de sommet d'articulation. Un **blob** (ou **composante 2-arête-connexe**) d'un graphe G est un sous-ensemble de sommets $S \subseteq V(G)$, maximal pour l'inclusion, tel que $G[S]$ ne contient pas d'isthme. Il est dit **trivial** s'il ne contient qu'un sommet.

Une **subdivision** d'une arête xy (resp. d'un arc (x,y)) dans un graphe (resp. graphe orienté) G consiste à supprimer cette arête (resp. cet arc), ajouter un sommet w, ainsi que deux arêtes xw et wy (resp. deux arcs (x,w) et (w,y)). Une **contraction** d'une arête xw ou d'arc (x,w) permet de réaliser l'opération inverse : elle consiste à attribuer à x l'ensemble des voisins de w, puis à supprimer w.

Deux graphes G_1 et G_2 (respectivement graphes orientés) sont **isomorphes** s'il existe une bijection $\phi : V(G_1) \to V(G_2)$ telle que $xy \in E(G_1) \Leftrightarrow \phi(x)\phi(y) \in E(G_2)$ (resp. $xy \in A(G_1) \Leftrightarrow \phi(x)\phi(y) \in A(G_2)$).

1.1.2 Arbres phylogénétiques

Définition 1.1 (Arbre enraciné et arbre phylogénétique) *Un **arbre enraciné** est un graphe orienté T dont $U(T)$ est un arbre, et tel que :*

- *un sommet, la **racine**, a degré entrant 0,*
- *des sommets, les **feuilles**, ont degré entrant 1 et degré sortant 0,*
- *les autres sommets ont degré entrant 1 et degré sortant strictement positif.*

*Les sommets qui ne sont pas des feuilles sont appelés **sommets internes**. Un **arbre phylogénétique** (enraciné ou non) **sur** un ensemble X de taxons (parfois appelé X**-arbre** [Barthélemy et Guénoche, 1988]) est un arbre (enraciné ou non) dont les feuilles sont étiquetées de façon bijective par X.*

On identifiera par la suite chaque feuille d'un arbre phylogénétique, enraciné ou non, à son étiquette. On considère habituellement que les arbres phylogénétiques (respectivement les arbres phylogénétiques enracinés) n'ont pas de sommet de degré 2 (resp. de degré entrant et sortant 1). Les arbres phylogénétiques (enracinés ou non) de degré maximal 3, et avec une racine de degré 2 pour les arbres enracinés, sont dits **binaires**.

Pour tout arc (x,y) d'un graphe orienté sans circuit G, x est appelé le **parent** de y, et y l'**enfant** de x. S'il existe dans G un chemin orienté de u à v alors u est appelé **ancêtre** de son **descendant** v, noté $v \preceq u$. La relation de **descendance** $\preceq$ est clairement une relation d'ordre. Dans un arbre enraciné, la racine est l'unique plus grand élément. On pourra préciser le nom du graphe G où se réalise la relation de descendance en indiquant $v \preceq_G u$. Un **plus petit ancêtre commun** d'un ensemble V de sommets est un sommet ancêtre de tous les sommets de V, minimal pour la relation de descendance, noté $lca_G(V)$.

Pour un arbre phylogénétique enraciné T, le **sous-arbre** de T induit par un sous-ensemble de taxons $S \subseteq X$, noté $T[S]$, est le sous-graphe induit par l'ancêtre commun de S et l'ensemble de ses descendants. On note $T_{|S}$ la **restriction** de T à S, c'est-à-dire l'arbre obtenu en supprimant les sommets de T qui n'ont pas de descendants dans S, puis en contractant chaque arc incident à un sommet de degré entrant au plus 1 et de degré sortant 1.

Notons que ces notions peuvent être étendues aux arbres non enracinés. Pour un arbre non enraciné T, on appellera **sous-arbres** de T les deux arbres enracinés obtenus par suppression d'une arête de T.

1.2 Propriétés combinatoires des arbres

1.2.1 Une richesse mathématique

Les arbres phylogénétiques, enracinés ou non, ont de nombreuses propriétés intéressantes.

Tout d'abord remarquons que les deux concepts sont liés : depuis un arbre phylogénétique enraciné T de racine ρ (dont un enfant est appelé x), il est possible d'obtenir un arbre phylogénétique non enraciné T' en considérant $U(T)$ ayant subi une contraction de l'arête $x\rho$. Inversement, pour tout arbre phylogénétique non enraciné T', on peut choisir une arête xy à subdiviser pour créer un sommet racine ρ, et orienter T' par un parcours depuis ρ pour obtenir un arbre phylogénétique enraciné T.

On peut également définir un arbre phylogénétique enraciné de manière récursive, en disant que c'est :

- soit un sommet étiqueté racine.
- soit un sommet racine ρ relié par des arcs (ρ, ρ_k) à un ensemble de sommets ρ_i, ..., ρ_j ($j > i$) qui sont chacun les sommets racine d'un arbre phylogénétique, dont les ensembles d'étiquettes sont tous disjoints.

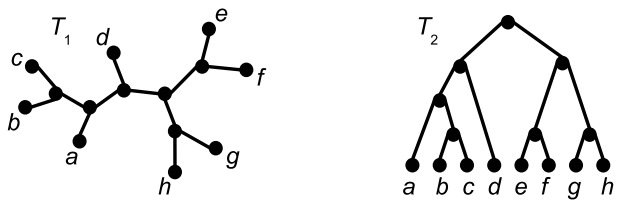
Cette définition récursive est une première illustration de leur richesse mathématique, elle peut être utilisée par exemple pour calculer le nombre d'arbres, binaires ou non, sur un ensemble de n taxons.

Une autre propriété intéressante est que tout ensemble de sommets possède un unique plus petit ancêtre commun. Elle découle du fait que pour tout sommet v d'un arbre phylogénétique enraciné T, il existe un unique chemin orienté de ρ à v dans T.

D'autres propriétés combinatoires intéressantes proviennent de l'analyse de sous-ensembles de feuilles, qui peuvent être utilisés pour représenter les arbres dont ils proviennent.

1.2.2 Décompositions en sous-ensembles de feuilles

On peut définir les arbres phylogénétiques en fonction de relations globales ou locales entre des sous-ensembles de feuilles, illustrées sur la figure 1.1. Nous verrons au début du chapitre 2 les intérêts combinatoires et algorithmiques de considérer ces sous-ensembles de feuilles plutôt que l'arbre dans sa globalité, et nous aborderons dans la section 1.4.1 les intérêts géométriques, qui nous permettront de mentionner brièvement les méthodes de reconstruction phylogénétique à partir des distances entre feuilles.



(i) (ii)

FIGURE 1.1 : Un arbre phylogénétique enraciné binaire T_1 (i) et un arbre phylogénétique non enraciné binaire T_2 (ii). L'arbre T_1 contient par exemple la bipartition $\{a,b,c\}|\{d,e,f,g,h\}$ et le quadruplet $ab|eh$, et l'arbre T_2 contient le clade $\{a,b,c\}$ et le triplet $b|eh$.

Définition 1.2 (Clades et bipartitions) *Un **clade** est un sous-ensemble non vide de taxons $C \subseteq X$. Étant donné un arbre T enraciné ou non, le clade C est **contenu** dans T (ou plus simplement C **est un clade de** T) s'il correspond à l'ensemble des feuilles d'un sous-arbre de T. En notant $\bar{C} = X - C$ l'ensemble complémentaire de C, pour un arbre non enraciné T, on appelle **bipartition** de T toute paire d'ensembles $B = C|\bar{C}$ telle que C est un clade de T (on dit alors que T **contient** B).*

Les bipartitions $C|\bar{C}$ et $\bar{C}|C$ sont considérées comme égales, et une bipartition $C|\bar{C}$ (respectivement un clade C) est dite **triviale** (resp. **trivial**) si C est un singleton.

$\mathcal{C}$ étant un ensemble de clades sur des taxons d'un ensemble X, on appelle $\mathcal{C}_{|S}$ la **restriction** de $\mathcal{C}$ à un sous-ensemble de taxons $S \subseteq X$, c'est-à-dire l'ensemble de clades $\{C \cap S \mid C \in \mathcal{C}\}$.

Définition 1.3 (Triplets) *Un **triplet** $a|bc$ (ou $bc|a$) est un arbre phylogénétique binaire enraciné sur trois feuilles a, b et c, où a, et le parent de b et c, sont des enfants de la racine.*

*Un triplet $a|bc$ est **contenu** dans un arbre phylogénétique enraciné T (ou plus simplement $a|bc$ **est un triplet de** T) si T contient deux sommets u et v, et des chemins de u à b, de u à c, de v à u et de v à a, ne partageant pas de sommet interne deux à deux.*

Remarquons qu'il est également possible de définir les triplets d'un arbre à partir d'ancêtres communs : $a|bc$ est un triplet de T si et seulement si le plus petit ancêtre commun de b et c dans T est descendant du plus petit ancêtre commun de a et b dans T. Un ensemble $\mathcal{R}$ de triplets sur un ensemble de feuilles X est **dense** si pour tout ensemble de trois feuilles de X, il existe au moins un triplet sur ces trois feuilles dans $\mathcal{R}$. La **restriction** d'un ensemble $\mathcal{R}$ de triplets à un sous-ensemble de taxons $S \subseteq X$ est notée $\mathcal{R}_{|S} = \{t = a|bc \mid a,b,c \in S \text{ et } t \in \mathcal{R}\}$.

Définition 1.4 (Quadruplets) *Un **quadruplet** $ab|cd$ est un arbre phylogénétique non enraciné sur quatre feuilles, tel que a et b ont un voisin u en commun, voisin d'un sommet v qui*

est aussi voisin de c *et* d. *Un quadruplet* $ab|cd$ *est* ***contenu*** *dans un arbre phylogénétique non enraciné* T *(ou plus simplement* $ab|cd$ ***est un quadruplet de*** T*) si* T *contient deux sommets* u *et* v *et cinq chemins de* a *à* u*, de* b *à* u*, de* u *à* v*, de* v *à* c *et de* v *à* d*, ne partageant pas de sommet interne deux à deux.*

Un ensemble Q de quadruplets sur un ensemble de feuilles X est **dense** si pour tout ensemble de quatre feuilles de X, il existe au moins un quadruplet sur ces quatre feuilles dans Q. Q étant un ensemble de quadruplets sur des taxons d'un ensemble X, on note $Q_{|S}$ la **restriction** de Q à un sous-ensemble de taxons $S \subseteq X$, c'est-à-dire l'ensemble de quadruplets $\{q = ab|cd \mid a,b,c,d \in S \text{ et } q \in Q\}$.

Un arbre phylogénétique peut être reconstruit de façon unique à partir de l'ensemble de tous ses clades, de toutes ses bipartitions, de tous ses triplets, ou de tous ses quadruplets [Buneman, 1971; Colonius et Schulze, 1981]. Nous détaillerons les aspects algorithmiques de cette reconstruction dans le chapitre 2.

L'ensemble des clades d'un arbre possède une propriété intéressante. On dit que deux clades se **chevauchent** s'ils ne sont ni inclus l'un dans l'autre, ni disjoints. Un ensemble de clades qui ne se chevauchent pas est une **famille laminaire**. Une propriété classique de la théorie des graphes (voir par exemple le théorème 13.21 de Schrijver [2003]) est que tout ensemble de clades d'un arbre enraciné est une famille laminaire, et qu'inversement, toute famille laminaire C est l'ensemble des clades d'un arbre enraciné, c'est pourquoi on dit aussi que C est une **hiérarchie** [Barthélemy et Guénoche, 1988].

De la même manière, dans le contexte non enraciné, deux bipartitions $B_1 = A_1|A_1'$ et $B_2 = A_2|A_2'$ sont **compatibles** si l'une des quatre intersections $A_1 \cap A_2$, $A_1 \cap A_2'$, $A_1' \cap A_2$ or $A_1' \cap A_2'$ est vide [Buneman, 1971]. Un ensemble de bipartitions est compatible si et seulement si il peut être représenté par un arbre non enraciné, où chaque arête correspond à une bipartition.

1.3 Propriétés combinatoires des réseaux

1.3.1 Réseaux abstraits et explicites

Comme on l'a vu en introduction, arbres et réseaux phylogénétiques peuvent être utilisés à des fins de classification ou de description de l'évolution. Ainsi, les arbres peuvent être considérés soit comme un ensemble d'arêtes qui traduisent les bipartitions entre deux ensembles de taxons qui ont un caractère différent, soit comme un ensemble d'embranchements qui traduisent des spéciations.

De la même manière, cette distinction implique une division des réseaux phylogénétiques en deux grandes sous-classes selon l'interprétation qu'on en fait [Huson et Bryant, 2006] : les réseaux phylogénétiques **abstraits** qui servent uniquement à classer et visualiser les données et les réseaux phylogénétiques **explicites** pour modéliser et décrire l'évolution.

En effet, dans ces derniers, chaque sommet du réseau représente un taxon (ancestral ou actuel), et chaque arête ou arc représente un transfert vertical, ou horizontal (hybridation, transfert horizontal de gène, etc.) de matériel génétique. Ces événements biologiques impliquent certaines contraintes, notamment de cohérence temporelle, qui permettent de définir les réseaux phylogénétiques explicites de façon formelle, comme nous le verrons dans la modélisation mathématique de la définition 1.5.

Les réseaux phylogénétiques abstraits peuvent en revanche recouvrir un cadre plus vaste d'objets mathématiques, sous réserve qu'ils décrivent des relations liées à l'évolution entre des espèces ou des organismes, selon la définition la plus large proposée par Daniel Huson sur la Wikipédia [1].

Les figures 1.2 et 1.3 donnent un aperçu de cette distinction en fournissant plusieurs exemples de réseaux phylogénétiques. Dans les réseaux abstraits de la figure 1.2, les sommets internes ne doivent pas être interprétés comme des taxons ancestraux, mais comme les artefacts d'une visualisation par un réseau. En revanche, cette interprétation est possible pour les réseaux explicites de la figure 1.3. Ces illustrations montrent également que les **réticulations**, c'est-à-dire les parties non arborées du réseau, peuvent être représentées de diverses manières.

Les définitions mathématiques des réseaux phylogénétiques explicites que nous proposons ci-dessous sont assez larges pour contenir la plupart des différentes sous-classes introduites dans la littérature, en autorisant des feuilles de degré entrant supérieur à 1 et des sommets de degré supérieur à 3. Nous discuterons toutefois ci-dessous, en particulier dans la section 1.3.3, de l'intérêt de restreindre ces définitions dans la suite de notre étude.

Définition 1.5 (Réseau phylogénétique explicite enraciné) *On appelle **réseau phylogénétique explicite enraciné** sur un ensemble* X *de taxons un graphe orienté* N *sans circuit et connexe tel que :*

- *un sommet, la **racine**, a degré entrant 0,*
- *des sommets, les **feuilles**, ont degré sortant 0, degré entrant strictement positif, et sont étiquetées de façon bijective par* X*,*
- *les autres sommets ont degré strictement supérieur à 2, degré entrant et sortant au moins 1.*

*Les sommets qui ne sont pas des feuilles sont appelés **sommets internes**. Les sommets de degré sortant strictement supérieur à 1 sont appelés **sommets de spéciation** et ceux de degré entrant strictement supérieur à 1 sont les **sommets hybrides**.*

Par exemple le réseau de la figure 1.4 a une racine ρ, neuf feuilles étiquetées de a à i et quinze sommets internes dont quatre sommets hybrides h_1, h_2, h_3 et h_4.

Définition 1.6 (Réseau phylogénétique explicite non enraciné) *On appelle **réseau phylogénétique explicite non enraciné** sur un ensemble* X *de taxons un réseau* N *tel que :*

1. "A phylogenetic network is any graph used to visualize evolutionary relationships between species or organisms", d'après http://en.wikipedia.org/wiki/Phylogenetic_network le 8 janvier 2006.

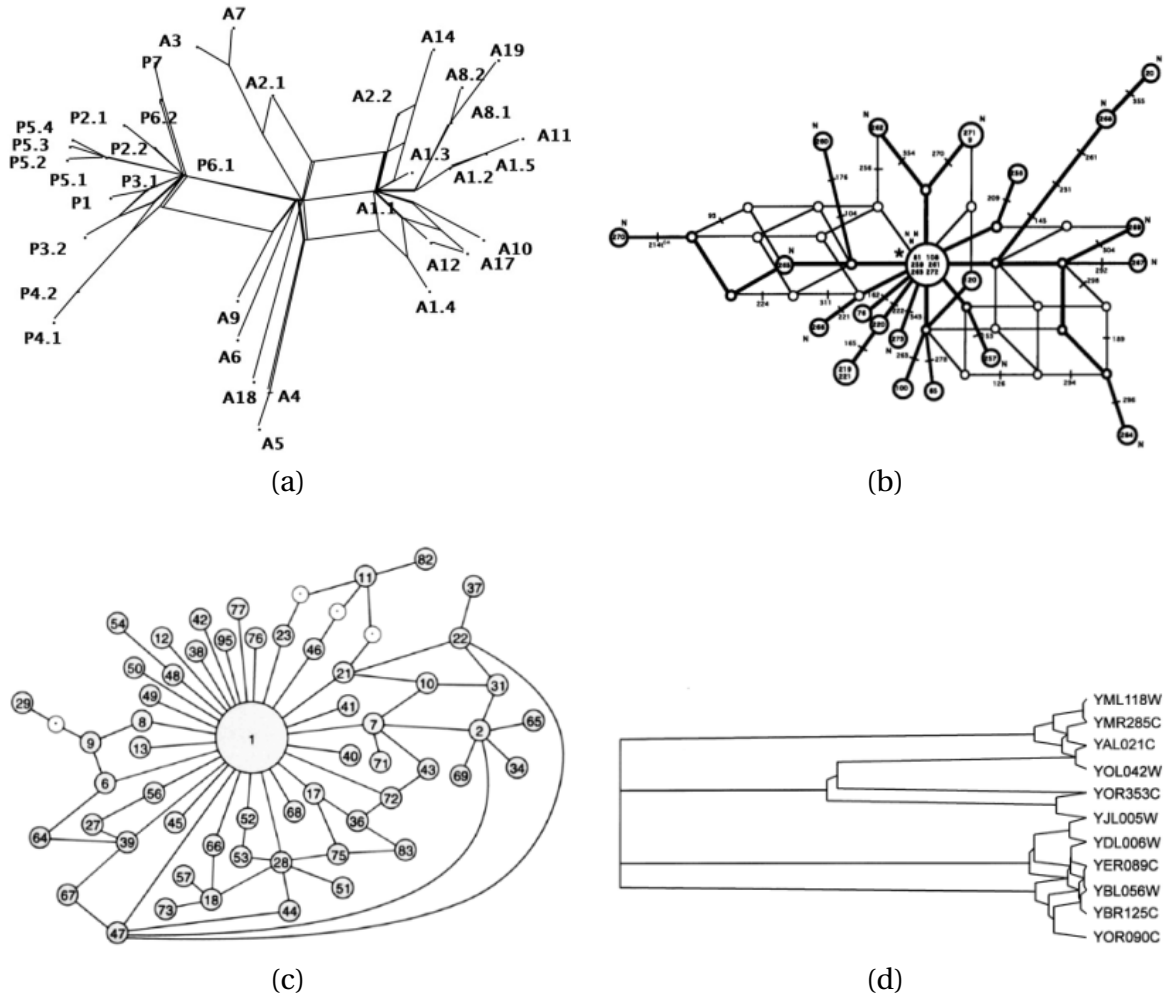
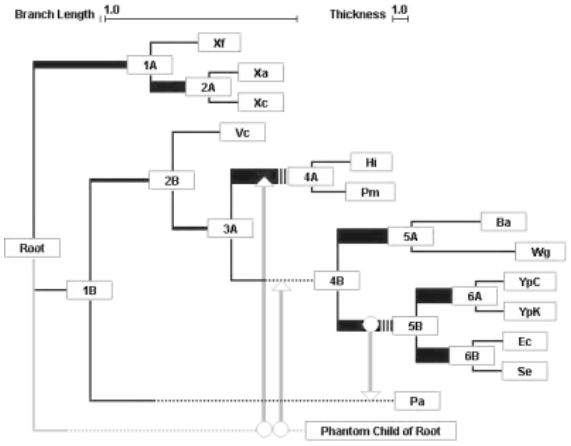


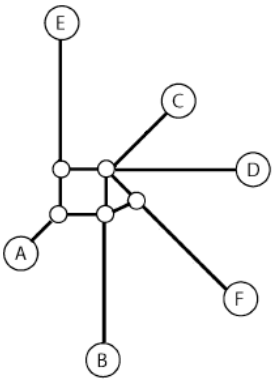
FIGURE 1.2 : Exemples de réseaux phylogénétiques abstraits : un réseau de bipartitions (a) [Huson et Bryant, 2006] construit avec SplitsTree, un réseau médian (b) [Bandelt *et al.*, 1995], un réseau couvrant minimal (c) [Excoffier et Smouse, 1994] construit avec Arlequin, et une pyramide (d) [Aude *et al.*, 1999] construite avec Pyramids. Précisons que l'ensemble des programmes dédiés aux réseaux phylogénétiques cités dans cette thèse sont décrits de façon concise, avec un lien vers leur page de téléchargement, sur http://www.atgc-montpellier.fr/phylnet/programs.

- *des sommets, les* ***feuilles****, ont degré 1 et sont étiquetées de façon bijective par* X*,*
- *les autres sommets, les* ***sommets internes****, ont degré strictement supérieur à 2.*

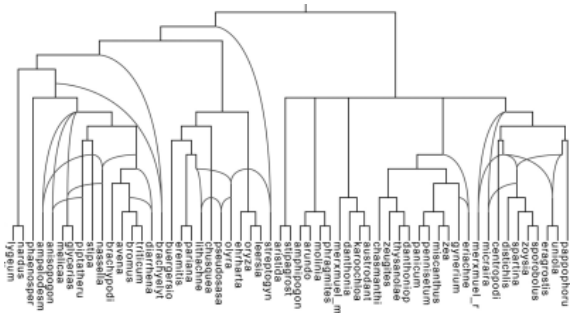
Remarque 1 *Un arbre phylogénétique enraciné (respectivement non enraciné) est en particulier un réseau phylogénétique explicite enraciné (respectivement non enraciné).*



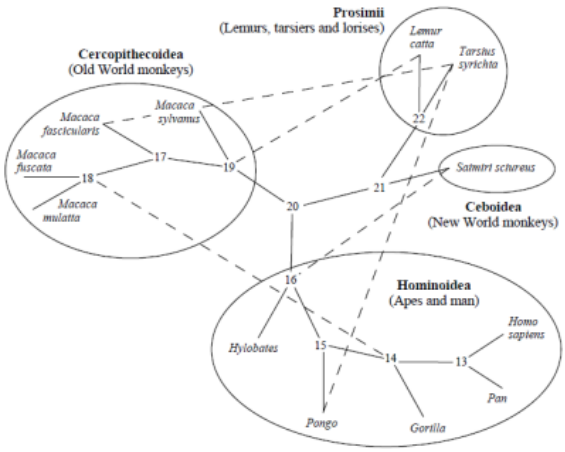
(a)



(b)



(c)



(d)

FIGURE 1.3 : Exemples de réseaux phylogénétiques explicites : un diagramme de synthèse (a) [MacLeod *et al.*, 2005] construit avec HorizStory, une union d'arbres maximalement parcimonieux (b) [Cassens *et al.*, 2005] construite avec CombineTrees, un réseau de niveau 4 (c) [van Iersel *et al.*, 2010a] construit avec Dendroscope [Huson *et al.*, 2007], et un réticulogramme (d) [Legendre et Makarenkov, 2000] construit avec T-Rex.

Un réseau phylogénétique explicite (enraciné ou non), dont tous les isthmes sont triviaux (incidents à des feuilles) est dit **simple**. Il est **binaire** si tous ses sommets ont degré au plus 3, comme celui de la figure 1.4.

Dans un réseau phylogénétique explicite enraciné, on peut distinguer les arcs dont la cible est un sommet de degré entrant strictement supérieur à 1 comme des **arcs d'hybridation**. Les autres arcs sont appelés **arcs de spéciation**.

1.3.2 Réseaux et sous-ensembles de feuilles

Définissons maintenant les triplets et quadruplets dans les réseaux phylogénétiques explicites binaires. Nous reprenons la définition classique apparue dans la littérature pour les triplets d'un réseau, qui généralise les définitions 1.3 et 1.4. Toutefois, nous verrons dans la section suivante que l'adaptation de ces définitions aux réseaux non binaires n'est pas immédiate et demande une discussion.

a) Triplets d'un réseau

Définition 1.7 (Triplets) *[Jansson et Sung, 2006] Pour trois feuilles distinctes* $a, b, c \in X$*, un* ***triplet*** $a|bc$ *est* ***contenu*** *dans un réseau phylogénétique explicite binaire enraciné* N *(ou plus simplement* $a|bc$ *est un triplet de* N*) si* N *contient deux sommets* u *et* v*, et des chemins de* u *à* b*, de* u *à* c*, de* v *à* u *et de* v *à* a*, ne partageant pas de sommet interne deux à deux.*

L'ensemble de tous les triplets d'un réseau N est noté $\mathcal{R}(N)$.

Remarque 2 *Il n'est plus possible de définir les triplets d'un réseau phylogénétique à partir des ancêtres communs, en particulier à cause de la perte de la propriété d'unicité du plus petit ancêtre commun dans les réseaux. En effet, on voit par exemple dans le réseau* N *de la figure 1.4 que les deux feuilles* g *et* h *ont deux plus petits ancêtres communs :* s_1 *et* s_2*. Les triplets* $g|hi$*,* $h|gi$ *et* $i|gh$ *sont tous trois contenus dans ce réseau, mais pas* $b|ac$*.*

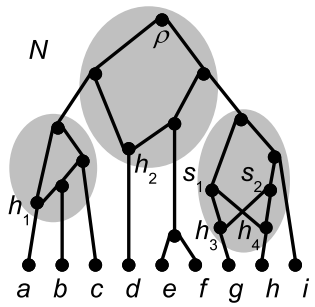


FIGURE 1.4 : Un réseau phylogénétique explicite enraciné N, de racine ρ et d'ensemble de taxons $X = \{a, b, c, d, e, f, g, h, i\}$. Comme dans les illustrations suivantes de réseaux et arbres enracinés, l'orientation des arcs n'est pas montrée mais on considère qu'ils sont tous orientés du haut vers le bas. Les zones grises sont des blobs contenant les sommets hybrides h_i, et tous les arcs non présents à l'intérieur d'une zone grise sont des isthmes.

b) Quadruplets d'un réseau

Définition 1.8 (Quadruplets) *[Gambette* et al.*, 2010] Pour quatre feuilles distinctes* $a, b, c, d \in X$*, un quadruplet* $ab|cd$ *est* ***contenu*** *dans un réseau phylogénétique explicite*

binaire non enraciné N *(ou plus simplement* $ab|cd$ *est un quadruplet de* N*) si* N *contient deux sommets* u *et* v *et cinq chemins, de* a *à* u*, de* b *à* u*, de* u *à* v*, de* v *à* c *et de* v *à* d*, ne partageant pas de sommet interne deux à deux.*

Par exemple le réseau non enraciné de la figure 1.5(i) contient les quadruplets $cd|ef$, $ce|df$ et $cf|de$, mais pas $cf|eg$. L'ensemble de tous les quadruplets d'un réseau N est noté $\mathcal{Q}(N)$.

On peut proposer une autre définition pour les quadruplets d'un réseau.

Définition 1.9 *Un quadruplet* $ab|cd$ *est contenu dans un réseau phylogénétique explicite non enraciné* N *s'il existe deux chemins non orientés à sommets disjoints dans* N*, l'un de* a *à* b *et l'autre de* c *à* d*.*

Proposition 1 *Les définitions 1.8 et 1.9 sont équivalentes.*

Démonstration. En effet, la définition 1.8 implique directement la définition 1.9, et la réciproque est vraie : comme a, b, c et d sont des feuilles, il doit y avoir un chemin (d'extrémités u et v) ne contenant pas a, b, c, ni d pour joindre les deux chemins disjoints $a-b$ et $c-d$ dans le réseau N qui est connexe. □

Quant à la définition des quadruplets d'un réseau abstrait, elle dépend du type de réseau abstrait considéré. Toutefois nous proposerons après la définition des bipartitions d'un réseau explicite non orienté, une façon de définir les quadruplets induits par une bipartition, et donc ceux contenus dans un réseau de bipartitions.

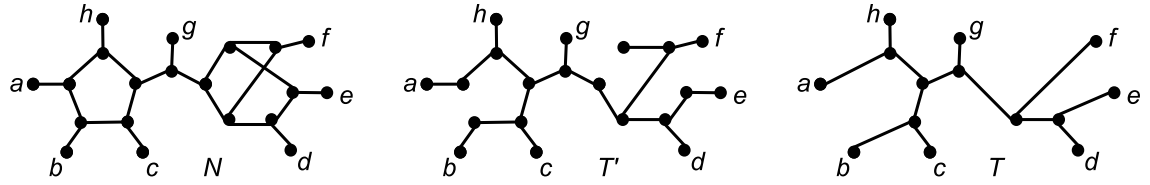
c) Arbres d'un réseau

Pour définir les clades et bipartitions d'un réseau phylogénétique, nous allons tout d'abord expliciter la définition d'un arbre contenu dans un réseau.

Définition 1.10 (Arbre contenu) *[Kanj* et al.*, 2008] Étant donné un réseau phylogénétique explicite enraciné (respectivement non enraciné)* N *sur un ensemble* X *de taxons, et un* X*-arbre phylogénétique enraciné (resp. non enraciné)* T*, on dit que* T *est* ***contenu*** *dans* N *s'il peut être obtenu depuis* N *par une suite de suppressions d'arcs (resp. d'arêtes), et de contractions d'arcs (resp. d'arêtes).*

L'ensemble de tous les arbres contenus dans un réseau phylogénétique explicite N est noté $\mathcal{T}(N)$.

Remarque 3 *Le concept d'arbre contenu dans un réseau explicite non enraciné se rapproche de celui d'arbre couvrant, comme montré en figure 1.5. En effet, si* T *est un arbre contenu dans* N*, l'arbre* T' *obtenu après les suppressions et avant les contractions d'arêtes indiquées dans la définition 1.10 est un arbre couvrant de* N*.*



(i) (ii) (iii)

FIGURE 1.5 : Un réseau non enraciné N (i), un arbre couvrant T′ de N (ii), et un arbre T contenu dans N (iii).

Remarque 4 *Si l'on étend la définition des arbres contenus aux* S*-arbres pour* $S \subseteq X$*, alors on obtient une nouvelle définition équivalente des triplets (respectivement, des quadruplets) contenus dans un réseau phylogénétique explicite enraciné (resp. non enraciné) binaire* N *: ce sont les* S*-arbres enracinés (resp. non enracinés), où* $|S| = 3$ *(resp.* $|S| = 4$*) et* $S \subseteq X$*, qui sont contenus dans* N*.*

Remarque 5 *On peut proposer un autre lien entre triplets et arbres contenus dans un réseau phylogénétique :* $\mathcal{R}(N) = \bigcup_{T \in \mathcal{T}(N)} \mathcal{R}(T)$*. En effet, pour tout triplet de* N*, il existe un arbre contenu dans* N *qui le contient, et inversement tout triplet d'un arbre* T′ *contenu dans* N *vérifie la condition sur les chemins disjoints dans* T′ *et a fortiori dans* N*.*

d) Clades d'un réseau

Définition 1.11 (Clades souples et stricts) *[Huson et Klöpper, 2007] Étant donné un réseau phylogénétique explicite enraciné* N *sur un ensemble* X *de taxons, le* ***clade souple*** C *est* ***contenu*** *dans* N *(ou plus simplement* C *est un clade souple de* N*) s'il est contenu dans un* X*-arbre* T *contenu dans* N*.* C *est un* ***clade strict*** *de* N *s'il est contenu dans tout* X*-arbre* T *contenu dans* N*.*

Par exemple, dans la figure 1.4, $\{g, h, i\}$ est un clade souple et strict, et $\{b, c\}$ ou $\{g, i\}$ sont des clades souples, mais pas stricts, de N. L'ensemble de tous les clades stricts d'un réseau N est noté $\mathcal{C}(N)$, et l'ensemble de tous ses clades souples est noté $\mathcal{S}(N)$. Notons que $\mathcal{C}(N) \subseteq \mathcal{S}(N)$.

On peut aussi remarquer qu'un clade strict d'un réseau N est l'ensemble des feuilles descendantes d'un sommet de N, et vice versa. Il est donc possible pour tout clade $C \subseteq X$ de lui associer un sommet v de N tel que C est l'ensemble des feuilles descendantes de v, noté $C_N(v)$. On dit alors que v **représente le clade strict** C. On étend cette notation aux arcs de N en considérant qu'un arc a de N représente un clade strict C si et seulement si son sommet cible représente C. Ceci nous permettra, dans les figures, d'étiqueter les arcs par les clades qu'ils représentent.

De la même manière, en notant $\mathcal{S}_N(v)$ l'ensemble des clades souples représentés par v dans les arbres contenus dans N, l'union des $\mathcal{S}_N(v)$ pour tous les sommets de N correspond aux clades souples de N. On dira alors qu'un sommet v de N **représente un clade souple** $C \subseteq X$ si $C \in \mathcal{S}_N(v)$, et qu'un arc représente le clade souple C si son sommet cible le fait.

Remarque 6 *La fonction de représentation des clades, stricts ou souples, par les sommets ou par les arcs, n'est pas bijective. En effet, pour le réseau* N *de la figure 1.4 contenant un sommet hybride* h_3 *de parents* s_1 *et* s_2 *et parent d'une feuille* g*, le singleton* $\{g\}$ *est représenté (en tant que clade souple ou strict) par* g*, par* h_3*, et par les arcs* (h_3, g)*,* (s_1, h_3) *et* (s_2, h_3)*. En général, on choisira pour représenter un clade souple ou strict* C *un arc dont la cible est un plus petit représentant de* C *pour la relation de descendance.*

Ces deux variantes de définition des clades, stricts et souples, correspondent respectivement au modèle "*Accumulation Phylogeny*" [Baroni and Steel, 2006] et "*Relaxed Accumulation Phylogeny*" [Willson, 2010a].

Enfin, nous pouvons noter un lien intéressant entre les triplets et les clades d'un réseau. Chaque clade C définit en effet de façon naturelle un ensemble de **triplets induits** $\mathcal{R}(C) = \{xy|z : x, y \in C, z \in \bar{C}\}$, et chaque ensemble $\mathcal{C}$ de clades induit également un ensemble de triplets $\mathcal{R}(\mathcal{C}) = \bigcup_{C \in \mathcal{C}} \mathcal{R}(C)$. On a alors la propriété suivante :

Proposition 2 *Si* C *est un clade souple d'un réseau phylogénétique explicite enraciné* N*, alors* $\mathcal{R}(C)$ *est un ensemble de triplets de* N*.*

Démonstration. Étant donné un réseau phylogénétique explicite enraciné N, $C \in \mathcal{S}(N)$ $\Rightarrow \exists T \in \mathcal{T}(N)$ tel que $C \in \mathcal{C}(T) \Rightarrow \forall x, y \in C, z \in \bar{C}, lca_T(\{x, y\}) \preceq lca_T(\{x, z\}) \Rightarrow \forall xy|z \in \mathcal{R}(C), xy|z \in \mathcal{R}(T) \Rightarrow \forall xy|z \in \mathcal{R}(C), xy|z \in \mathcal{R}(N)$. □

La réciproque de cette propriété est fausse dans le cas général, comme le montre l'exemple de la figure 1.6. Cependant elle est vraie pour des classes restreintes de réseaux phylogénétiques comme nous le verrons en section 3.3.1. D'autres auteurs ont approfondi les liens entre triplets et clades d'un réseau [van Iersel and Kelk, 2011], avec des conséquences importantes sur la complexité de certains problèmes de reconstruction que nous détaillerons en section 2.1.2.

e) Bipartitions d'un réseau

Définition 1.12 (Bipartitions) *Étant donné un réseau phylogénétique explicite non enraciné* N *sur un ensemble* X *de taxons, on appelle* ***bipartition*** *de* N *toute bipartition d'un* X*-arbre contenu dans* N *(on dit alors que* N ***contient*** B*).*

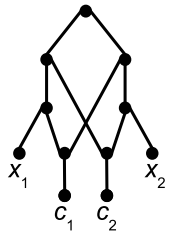


FIGURE 1.6 : Réseau N où l'équivalence clades-triplets induits n'est pas respectée : $c_1c_2|x_1 \in \mathcal{R}(N)$ et $c_1c_2|x_2 \in \mathcal{R}(N)$ mais $\{c_1, c_2\} \notin \mathcal{S}(N)$.

Par exemple, comme $B = \{a, h\}|\{b, c, d, e, f, g\}$ est une bipartition de l'arbre T de la figure 1.5(iii) qui est contenu dans le réseau N de la figure 1.5(i), alors B est une partition de N. L'ensemble de toutes les bipartitions d'un réseau N est noté $\mathcal{B}(N)$.

Nous pouvons donner une autre définition des bipartitions d'un réseau explicite non enraciné. Cette seconde définition est basée sur les coupes minimales, et similaire à celle donnée par Brandes et Cornelsen [2009]. Bien qu'ils affirment que les réseaux phylogénétiques explicites "représentent différemment les bipartitions" par rapport à la représentation en coupes minimales qu'ils proposent pour les réseaux qu'ils considèrent, nous prouvons que la définition ci-dessous et la définition 1.12 sont équivalentes.

Définition 1.13 *Une bipartition $A|\bar{A}$ est contenue dans un réseau phylogénétique non enraciné* N *s'il existe une coupe minimale de* N *qui déconnecte* A *de* $\bar{A}$.

Proposition 3 *Les définitions 1.12 et 1.13 sont équivalentes.*

Démonstration. Supposons que N est déconnecté par une coupe minimale E en deux réseaux : N_A qui contient toutes les feuilles de A et $N_{\bar{A}}$ qui contient $\bar{A}$, comme montré dans l'exemple de la figure 1.7. Soient alors uv une arête de E, et deux arbres couvrants T_u de N_A d'une part, et T_v de $N_{\bar{A}}$ d'autre part, tels que $u \in T_u$ et $v \in T_v$. Alors la bipartition $A|\bar{A}$ est contenue dans un arbre T' (contenu dans N) obtenu par l'union de T_u, T_v et uv, suivi d'un nombre nécessaire de contractions d'arêtes pour faire disparaître tout sommet de degré 2 (en évitant bien sûr la contraction de l'arête uv).

Supposons maintenant que la bipartition $A|\bar{A}$ est contenue dans un arbre contenu dans N. Alors il existe une arête x de T qui déconnecte T en deux sous-arbres, T_A qui contient A et $T_{\bar{A}}$ qui contient $\bar{A}$. Rappelons que d'après la remarque 3, tout arbre contenu dans un réseau phylogénétique non enraciné N peut être associé à un arbre couvrant de N. Appelons donc T'_A l'arbre couvrant de N associé à T_A et $T'_{\bar{A}}$ l'arbre couvrant de N associé à $T_{\bar{A}}$. Construisons l'ensemble d'arêtes E de la manière suivante : pour tout chemin d'un sommet de T'_A vers un sommet de $T'_{\bar{A}}$, si ce chemin ne contient aucune arête de T'_A, ni de $T'_{\bar{A}}$, ni x, on ajoute la première arête de ce chemin (incidente au sommet de T'_A) dans E. Finalement, on ajoute x dans E. Cet ensemble E est une coupe minimale de N qui

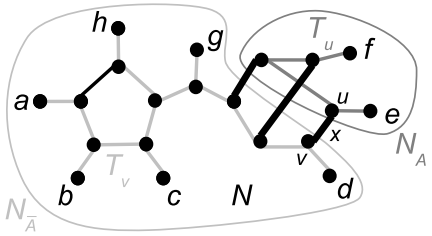
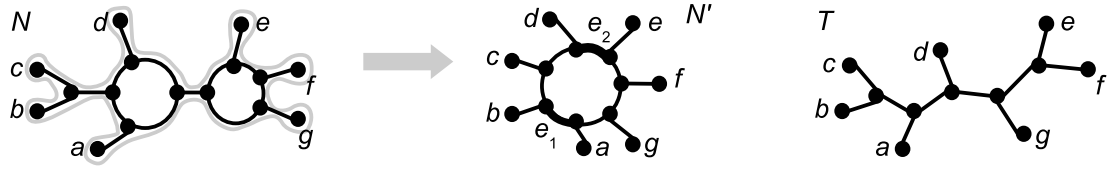


FIGURE 1.7 : Un réseau phylogénétique explicite non enraciné N, séparé en deux réseaux N_A et $N_{\bar{A}}$ par une coupe indiquée en gras et représentant la bipartition $A|\bar{A}$, pour $A = \{e, f\}$. Les arêtes du sous-arbre couvrant T_u de N_A sont en gris foncé, et celles du sous-arbre couvrant T_v de $N_{\bar{A}}$ sont en gris clair.

déconnecte A de $\bar{A}$. En effet, c'est une coupe car par définition, en supprimant les arêtes de E, on déconnecte T'_A de $T'_{\bar{A}}$. De plus, s'il existait une autre coupe $E' \subsetneq E$, considérons une arête $x' \in E - E'$: elle se trouverait sur un chemin de T'_A à $T'_{\bar{A}}$. Or aucune autre arête de ce chemin n'appartient à E', donc T'_A et $T'_{\bar{A}}$ ne seraient pas déconnectés, et donc E' ne serait pas une coupe : absurde ! □

Remarque 7 *Le fait que toutes les bipartitions d'un arbre non enraciné* T *sont contenues dans un réseau phylogénétique non enraciné* N *n'implique pas nécessairement que* T *est contenu dans* N. *Par exemple, les bipartitions de l'arbre* T *de la figure 1.8(iii) sont contenues dans le réseau* N′ *de la figure 1.8(ii) mais* T *n'est pas contenu dans* N′.

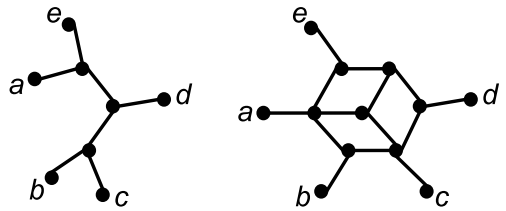


(i) (ii) (iii)

FIGURE 1.8 : Un réseau non enraciné N (i) et un réseau simple non enraciné $N' = \mathtt{Simple}(N)$ (ii) tel que $\mathcal{B}(N) \subseteq \mathcal{B}(N')$ et $\mathcal{Q}(N) \subseteq \mathcal{Q}(N')$. L'arbre T est contenu dans N mais pas dans N′, alors que $\mathcal{B}(T) \subseteq \mathcal{B}(N')$.

De même que pour les clades et les triplets dans le contexte enraciné, une bipartition $B = A|\bar{A}$ est naturellement associée à un ensemble de **quadruplets induits** $\mathcal{Q}(B) = \{ab|cd : a, b \in A, c, d \in \bar{A}\}$, tel que si B est contenue dans un réseau explicite non enraciné N, alors $\mathcal{Q}(B)$ est aussi contenu dans N.

Précisons également que la notion d'arbre contenu, qui est bien définie pour les réseaux explicites, l'est moins pour les réseaux abstraits. Si l'on considère par exemple les **réseaux de bipartitions**, qui permettent de visualiser chaque bipartition sous forme d'une coupe minimale constituée par un ensemble parallèle d'arêtes d'un graphe non orienté [2], on peut s'interroger sur la définition la plus appropriée. En fait, les méthodes existantes de reconstruction de réseaux abstraits à partir d'arbres ne procèdent pas directement à partir des arbres, mais à partir de leurs bipartitions : que ce soit pour construire des réseaux médians [Holland et Moulton, 2003] ou des réseaux de bipartitions [Huson *et al.*, 2004], les auteurs ne précisent pas de quelle façon ils contiennent les arbres, mais seulement comment ils contiennent les bipartitions de ces arbres. Or, si nous choisissons de baser la définition d'arbre contenu dans un réseau abstrait sur le concept d'arbre couvrant, comme pour les réseaux explicites, et comme le font Woolley *et al.* [2008], nous obtenons des arbres contenus dont les bipartitions ne sont pas nécessairement toutes contenues dans le réseau. Par exemple, l'arbre T de la figure 1.9(i) peut être obtenu à partir du réseau de bipartitions N de la figure 1.9(ii) suite à des opérations de suppression puis de contraction d'arêtes. Toutefois, la bipartition $\{a,d,e\}|\{b,c\}$ est contenue dans T mais pas dans N. En effet, outre les bipartitions triviales, N ne contient que 3 bipartitions représentées par les ensembles parallèles d'arêtes : $\{a,b,c\}|\{d,e\}$, $\{a,b,e\}|\{c,d\}$ et $\{a,e\}|\{b,c,d\}$. Ainsi, nous n'approfondirons pas l'étude combinatoire des arbres ou quadruplets contenus dans les réseaux phylogénétiques abstraits.



(i) (ii)

FIGURE 1.9 : Un arbre phylogénétique non enraciné T (i) et un réseau de bipartitions N (ii) qui le contient en tant qu'arbre couvrant, mais ne contient pas la bipartition $\{a,d,e\}|\{b,c\}$ pourtant contenue dans T.

1.3.3 Multifurcations et multiréticulations

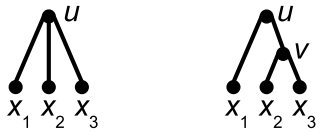
Les triplets et quadruplets étant des arbres binaires, les définitions 1.3, 1.4, 1.7 et 1.8 sont naturelles dans des arbres ou réseaux binaires. Toutefois, pour les arbres et sommets non binaires, une discussion s'impose selon les données disponibles en entrée et les ré-

2. La définition formelle des réseaux de bipartitions [Dress et Huson, 2004] sera donnée en section 1.4.1

sultats recherchés. Pour nous en convaincre, focalisons-nous sur le cas des triplets et des clades contenus dans les arbres et réseaux enracinés.

Dans un réseau phylogénétique explicite enraciné, on appelle **multifurcation** un sommet de degré sortant strictement supérieur à 2, et **multiréticulation** un sommet de degré entrant strictement supérieur à 2. Il y a deux façons d'interpréter la multifurcation dans l'arbre non-binaire de la figure 1.10(i) :

- soit on considère que les trois triplets $x_1|x_2x_3$, $x_2|x_1x_3$ et $x_3|x_1x_2$ sont possibles, et devraient donc être contenus dans cet arbre. Ainsi, la multifurcation exprime une incertitude : on ne sait pas quelle a été la première spéciation qui a donné naissance à ces trois espèces. Dans ce cas il faut autoriser les deux sommets u et v de la définition 1.7 à être égaux.
- soit on sait qu'il est impossible de trouver une configuration binaire plus probable qu'une autre, on rejette donc les trois triplets. On considère alors que les deux spéciations successives ont eu lieu à un intervalle de temps très bref (voire simultanément, car elles correspondent en fait à un processus progressif, comme suggéré par la figure 0.4 page 6). Dans ce cas il faut obliger les deux sommets u et v de la définition 1.7 à être distincts.



(i) (ii)

FIGURE 1.10 : Un arbre phylogénétique enraciné à trois feuilles avec une multifurcation (i) et un raffinement de cet arbre, le triplet $x_1|x_2x_3$ (ii).

Classiquement, dans le contexte de reconstruction phylogénétique à partir de triplets, les multifurcations sont considérées comme des incertitudes, et c'est la première interprétation qui est choisie. En effet, la seconde interprétation implique de trouver un critère de rejet de l'ensemble des trois triplets concernant 3 feuilles, difficile à définir. Ainsi, on considérera plutôt que l'arbre enraciné de la figure 1.10(i) contient les trois triplets possibles.

En revanche, pour les clades, la seconde interprétation est plus simple d'utilisation : en effet, le sommet multifurcation représente naturellement le clade constitué par l'ensemble de ses feuilles descendantes. Ainsi, on considérera que l'arbre enraciné de la figure 1.10(i) contient le clade $\{x_1, x_2, x_3\}$ mais pas le clade $\{x_1, x_2\}$, ni $\{x_1, x_3\}$, ni $\{x_2, x_3\}$.

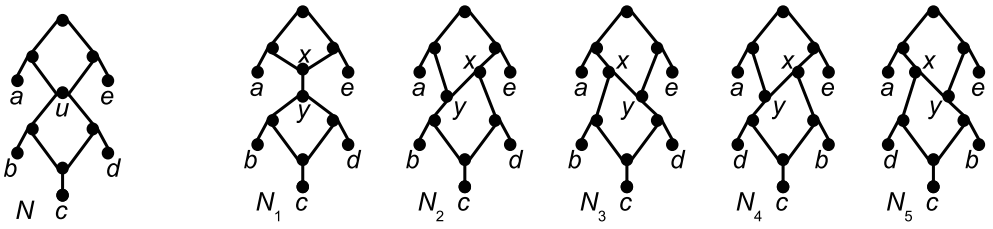
Notons que la présence de multiréticulations de degré sortant 1 ne nécessite en revanche aucune discussion car elle n'a aucune incidence sur l'ensemble de triplets ou de clades contenus.

On définit alors le **raffinement** d'un arbre ou d'un réseau enraciné N comme un arbre ou un réseau enraciné N' tel que N peut être obtenu à partir de N' par une suite de

contractions d'arcs (où le réseau obtenu après chaque contraction est bien un réseau phylogénétique explicite enraciné).

Ainsi, en précisant dans la définition 1.7 que les sommets u et v peuvent être identiques, alors $\mathcal{R}(N') \subseteq \mathcal{R}(N)$ pour tout raffinement N' de N. En revanche, $\mathcal{S}(N) \subseteq \mathcal{S}(N')$. On peut également préciser que $\mathcal{R}(N) = \bigcup_{N' \text{ raffinement de } N} \mathcal{R}(N')$ et que $\mathcal{S}(N) = \bigcap_{N' \text{ raffinement de } N} \mathcal{S}(N')$. Pour éviter ce type de paradoxe, il est nécessaire de bien préciser quelle définition est utilisée dans le cas où l'on autorise les multifurcations.

Examinons maintenant les situations où des sommets sont à la fois sommet de spéciation et sommet hybride, à l'aide de l'exemple du réseau N de la figure 1.11(i) qui contient un sommet u de degré entrant 2 et de degré sortant 2. Pour lever l'incertitude, examinons l'ensemble de tous ses raffinements possibles N_i présentés dans les figures 1.11(ii-vi).



(i) (ii) (iii) (iv) (v) (vi)

FIGURE 1.11 : Un réseau N contenant un sommet de spéciation hybride (i) et ses cinq raffinements possibles N_1 (ii), N_2 (iii), N_3 (iv), N_4 (v) et N_5 (vi).

On y remarque que le seul raffinement de ce réseau qui contient exactement les mêmes triplets, clades souples et arbres que N est N_1. Ainsi, dans le contexte de reconstruction de réseaux enracinés à partir de clades, d'arbres ou de triplets, comme défini dans la section 1.3.2, il est inutile de considérer qu'un sommet u puisse être à la fois sommet de spéciation et sommet hybride, car le réseau obtenu en remplaçant u par deux sommets x et y, où x est parent de y, et où les parents de x sont ceux de u et les enfants de y sont ceux de u, contient les mêmes triplets.

Ceci explique que bien que les réseaux phylogénétiques explicites enracinés contenant des sommets de spéciation hybrides aient été considérés dans la littérature [Choy *et al.*, 2005; Rosselló et Valiente, 2009], ils ne l'ont cependant jamais été dans le contexte de reconstruction à partir de triplets ou de clades.

On peut toutefois tenir à la possibilité de représenter l'incertitude par de tels sommets de spéciation hybrides, et vouloir affirmer que le triplet $b|ed$ (contenu dans N_2 et N_3 mais pas dans N_1 ni N_4 ni N_5) ou $a|be$ (contenu uniquement dans tous les raffinements sauf N_3) soient également contenus dans N. Dans ce cas, une simple adaptation de la définition 1.7 suffit à autoriser cette interprétation du réseau N comme représentant l'ensemble de ses raffinements possibles : remplacer "ne partageant pas de sommet interne" par "ne partageant pas d'arc". Un tel choix est cependant lié à de nombreuses contraintes combi-

natoires car les divers raffinements de N ont d'importantes différences de structure. Nous verrons par exemple dans la section suivante que N_1 est un réseau à une couche de réticulation, et que c'est un réseau de niveau 1, alors que les quatre autres raffinements ne le sont pas.

Ainsi, dans la suite de ce manuscrit, nous éviterons la présence de sommets de spéciation hybrides, et dans le contexte de la reconstruction à partir de triplets nous nous focaliserons sur les réseaux binaires. Pour les clades, nous choisirons l'interprétation selon laquelle le sommet d'un arbre de degré sortant strictement supérieur à 2 est associé à un unique clade qui est l'ensemble de ses descendants.

1.4 Restrictions sur les modèles de réseaux

De la même façon que dans la section précédente nous avons justifié certaines restrictions sur les réseaux phylogénétiques que nous étudierons dans ce manuscrit, d'autres auteurs ont introduit des sous-classes de réseaux phylogénétiques, pour prendre en compte des contraintes biologiques, ou bien pour pallier une trop grande richesse combinatoire qui se traduirait par des algorithmes trop lents et inutilisables en pratique. L'introduction de ces restrictions permet donc d'obtenir des propriétés de structure qui conduisent à des algorithmes rapides (de reconstruction de comparaison, de génération aléatoire, de visualisation...), dont certains seront cités ou décrits au chapitre 2. L'arbre étant un modèle simple et fondé biologiquement, mais très restreint, de réseau phylogénétique, ce sont souvent des généralisations de ce modèle qui ont été proposées.

Nous présenterons tout d'abord des restrictions introduites sur certaines classes de réseaux phylogénétiques abstraits, puis évoquerons celles qui concernent les réseaux phylogénétiques explicites, en donnant en particulier de nouveaux résultats de structure pour ces réseaux. Nous terminerons cette section par une synthèse des relations connues ou découvertes à l'occasion de cette thèse, entre sous-classes de réseaux phylogénétiques.

1.4.1 Restrictions sur les ensembles de clades et de bipartitions

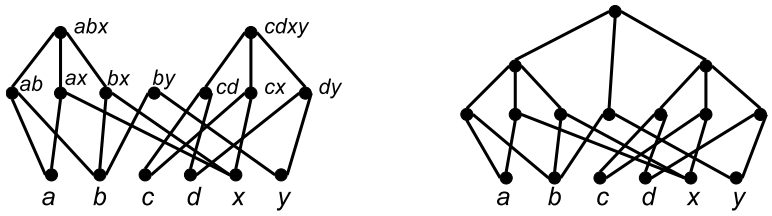
Les réseaux phylogénétiques abstraits servant à classifier des données, et à visualiser des relations entre elles, plutôt qu'à décrire une histoire évolutive explicite, il existe souvent des manières directes, rapides d'un point de vue algorithmique, et sans ambiguïté, pour associer un réseau abstrait à ces données (clades ou bipartitions), comme nous le détaillerons ci-dessous.

Il est aussi possible d'associer une distance canonique entre les feuilles concernées par un ensemble de clades. En effet, à partir de la mesure de similarité $S_{\mathcal{C}} : X \times X \to \mathbb{R}$ telle que $S_{\mathcal{C}}(a,b) = |\{C \in \mathcal{C} \mid a,b \in C\}|$, pour $a,b \in X$, il est possible, selon certaines restrictions sur les clades, de les retrouver depuis $S_{\mathcal{C}}(a,b)$ [Bandelt et Dress, 1989; Bryant et Berry, 2001]. De plus, en appliquant la **transformée de Farris** (voir par exemple [Semple et Steel, 2003]

ou [Dress *et al.*, 2007]) à cette mesure de similarité, on obtient une **distance** $D_{\mathcal{C}}$, c'est-à-dire une fonction de $X \times X$ dans $\mathbb{R}^+$, qui est symétrique, satisfait l'inégalité triangulaire, et s'annule uniquement pour des taxons égaux.

a) Restrictions sur les ensembles de clades

Pour les clades stricts, une manière naturelle de leur associer un réseau phylogénétique enraciné est de considérer le **diagramme de Hasse** des clades pour la relation d'inclusion [Huson et Rupp, 2008], illustré en figure 1.12(i). Ce réseau abstrait est construit de la manière suivante : son ensemble de sommets est l'ensemble $\mathcal{C}$ des clades fourni en entrée, et un sommet v correspondant à un clade $C(v) \in \mathcal{C}$ est un parent de u associé au clade $C(u) \in \mathcal{C}$ si $C(u) \subseteq C(v)$ et $\forall C \in \mathcal{C} - \{C(u), C(v)\}$, $C(u) \subseteq C \Rightarrow C \not\subseteq C(v)$. Le réseau ainsi associé aux clades d'une hiérarchie est bien un arbre. Ainsi, on considérera parfois par abus de langage qu'une hiérarchie, ensemble de clades, est un arbre phylogénétique. Plus généralement, on fera de même pour désigner sous le nom de **réseau de clades stricts** des ensembles de clades, en faisant référence au réseau naturel N qui les contient en tant que clades stricts, i.e. tel que $\mathcal{C}(N) = \mathcal{C} \cup X$, comme montré en figure 1.12(ii).



(i) (ii)

FIGURE 1.12 : Le diagramme de Hasse de l'ensemble $\mathcal{C} = \{\{a,b\},\{a,b,x\},\{a,x\},\{b,x\},\{b,y\},\{c,d\},\{c,d,x,y\},\{c,x\},\{d,y\}\}$ (i) et le réseau phylogénétique abstrait enraciné qu'on associe naturellement à $\mathcal{C}$ (ii).

Ces diverses classes de familles d'ensembles sont présentées par exemple dans la thèse de Brucker [2001]. Parmi les résultats sur ces objets, ceux qui nous intéressent en particulier en phylogénie sont les liens entre les distances feuille à feuille et les ensembles de clades stricts des réseaux correspondants. Nous verrons en section 1.5.1 l'intérêt de les mentionner dans cette thèse.

Une **hiérarchie faible** [Bandelt et Dress, 1989] est un ensemble $\mathcal{C}$ de clades tel que l'intersection de trois clades $C_1, C_2, C_3 \in \mathcal{C}$ est toujours égale à l'intersection de deux d'entre eux. Les hiérarchies faibles sont également appelées **médinclus** [Batbedat, 1988, 1989].

Une **prépyramide** [Bandelt, 1992] est un ensemble $\mathcal{C}$ de clades de X tel qu'il existe un ordre σ sur X où tout clade de $\mathcal{C}$ est un **intervalle** de σ, c'est-à-dire un ensemble d'éléments consécutifs dans σ.

Une **pyramide** [Diday, 1986] (respectivement une **quasi-hiérarchie** [Bandelt, 1992]) est une prépyramide (resp. une hiérarchie faible) close par intersection non vide, qui contient les singletons, l'ensemble X mais pas l'ensemble vide.

Une k-**hiérarchie faible** [Bertrand et Janowitz, 2002] est un ensemble de clades tel que l'intersection de $k+1$ clades est toujours égale à l'intersection de k d'entre eux, et qui est clos par intersection non vide, contient les singletons, l'ensemble X mais pas l'ensemble vide. Notons que les 2-hiérarchies faibles sont exactement les quasi-hiérarchies.

b) Restrictions sur les ensembles de bipartitions

De même que pour les ensembles de clades, il existe une manière canonique d'associer un réseau phylogénétique abstrait, mais non enraciné, à un ensemble de bipartitions. Il s'agit du **réseau médian** [Guénoche, 1986; Bandelt *et al.*, 1995], qui est un réseau de bipartitions qui contient de plus la propriété des **graphes médians**, c'est-à-dire que pour tout ensemble de trois sommets a, b, c du graphe, il existe un unique sommet qui appartient à un plus court chemin entre a et b, entre a et c, et entre b et c.

Un **réseau de bipartitions** N est un réseau biparti connexe dont il existe un coloriage de ses arêtes assurant, pour toute paire de sommets u, v de N, l'existence d'un ensemble C de couleurs tel que tous les plus courts chemins entre u et v contiennent exactement une fois chaque couleur de C. Ces réseaux, formellement définis par Dress et Huson [2004], peuvent être dessinés de telle manière que les arêtes de la même couleur sont parallèles, de même longueur, et constituent une coupe minimale du réseau, comme en figure 1.2(a), page 22, ou en figure 1.9(ii), page 30. Chaque ensemble d'arêtes de même couleur correspond également à une bipartition car, en tant que coupe, il sépare deux ensembles complémentaires de feuilles du réseau.

Un ensemble $\mathcal{B}$ de bipartitions est **faiblement compatible** si pour tout ensemble de trois bipartitions $A_1|\bar{A}_1$, $A_2|\bar{A}_2$ et $A_3|\bar{A}_3$, l'une des quatre intersections suivantes est vide : $A_1 \cap A_2 \cap A_3$, $A_1 \cap \bar{A}_2 \cap \bar{A}_3$, $\bar{A}_1 \cap A_2 \cap \bar{A}_3$, $\bar{A}_1 \cap \bar{A}_2 \cap A_3$ [Bandelt et Dress, 1992b]. On peut le définir de manière équivalente en disant que pour tous taxons $a, b, c, d \in X$, il n'existe pas dans $\mathcal{B}$ trois bipartitions qui séparent respectivement a et b de c et d, a et c de b et d, et a et d de b et c.

L'ensemble $\mathcal{B}$ est **circulaire** [Bandelt et Dress, 1992b] s'il existe un ordre σ sur X tel que pour toute bipartition $A|\bar{A}$ de $\mathcal{B}$, A ou $\bar{A}$ est un intervalle de σ. Il est k-**compatible** s'il ne contient pas d'ensembles de strictement plus de k bipartitions non compatibles deux à deux.

Remarque 8 *Les structures qui existent pour les bipartitions et les réseaux non enracinés peuvent également être utilisées dans un contexte enraciné. En effet, pour un réseau phylogénétique* N*, on peut considérer son graphe non orienté sous-jacent* $U(N)$ *et lui ajouter un* ***exogroupe*** o *adjacent à la racine de* N*, pour "marquer" la position de cette racine dans le réseau non enraciné obtenu. Dans ce réseau, chaque bipartition* $A|\bar{A}$ *peut être considérée*

comme le clade A *si* $o \notin A$*, ou comme le clade* $\bar{A}$ *sinon. Ainsi, certaines restrictions sur les bipartitions se traduisent en restrictions sur les clades, comme nous le détaillerons en section 1.5.3.*

1.4.2 Réseaux à une couche de réticulation

Nous passons maintenant aux restrictions sur les réseaux phylogénétiques explicites, et en particulier, dans cette section et la suivante, ceux qui sont enracinés.

Définition 1.14 *[Huson et Klöpper, 2007] Un réseau phylogénétique explicite enraciné* N *est appelé* ***réseau à une couche de réticulation*** *si pour tout sommet hybride* h *de* N*, et toute paire d'arcs d'hybridation* p *et* q *incidents à* h*, il existe un cycle dans le graphe non orienté sous-jacent* $U(N)$ *contenant uniquement les arcs* p *et* q *et des arcs de spéciation de* N*.*

Ainsi, tout réseau à une couche de réticulation peut être décomposé en un arbre (constitué par les arcs de spéciation de N, et d'un arc d'hybridation par sommet hybride) auquel on ajoute une couche d'arcs d'hybridation (les arcs gris dans la figure 1.13). Cette définition permet qu'un sommet hybride h' apparaisse sous un autre sommet hybride h, mais seulement quand tous les chemins orientés de la racine à h' contiennent h, comme c'est le cas dans la figure 1.13. Elle ne permet pas de représenter d'autres cas de "réticulations de réticulations".

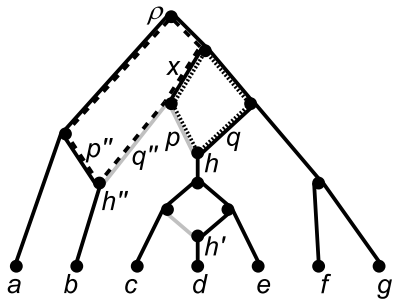


FIGURE 1.13 : Un réseau à une couche de réticulation contenant trois sommets hybrides dont h et h'. Les deux cycles associés à h et h' sont indiqués par des lignes respectivement en tirets et en pointillés. Contrairement à un réseau de niveau 1, dans lequel les cycles sont arcs-disjoints, ces deux cycles partagent un arc x, qui est nécessairement un arc de spéciation dans un réseau à une couche de réticulation.

Une conséquence utile de cette définition est que tout sommet hybride h d'un réseau à une couche de réticulation est un sommet d'articulation qui sépare ses descendants du reste du réseau [Huson et Klöpper, 2007].

Les réseaux à une couche de réticulation ont une propriété intéressante vis-à-vis des clades : ils peuvent être reconstruits à partir de n'importe quel ensemble de clades souples, d'après la proposition suivante.

Proposition 4 *Soit* $\mathcal{C} \subseteq \mathcal{P}(X)\setminus\emptyset$. *Il existe un réseau à une couche de réticulation qui contient l'ensemble de clades souples* $\mathcal{C}$.

Démonstration. Construisons ce réseau "universel" N, qui contient tout ensemble de clades (illustré en figure 1.14), de la manière suivante :

– la racine ρ a deux enfants u et v,

– les feuilles x_i, pour $i \in [1..n]$, ont chacune un parent x_i' dont les parents sont u et v.

Ainsi, pour tout clade $C \in \mathcal{C}$, C est contenu dans N de la manière suivante : l'arbre phylogénétique qui associe à toutes les feuilles de C le parent u, et qui associe à toutes les feuilles de $\bar{C}$ le parent v est contenu dans le réseau N, et contient le clade C. □

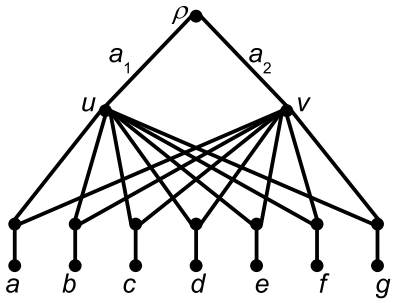


FIGURE 1.14 : Tout sous-ensemble non vide de l'ensemble de taxons $\{a, b, c, d, e, f, g\}$ est un clade souple de ce réseau à une couche de réticulation.

1.4.3 Réseaux de niveau k

a) Définition des réseaux de niveau k

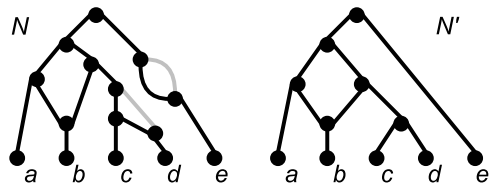
Définition 1.15 *[Choy* et al.*, 2005] Un réseau* N *de* ***niveau*** k *est un réseau enraciné explicite binaire dont chaque blob contient au plus* k *sommets hybrides. Il est* ***strictement de niveau*** k *s'il est de niveau* k *mais pas de niveau* $k-1$.

Plusieurs remarques s'imposent sur cette définition qui varie légèrement selon les articles et les contextes. Tout d'abord, les définitions utilisées classiquement n'évoquent pas les blobs, mais les blocs du réseau. Dans le cas où nous nous restreignons aux réseaux binaires (comme dans [Jansson et Sung, 2006] et la plupart des articles apparus depuis, mais contrairement à Choy *et al.* [2005] qui ne s'intéressaient pas à une problématique de

reconstruction de réseaux), les deux concepts sont équivalents, excepté pour les isthmes, qui correspondent chacun à un bloc (car en supprimant un de leurs sommets, le sommet unique restant est un graphe connexe, deux sommets séparés par un isthme constituent donc un graphe biconnexe) mais à deux blobs triviaux. Toutefois, si nous autorisons les sommets de réticulation hybrides, comme nous l'avons vu en section 1.3.3, considérer les blocs consiste à dire que le réseau N de la figure 1.11(i) de la page 32 a deux blocs, qu'il est de niveau 1, et donc le rendre équivalent à son raffinement N_1. Pour considérer que le sommet u représente une réelle incertitude et que N a également les raffinements N_2 à N_5, il faut prendre en compte le fait que les réseaux N_2 à N_5 sont de niveau 2, et donc utiliser le concept de blob dans la définition du réseau pour affirmer que N est de niveau 2. Ainsi, cette définition peut s'étendre aux réseaux non binaires.

En ce qui concerne les ajustements, rappelons que nous autorisons ici les feuilles hybrides, c'est-à-dire les sommets de degré entrant 2 et de degré sortant nul. Ce point sera un détail par la suite, sauf justement quand on étudiera l'unicité des réseaux qui contiennent un certain ensemble de triplets où, pour éviter les ambiguïtés inutiles, nous forcerons chaque feuille à n'avoir qu'un parent.

Pour la même raison, dans les chapitres 2 et 3, nous considérerons également qu'un réseau de niveau k ne contient aucun blob à 2 ou 3 sommets [3] : en effet, dans ce cas on autoriserait le réseau de la figure 1.15(i), alors que celui-ci est moins parcimonieux (en termes de nombre d'arêtes et de sommets) que le réseau N' de la figure 1.15(ii), qui contient exactement les mêmes arbres, clades souples et stricts, et triplets, et qu'on peut donc considérer comme tout aussi informatif biologiquement.



(i) (ii)

FIGURE 1.15 : Un réseau N de niveau 1 (selon une définition étendue) avec des blobs de moins de quatre sommets (a) et un réseau N' de niveau 1 selon la définition que nous employons ici (b), plus parcimonieux en termes de nombre d'arêtes et de sommets, qui contient exactement le même ensemble d'arbres, de clades souples et stricts, et de triplets.

Toutefois, pour des raisons de simplicité de définition des générateurs (voir la définition 1.16 ci-dessous), nous considérons dans cette section 1.4.3 que ces restrictions ne s'appliquent pas.

3. Pour être précis, parler d'un blob à deux sommets correspond à considérer non pas les **graphes simples** qui relient deux sommets par au plus une arête ou un arc, mais également les **multigraphes**, où on autorise plus d'une arête entre deux sommets. C'est ce que nous ferons dans cette section 1.4.3.

Remarque 9 *Les réseaux de niveau 0 sont les arbres phylogénétiques binaires enracinés. Les réseaux de niveau 1 ont été introduits auparavant sous le nom de "galled trees" [Gusfield* et al.*, 2004] en référence aux troncs ou branches d'arbres qui présentent des loupes ("gall", en anglais), c'est-à-dire des excroissances dont la forme circulaire peut rappeler celle des blobs d'un réseau de niveau 1. Ce sont des graphes planaires extérieurs.*

On peut également définir les réseaux de niveau 1 en disant qu'il s'agit de réseaux enracinés explicites binaires dont tous les cycles du graphe non orienté sous-jacent ne partagent aucun sommet, condition apparue dans [Ma *et al.*, 2000]. Cette définition a l'avantage de s'étendre directement au contexte non enraciné [Semple et Steel, 2006] alors que pour les niveaux strictement supérieurs à 2, aucune adaptation non enracinée n'avait été définie avant ce travail de thèse (nous en proposons une dans la section 1.4.4).

Remarque 10 *Le plus petit ancêtre commun de deux sommets est unique dans un arbre ou un réseau de niveau 1, mais ce n'est pas nécessairement le cas dans un réseau de niveau 2 : dans le réseau* N *de la figure 1.6 page 28,* c_1 *et* c_2 *ont deux plus petits ancêtes communs : le grand-parent de* x_1 *et le grand-parent de* x_2.

b) Décomposition des réseaux de niveau k

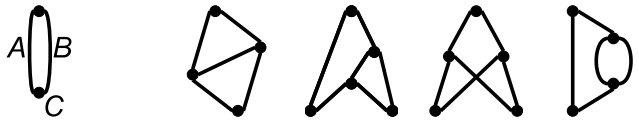
Le paramètre de niveau permet d'évaluer à quel point la structure du réseau s'éloigne de celle d'un arbre. Toutefois cette structure générale d'arbre, qui est proche de celle de l'**arbre des blocs** de la décomposition en blocs des graphes, n'a pas été explicitée par les auteurs qui ont étudié les réseaux phylogénétiques de niveau k. Elle est pourtant intéressante à plus d'un titre : non seulement pour donner une image globale simplifiée du réseau, mais aussi parce qu'à niveau borné, les blobs peuvent y être codés par un ensemble fini de **générateurs**, motifs de graphes qui constituent une sorte d'alphabet de construction des réseaux de niveau k, et que nous définissons maintenant.

Définition 1.16 *[van Iersel* et al.*, 2009a] Un générateur de niveau* k *est un réseau phylogénétique strictement de niveau* k *sans isthme.*

La liste des générateurs de niveau 1 et 2 est présentée en figure 1.16 (le générateur de niveau 0 est un unique sommet isolé). Les **côtés** d'un générateur sont ses arcs et ses sommets hybrides de degré sortant 0.

Les générateurs de niveau k avaient initialement été introduits comme base de la structure des réseaux simples de niveau k [van Iersel *et al.*, 2009a]. Nous montrons maintenant qu'ils peuvent servir à décomposer les réseaux de niveau k en toute généralité.

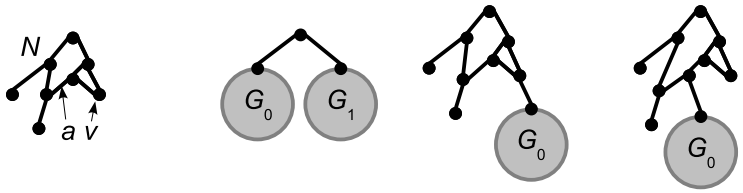
Définition 1.17 *Étant donné un ensemble* S_k *de générateurs de niveau au plus* k*, et un réseau phylogénétique* N*, on définit les règles suivantes, illustrées en figure 1.17 :*



$\mathcal{G}^1$ $\mathcal{G}^2_a$ $\mathcal{G}^2_b$ $\mathcal{G}^2_c$ $\mathcal{G}^2_d$

FIGURE 1.16 : L'unique générateur $\mathcal{G}^1$ de niveau 1, et les quatre générateurs de niveau 2 : $\mathcal{G}^2_a$, $\mathcal{G}^2_b$, $\mathcal{G}^2_c$ et $\mathcal{G}^2_d$.

- $\mathtt{MergeRoot}_k(G_0, G_1)$ *est obtenu en accrochant les générateurs* G_0 *et* $G_1 \in S_k$ *sous une racine.*
- $\mathtt{Attach}_k(v, G, N)$ *est le réseau obtenu en ajoutant un arc d'un sommet hybride* $v \in N$ *de degré sortant 0 vers une copie d'un générateur* $G \in S_k$.
- $\mathtt{Attach}_k(a, G, N)$ *est le réseau obtenu en subdivisant l'arc* a *et en ajoutant un arc depuis le sommet ainsi créé vers une copie de* $G \in S_k$.



(a) (b) (c) (d)

FIGURE 1.17 : Règles de construction d'un réseau de niveau k à partir de générateurs de niveau au plus k : un réseau phylogénétique N (a), et le réseau obtenu en appliquant $\mathtt{MergeRoot}_k(G_0, G_1)$ (b), $\mathtt{Attach}_k(v, G_0, N)$ (c), et $\mathtt{Attach}_k(a, G_0, N)$ (d).

Notons que la règle $\mathtt{MergeRoot}_k$ ne peut être utilisée qu'une fois, et qu'elle est utilisée pour les réseaux de niveau k dont la racine est un sommet d'articulation.

Théorème 1 N *est un réseau de niveau* k *si et seulement si il existe une suite de* $r \in \mathbb{N}$ *localisations (arcs ou sommets hybrides)* $(\ell_j)_{j\in[1,r]}$ *et une suite de générateurs* $(G_j)_{j\in[0,r]}$ *de* S_k *tels que :*

$$\begin{aligned} N = \mathtt{Attach}_k(&\ell_r, G_r, \mathtt{Attach}_k(\dots \\ &\mathtt{Attach}_k(\ell_2, G_2, \mathtt{Attach}_k(\ell_1, G_1, G_0))\dots)), \\ \textit{ou } N = \mathtt{Attach}_k(&\ell_r, G_r, \mathtt{Attach}_k(\dots \\ &\mathtt{Attach}_k(\ell_2, G_2, \mathtt{MergeRoot}_k(G_1, G_0))\dots)). \end{aligned}$$

Démonstration. $\Leftarrow$: L'implication est triviale car toutes les règles ci-dessus, pour tout générateur G_j de niveau i tel que $i \leq k$, créent un isthme vers un nouveau blob à k sommets hybrides ou moins, ce qui produit un réseau de niveau k.

$\Rightarrow$: Nous prouvons par récurrence sur p, que pour tout k, un réseau N de niveau k à p sommets peut être obtenu par des applications successives de la règle $Attach$, après une application éventuelle de la règle $MergeRoot$.

Fixons k.

Initialisation : si $p = 1$ alors le seul réseau possible est $\mathcal{G}^0$, ce qui correspond à ne pas appliquer (en choisissant $r = 0$ dans la définition) la règle $Attach$ au générateur $\mathcal{G}^0$ de niveau 0.

Récursion : supposons maintenant que tous les réseaux à strictement moins de p sommets vérifient la propriété voulue, et soit N un réseau à p sommets.

Si N contient une feuille l, alors :

- soit elle a au moins un grand-parent u, alors :
  - soit son parent est un sommet de spéciation l' dont l'autre fils est v, auquel cas on supprime l et l', et on ajoute un arc de u vers v. Le réseau N' obtenu a moins de p sommets, donc l'hypothèse de récurrence s'applique, et
  $$N = Attach_k((u,v), \mathcal{G}^0, N'),$$
  ce qui donne la propriété voulue.
  - soit son parent est un sommet hybride h, auquel cas on supprime l. Le réseau N' obtenu a moins de p sommets, donc l'hypothèse de récurrence s'applique, et $N = Attach_k(h, \mathcal{G}^0, N')$, ce qui donne la propriété voulue.
- soit elle n'a pas de grand-parent, c'est-à-dire que son parent est la racine ρ. Alors le réseau N', obtenu en supprimant l et ρ, a moins de p sommets, donc on peut lui appliquer l'hypothèse de récurrence et obtenir :
  - soit
  $$N' = Attach_k(\ell_r, G_r, Attach_k(\dots Attach_k(\ell_2, G_2, Attach_k(\ell_1, G_1, G_0))\dots)),$$
  auquel cas
  $$N = Attach_k(\ell_r, G_r, Attach_k(\dots Attach_k(\ell_2, G_2, Attach_k(\ell_1, G_1, MergeRoot_k(\mathcal{G}^0, G_0)))\dots));$$
  - soit
  $$N' = Attach_k(\ell_r, G_r, Attach_k(\dots Attach_k(\ell_2, G_2, MergeRoot_k(G_1, G_0))\dots))$$
  auquel cas

$$N = Attach_k(\ell_r, G_r, Attach_k(\ldots \\ Attach_k(\ell_2, G_2, Attach_k(\ell', G_1, MergeRoot_k(G_0, \mathcal{G}^0)))\ldots)),$$

où ℓ' est l'arc reliant la racine à G_0 dans $MergeRoot_k(G_0, \mathcal{G}^0)$.

Sinon, N ne contient qu'une racine, des sommets de degré entrant 2, et des sommets de degré entrant 1 et de degré sortant 2 :

- Soit N est sans isthme, alors c'est un générateur, et il a au plus k sommets hybrides (comme N est de niveau k), donc la propriété voulue est vraie.
- Soit N contient au moins un isthme. Considérons un blob B dont les sommets n'ont pas de descendant en dehors de ce blob. B est donc un générateur de niveau k, on le traite comme la feuille l ci-dessus en remplaçant $\mathcal{G}^0$ par B dans les formules de décomposition.

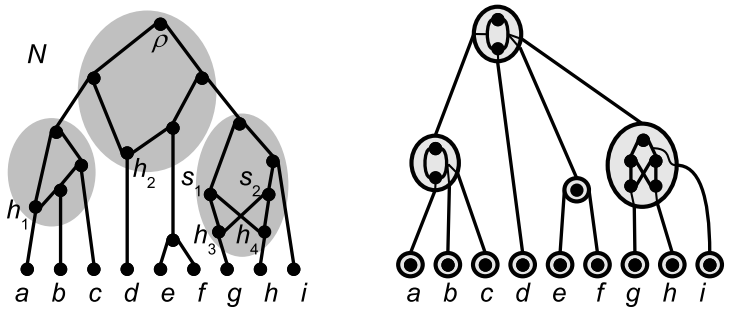
□

Le théorème 1 représente les réseaux de niveau k par une suite de règles sur un ensemble fini de générateurs. Sous cette forme, la caractérisation n'implique pas la canonicité de la représentation : deux suites de règles différentes peuvent conduire à un même réseau phylogénétique (typiquement, en changeant simplement l'ordre dans lequel les règles sont appliquées). Toutefois, l'enchaînement de ces règles induit une description de tout réseau de niveau k par un arbre de décomposition en générateurs, montré en figure 1.18(b). Il correspond globalement à l'arbre de décomposition en composantes biconnexes du graphe, avec un intérêt supplémentaire dans notre cas : pouvoir étiqueter les sommets de l'arbre de décomposition par un générateur extrait d'un ensemble fini (à niveau fixé). Ainsi, une application possible de cet arbre de décomposition serait de l'utiliser, avec un étiquetage approprié, comme un codage bijectif des réseaux de niveau k (k étant fixé), qui constituerait la base d'une formule d'énumération, en complément de celle donnée par Semple et Steel [2006] pour les réseaux non enracinés de niveau 1.

c) Décomposition des générateurs de niveau k

Dans [van Iersel *et al.*, 2009a], les générateurs de niveau 2 sont identifiés par une analyse de cas, qui est généralisée par un algorithme de calcul des 65 générateurs de niveau 3 [Kelk, 2008]. Nous donnons maintenant deux règles R1 et R2 pour calculer les générateurs de niveau $(k+1)$ à partir des générateurs de niveau k. Elles permettent, comme nous le verrons en section 3.2, de fournir un algorithme incrémental pour construire les générateurs de niveau k, qui en pratique parvient à construire tous ceux de niveaux 4 et 5.

Définition 1.18 *Soit* N *un générateur de niveau* k. *On définit l'ordre partiel* $\succ_N$ *sur ses côtés : étant donné deux côtés* X *et* Y *de* N, $Y \succ_N X$ *si on peut accéder à la source de l'arc* Y *(ou à* Y *lui-même, si c'est un sommet) depuis la cible de l'arc* X, *ou* X *lui-même si* X *est un sommet.*



(a) (b)

FIGURE 1.18 : Un réseau phylogénétique de niveau 2 (a) et son arbre de décomposition en générateurs (b) : la numérotation sur les arcs de l'arbre de décomposition indique dans quel ordre les générateurs sont attachés aux arcs du générateur du sommet père.

Règle R1 : *le réseau* $R1(N,X,Y)$ *est obtenu en choisissant deux côtés* X *et* Y *de* N*, tel que si* $X = Y$*,* X *n'est pas un sommet hybride, et en attachant un nouveau sommet hybride à* X *et* Y*, comme montré en figures 1.19(b-e).*

Règle R2 : *le réseau* $R2(N,X,Y)$ *est obtenu en choisissant un côté* X *de* N *et un arc* $Y \not\succ_N X$ *de* N*, et en ajoutant un arc de* X *vers* Y*, créant ainsi un nouveau sommet hybride dans l'arc* Y*, comme montré en figures 1.19(f-h).*

Notons que les deux côtés X et Y ont un rôle symétrique pour la règle R1 mais pas pour R2. Quand on construit $R1(N,X,Y)$ depuis N, on dit qu'on applique la règle R1 sur X et Y, de même pour R2. Notons aussi que dans la définition de $R1(N,X,X)$, X doit être soit un arc, soit dans le cas particulier où $N = \mathcal{G}^0$, son unique sommet.

Proposition 5 *Soit* N*, un générateur de niveau* k *et* X *et* Y *deux côtés de* N *tels que* $N_1 = R1(N,X,Y)$*, (respectivement* $N_2 = R2(N,X,Y)$*) est bien défini. Alors* N_1 *(respectivement* N_2*) est un générateur de niveau* $(k+1)$*.*

Démonstration. Comme N_1 et N_2 sont bien définis, les côtés X et Y respectent les conditions respectives sur les règles R_1 et R_2 de la définition 1.18. Ces définitions impliquent que l'acyclicité du réseau est préservée. Ainsi, il reste à montrer que pour tout type de côté (arc ou sommet hybride) X et Y, l'application des règles R1 et R2 ajoute toujours des sommets de spéciation et exactement un sommet hybride de degré sortant ≤ 1.

Vérifions-le tout d'abord quand la règle R1 est appliquée pour obtenir $R1(N,X,Y)$:

- si $N = \mathcal{G}^0$, alors appliquer R1 donne le générateur $\mathcal{G}^1$ de niveau 1.
- si X et Y sont des sommets hybrides distincts, ils ont degré sortant 0 puisqu'ils sont des côtés de N, donc l'application de R1 leur donnera un degré sortant de 1, et créera un nouveau sommet hybride de degré sortant 0 (figure 1.19(b)).

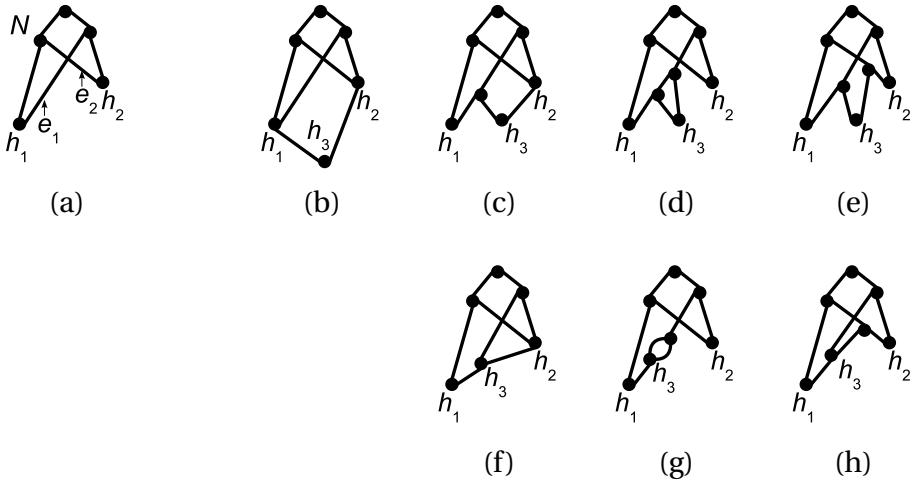


FIGURE 1.19 : Résultats d'application des règles R1 et R2 sur un générateur N de niveau 2 (a), en fonction du type de côté (arc ou sommet hybride) où elles sont appliquées : $R1(N,h_1,h_2)$ (b), $R1(N,e_1,h_2)$ (c), $R1(N,e_1,e_1)$ (d), $R1(N,e_1,e_2)$ (e), $R2(N,h_2,e_1)$ (f), $R2(N,e_1,e_1)$ (g), $R2(N,e_2,e_1)$ (h). Dans tous les cas un nouveau sommet hybride h_3 est créé.

- si X est un sommet hybride, et qu'Y est un arc, alors appliquer R1 donne un degré sortant de 1 à X, ajoute un nouveau sommet hybride de degré sortant 0 et crée un nouveau sommet de spéciation "à l'intérieur" de l'arc Y (dont le parent est la source de Y et dont les enfants sont la cible de Y et le sommet hybride nouvellement créé), comme montré en figure 1.19(c). Par symétrie, on obtient également un générateur valide si X est un arc et Y un sommet hybride.
- si X et Y sont tous deux des arcs (éventuellement le même arc comme en figure 1.19(d)) alors appliquer R1 crée deux sommets de spéciation, l'un à l'intérieur de X et l'autre à l'intérieur de Y (figure 1.19(e)).

Dans tous les cas, on obtient $R1(N,X,Y)$ à partir de N en ajoutant un sommet hybride et éventuellement des sommets de spéciation. Ainsi, $R1(N,X,Y)$ respecte la définition d'un réseau strictement de niveau $(k+1)$. Comme la transformation préserve l'absence d'isthme, $R1(N,X,Y)$ est un générateur de niveau $(k+1)$.

Vérifions-le maintenant quand la règle R2 est appliquée pour obtenir $R2(N,X,Y)$:

- si X est un sommet hybride, et qu'Y est un arc, alors appliquer R2 donne un degré sortant de 1 à X, et crée un nouveau sommet hybride de degré sortant 1 à l'intérieur d'Y (dont les parents sont X et la source de Y, et dont l'enfant est la cible de Y), comme montré en figure 1.19(f).
- si X et Y sont tous deux des arcs (éventuellement le même, voir figure 1.19(g)) alors appliquer R2 crée un sommet de spéciation à l'intérieur de X et un sommet hybride de degré sortant 0 à l'intérieur de Y (figure 1.19(h)).

Dans tous les cas, on obtient $R2(N,X,Y)$ à partir de N en ajoutant un sommet hybride et éventuellement des sommets de spéciation. Ainsi, $R2(N,X,Y)$ est un générateur de niveau k+1. □

Nous avons vu dans la proposition 5 qu'il est possible de construire des générateurs de niveau $(k+1)$ à partir des générateurs de niveau k, il reste à montrer que tout générateur de niveau $(k+1)$ peut être construit de cette façon.

Nous allons tout d'abord définir une opération de suppression d'un sommet hybride, puis montrerons dans le lemme 1 que cette opération, une sorte d'inverse de R1 et R2, permet d'obtenir un générateur de niveau k à partir de tout générateur de niveau $(k+1)$.

Définition 1.19 *Soit N un réseau phylogénétique de niveau $(k+1)$, et v un sommet de N, qui n'est pas un enfant de la racine (sauf éventuellement si $N = \mathcal{G}^1$). On définit de la manière suivante la R1R2-**suppression** de v, qui fournit un réseau N' de niveau k. Le sommet v est tout d'abord supprimé du réseau, tout comme ses arcs incidents. Puis, dans plusieurs cas, des arcs sont ajoutés au réseau pour maintenir sa connectivité.*

Si v a degré sortant 0, alors cinq cas se présentent (voir figure 1.20 (a)-(e)) :

(a) les parents de v sont des sommets hybrides distincts X et Y, alors après suppression de v, les sommets X et Y ont degré sortant 0, et aucun autre sommet n'est modifié, comme montré en figure 1.20(a). En appelant N' le réseau obtenu après suppression, on note $N = R1(N',X,Y)$,

(b) un parent de v, disons Y, est un sommet hybride, et l'autre, X, est un sommet de spéciation, alors, comme montré en figure 1.20(b), après suppression de v, X, et ajout d'un arc e_X depuis le parent de X vers le second enfant de X (autre que v), on obtient un réseau N' tel que $N = R1(N',Y,e_X)$,

(c) les parents de v sont des sommets de spéciation, X et Y, tels que X n'est ni enfant ni parent de Y, comme en figure 1.20(c). Alors, après suppression de v, X, Y, ajout d'un arc e_X du parent de X au second enfant de X (autre que v), et d'un arc e_Y du parent de Y au second enfant de Y (autre que v), on obtient un réseau N' tel que $N = R1(N',e_X,e_Y)$.

(d) les parents de v sont des sommets de spéciation, X et Y, tels que X est le parent de Y, comme en figure 1.20(d). Alors, après suppression de v, X, Y, et ajout d'un arc e_{XY} du parent de X au second enfant de Y (autre que v), on obtient un réseau N' tel que $N = R1(N',e_{XY},e_{XY})$.

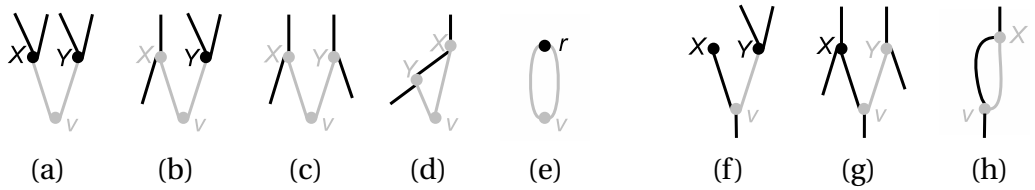
(e) v est le seul enfant de la racine. Alors N est forcément $\mathcal{G}^1$, comme montré en figure 1.20(e), on supprime alors v et ses deux arcs incidents pour obtenir le générateur $N' = \mathcal{G}^0$ de niveau 0 avec un sommet A_0 et $N = R1(N',A_0,A_0)$.

Si v a degré sortant 1, alors trois cas se présentent (voir figure 1.20 (f)-(h)) :

(f) au moins un parent de v, disons Y, est un sommet hybride. Alors, après suppression de v et ajout d'un arc e_X de X à l'enfant de v, le sommet Y a degré sortant 0 et le degré d'aucun autre sommet n'est changé, comme montré en figure 1.20(f), on obtient un réseau N' tel que $N = R2(N',Y,e_X)$,

(g) les deux parents de v sont des sommets de spéciation distincts X et Y. Alors, comme montré en figure 1.20(g), après suppression de v, Y, et ajout d'un arc e_Y du parent de Y au second enfant de Y (autre que v), on obtient un réseau N' tel que $N = R2(N', e_Y, e_X)$.

(h) v a seulement un parent, le sommet de spéciation X. Alors, comme montré en figure 1.20(h), après suppression de v, X, et ajout d'un arc e_X du parent de X au enfant de v, on obtient un réseau N' tel que $N = R2(N', e_X, e_X)$.



(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 1.20 : Différents cas possibles pour l'inversion des règles R1 et R2, en fonction du sommet hybride v créé par la règle : de degré sortant 0 (a-e) pour la règle R1 ou de degré sortant 1 (f-h) pour la règle R2. Les arcs et sommets gris sont supprimés lors de l'inversion de la règle.

Notons que dans tous les cas, les sommets de N' respectent les conditions sur les degrés de la définition d'un réseau phylogénétique binaire enraciné.

L'utilisation des noms R1 et R2 ci-dessus est un abus de notation car nous n'avons aucune garantie, en écrivant $N = Ri(N', X, Y)$ dans cette définition des inversions de règles, que N' est un générateur, ni que X ou Y sont des côtés. Toutefois, nous ne les utiliserons que lorsque ce sera effectivement le cas.

Lemme 1 *Soit N un générateur de niveau $(k+1)$. Il existe un sommet v de N tel que la R1R2-suppression de v dans N donne un générateur de niveau k.*

Démonstration. Prouvons-le par récurrence.

Initialisation :

Appelons A_0 le seul sommet de $\mathcal{G}^0$, A et B les deux arcs de $\mathcal{G}^1$ et C son sommet hybride. On peut alors vérifier l'initialisation pour $k \leq 1$ (voir figure 1.16) car $\mathcal{G}^1 = R1(\mathcal{G}^0, A_0, A_0)$ (le sommet C est R1R2-supprimé, on est dans le cas (e)), et pour $k = 1$ on supprime les sommets hybrides dont aucun parent n'est la racine : $\mathcal{G}^2_a = R1(\mathcal{G}^1, B, C)$ (b), $\mathcal{G}^2_b = R1(\mathcal{G}^1, B, B)$ (d), $\mathcal{G}^2_c = R1(\mathcal{G}^1, A, B)$ (c) et $\mathcal{G}^2_d = R2(\mathcal{G}^1, B, B)$ (h).

Récursion :

Fixons $k \geq 2$. Supposons que la propriété voulue est vraie pour tout générateur de niveau j, avec $j < k$, et prouvons-la pour le niveau k. Ainsi, considérons un générateur N de niveau $(k+1)$, et prouvons qu'il existe un sommet de N tel que la R1R2-suppression

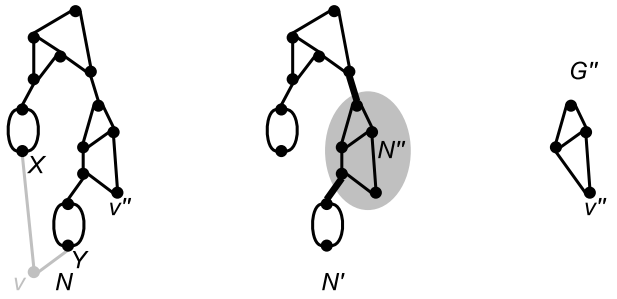
de v dans N donne un générateur de niveau k. N contient au moins trois sommets hybrides, donc au moins l'un des trois, disons v, n'est pas un enfant de la racine. On le R1R2-supprime alors, pour obtenir un **réseau** N' de niveau k.

Nous devons maintenant prouver que soit N' est un **générateur** de niveau k, soit qu'il était possible de choisir un autre sommet hybride v'' dont la R1R2-suppression aurait donné un générateur de niveau k.

Si N' ne contient pas d'isthme, alors c'est un générateur de niveau k par définition. Supposons donc que N' contient au moins un isthme, comme illustré en en figure 1.21(b). Soit N'' un blob de N' qui ne contient pas la racine de N'. Notons que selon la définition 1.19, la R1R2-suppression de v dans N n'a créé aucune feuille dans N', et donc N'' est un blob non trivial, donc il a au moins un sommet hybride.

Si N'' est un générateur de niveau j, avec $0 < j < k$, nous l'appelons G''. Sinon, ce n'est même pas un réseau de niveau k à cause de la condition sur les degrés. Nous affirmons en effet que dans ce cas, N'' contient exactement un sommet de degré entrant et sortant 1. En effet, N'' n'est connecté que par des isthmes au reste du réseau N'. Comme N ne contient pas d'isthme, la présence de plusieurs blobs dans N' provient de la R1R2-suppression de v. Comme v a degré entrant 2, on n'a supprimé que deux arcs en supprimant v, et donc modifié le degré d'au plus deux sommets de N'. N' contient donc au plus deux isthmes incidents à des sommets de N'', comme montré en figure 1.21(b). La cible de l'un de ces deux isthmes est la racine de N'' car N'' ne contient pas la racine de N', donc il existe dans N' au plus un isthme dont la source est dans N''.

Ceci est le seul cas problématique qui empêche N'' d'être un générateur, puisqu'il correspond à la présence d'un sommet de degré entrant et sortant 1 dans le réseau N'' considéré indépendamment du reste du réseau N'. Dans ce cas, considérons le générateur G'' de niveau j obtenu en supprimant ce sommet et en reliant son parent à son enfant dans N'', comme illustré en figure 1.21(c).



(a) (b) (c)

FIGURE 1.21 : Si suite à la suppression de v dans le réseau N (a) on obtient un réseau N' qui contient au moins un isthme (b), alors on trouve un autre sommet v'' (c) qui peut être R1R2-supprimé de N afin d'obtenir un générateur de niveau $(k-1)$.

Dans les deux cas, on applique l'hypothèse de récurrence sur ce générateur G'' de niveau j : comme $j > 0$, il contient un sommet hybride v'' qui peut être R1R2-supprimé. Même si la R1R2-suppression de v'' dans G'' fournit un générateur de niveau $(j-1)$ valide (qui ne contient pas d'isthme), il reste à prouver que la R1R2-suppression de v'' dans N fournit un générateur de niveau $(k-1)$ valide. La figure 1.21(c) montre que cela n'est pas toujours évident : dans cet exemple, un des parents de v'' dans G'' est un sommet de spéciation et l'autre est un sommet hybride, ce qui correspond au cas (b) de R1R2-suppression de v'', alors que dans N, les deux parents de v'' sont des sommets de spéciation, ce qui correspond au cas (c). Ainsi, il faut déterminer quel cas de R1R2-suppression s'applique dans N en fonction de celui qui s'applique dans G''.

Soient X'' et Y'', les parents de v'' dans G'' (on utilisera les noms X'' et Y'' de la même manière que X et Y dans la définition de la R1R2-suppression). On rappelle que tous les cas sont illustrés en figure 1.20. Quand on fera référence aux différents cas dans G'', on les fera suivre par une étoile "$*$".

Considérons tout d'abord le cas $(a*)$. Si v'' a aussi degré sortant 0 dans N, alors :
- si les deux parents de v'' dans N sont des sommets hybrides, alors on R1R2-supprime v'' dans N en suivant le cas (a).
- si exactement un des parents de v'' dans N n'est pas un sommet hybride, alors on R1R2-supprime v'' dans N en suivant le cas (b).
- si aucun parent de v'' dans N n'est un sommet hybride, alors on R1R2-supprime v'' dans N en suivant le cas (c).

Sinon, v'' a degré sortant 1 dans N, et donc on R1R2-supprime v'' en suivant le cas (f). Notons que v'' a forcément au moins un de ses deux parents dans N qui est un sommet hybride. En effet, au moins un des deux sommets X'' et Y'' reste parent de v'' dans N : comme il y a déjà un arc de source v'' dans N, il y a au plus un autre arc dont la source est dans N'' et la cible dans N mais pas dans N'' (soit un arc de cible v, soit un isthme de N'), et donc qui implique la présence d'un sommet de spéciation qui s'intercale dans N entre v'' et un de ses parents dans N'', X'' et Y''.

Considérons maintenant le cas $(b*)$. Si v'' a aussi degré sortant 0 dans N, alors on procède de façon similaire au cas (a). Sinon, v'' a degré sortant 1 dans N :
- si l'un des parents de v'' dans N est un sommet hybride, alors on R1R2-supprime v'' dans N en suivant le cas (f).
- sinon on R1R2-supprime v'' dans N en suivant le cas (g).

Considérons maintenant le cas $(c*)$. Si v'' a aussi degré sortant 0 dans N alors :
- si exactement un de ses parents dans N est un sommet hybride, alors on R1R2-supprime v'' dans N en suivant le cas (b).
- sinon, on R1R2-supprime v'' dans N en suivant le cas (c).

Sinon, v'' a degré sortant 1 dans N, et on procède de façon similaire au cas $(b*)$.

Considérons maintenant le cas $(d*)$. Si v'' a aussi degré sortant 0 dans N alors, si X'' et Y'' sont encore les parents de v'' dans N alors on R1R2-supprime v'' dans N en suivant le cas (d), sinon :
- si exactement un des deux parents de v'' dans N est un sommet hybride, alors on R1R2-supprime v'' dans N en suivant le cas (b).
- sinon on R1R2-supprime v'' dans N en suivant le cas (c).

Sinon, v'' a degré sortant 1 dans N, et on procède de manière similaire au cas $(b*)$.

Considérons maintenant le cas $(e*)$. Si v'' a aussi degré sortant 0 dans N alors le cas où v'' a encore un seul parent dans N ne peut se produire (sinon N contiendrait un isthme), donc :
- si exactement un des deux parents de v'' dans N est un sommet hybride, alors on R1R2-supprime v'' dans N en suivant le cas (b).
- sinon on R1R2-supprime v'' dans N en suivant le cas (c).

Sinon, v'' a degré sortant 1 dans N. Si v'' a encore un seul parent dans N alors on R1R2-supprime v'' en suivant le cas h. Sinon, on procède de manière similaire au cas $(b*)$.

Considérons les cas $(f*)$ et $(g*)$:
- si l'un des parents de v'' dans N est un sommet hybride, alors on R1R2-supprime v'' dans N en suivant le cas (f).
- sinon on R1R2-supprime v'' dans N en suivant le cas (g).

On considère enfin le cas $(h*)$: si v'' a encore un seul parent dans N, alors on R1R2-supprime v'' en suivant le cas (h), sinon, on procède de même qu'au cas $(f*)$.

On peut aussi vérifier que dans tous les cas, la R1R2-suppression de v'' dans N préserve l'absence d'isthme assurée quand on R1R2-suprime v'' dans G''.

Dans tous les cas, nous avons trouvé un sommet hybride qui peut être supprimé pour obtenir un générateur de niveau k. Ainsi, la proposition est vérifiée. □

On en déduit directement le corollaire suivant.

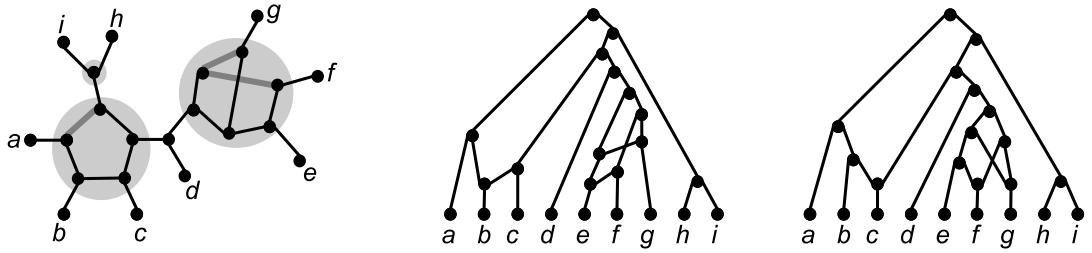
Corollaire 1.20 *Pour tout générateur* N *de niveau* $(k+1)$*, il existe un générateur* N' *de niveau* k*, et des côtés* X *et* Y *de* N' *tels que* $N = R1(N', X, Y)$ *ou* $N = R2(N', X, Y)$.

1.4.4 Réseaux non enracinés de niveau k

a) Définition

Nous étendons la définition des réseaux (enracinés) de niveau k aux réseaux non enracinés. Comme le concept de sommet de spéciation et de sommet hybride n'existe pas dans les réseaux explicites non enracinés, nous proposons la définition suivante avant de montrer dans la proposition 6 son lien avec la définition du niveau en contexte enraciné.

Définition 1.21 *Un **réseau non enraciné** X **de niveau** k sur un ensemble X de taxons est un réseau phylogénétique explicite binaire non enraciné tel qu'un arbre connectant tous les sommets de N peut être obtenu en supprimant au plus k arêtes par blob, comme illustré en figure 1.22(A).*



(A) (B) (C)

FIGURE 1.22 : Un réseau non enraciné N de niveau 2 sur l'ensemble de taxons $\{a,b,c,d,e,f,g,h,i\}$ (A). Tous les sommets non étiquetés sont des sommets internes, les zones grises sont des exemples de blobs et les arcs en gras sont tels que leur suppression transforme N en arbre. Tout enracinement de ce réseau - (B) ou (C) par exemple - est un réseau de niveau 2.

Remarque 11 *Un réseau non enraciné de niveau 0 est un arbre phylogénétique binaire non enraciné. Précisons que la définition des réseaux non enracinés de niveau 1 correspond à celle introduite des "unrooted binary galled trees" par Semple et Steel [2006] où une formule de dénombrement de ces objets est donnée en fonction de leur nombre de feuilles, d'arêtes et de blobs.*

Précisons enfin que la discussion sur les restrictions à apporter à la définition des réseaux enracinés de niveau k s'applique aussi pour les réseaux non enracinés de niveau k. Nous considérons donc désormais que dans ces réseaux un blob a au moins 4 sommets.

b) Enracinement de réseaux de niveau k non enracinés

Dans cette section, nous illustrons le fait qu'il existe plusieurs manières d'enraciner un réseau non enraciné de niveau k. Nous donnons tout d'abord une définition pour décrire formellement l'opération qui permet d'obtenir un réseau enraciné de niveau k à partir d'un réseau non enraciné de niveau k.

Définition 1.22 ***Enraciner** un réseau $N=(V,E)$ non enraciné de niveau k consiste à obtenir un réseau phylogénétique binaire enraciné $N'=(V\cup\{r\},A)$ de la façon suivante :*

(i) *choisir une arête xy de N et la subdiviser pour créer un sommet r de degré 2, voisin de x et y, qui deviendra la racine ;*

(ii) *choisir un ordre* $\sigma : V \cup \{r\} \to [0..|V|]$ *(cet ordre est une extension linéaire de la relation de descendance dans le réseau enraciné* N'*) tel que* $\sigma(r) = 0$ *et* $\forall u \in V, \exists v \in V \cup \{r\}$ *tel que :*
- $uv \in E$,
- *et* $\sigma(v) < \sigma(u)$ *(c'est-à-dire que* v *est un parent de* u *dans* N'*),*
- *et si* u *a degré 3 dans* N*, alors il existe au moins un sommet* $v' \in V \cup \{r\}$ *tel que* $uv' \in E$ *et* $\sigma(u) < \sigma(v')$ *(c'est-à-dire que si* u *a degré 3, il a au moins un descendant,* v'*, dans* N'*);*

(iii) *définir l'ensemble d'arcs* $A = \{(u,v)$ *tels que* $uv \in E$ *et* $\sigma(u) < \sigma(v)\}$.

Remarque 12 *Notre définition diffère de celle proposée par Semple et Steel [2006] qui consiste à appliquer seulement l'étape* (i)*. Ce choix se justifie pour la problématique étudiée par les auteurs, le dénombrement des réseaux non enracinés de niveau 1, où cet enracinement apparaît seulement comme une étape technique du calcul. Pour faire le lien avec les réseaux enracinés de niveau* k*, comme nous allons le voir ci-dessous dans le théorème 2, les étapes* (ii) *et* (iii) *de l'enracinement sont nécessaires.*

Quelle que soit la position de racine choisie à l'étape (i), il est important de remarquer que de nombreux enracinements sont possibles, en fonction de l'ordre choisi à l'étape (ii), comme montré en figures 1.22(B) et 1.22(C), et dans la proposition suivante.

Proposition 6 *Pour tout entier* $k \geq 2$*, il existe un réseau non enraciné* N_k *de niveau* k *avec* $2k$ *feuilles tel que* N_k *a au moins* 2^k *enracinements pour lesquels la racine* r *est placée sur la même arête de* N_k.

Démonstration. Nous décrivons tout d'abord comment construire récursivement N_k. Pour construire N_1, considérons un cycle avec 2 sommets v_0^0 et v_0^1 reliés respectivement à une feuille x_0^0 et une feuille x_0^1. Pour construire N_{k+1} à partir de N_k, appelons e_{k-1}^0 l'arête incidente à v_{k-1}^0 et pas à v_{k-1}^1, et e_{k-1}^1 l'arête incidente à v_{k-1}^1 et pas à v_{k-1}^0. Subdivisons ces arêtes (pour constuire N_2 à partir de N_1, comme les deux arêtes relient v_0^0 et v_0^1, on subdivise simplement deux fois l'une des deux), et connectons les deux sommets créés par la subdivision, puis subdivisons deux fois l'arête ainsi créée, pour obtenir deux sommets : v_k^0 (le plus proche de v_{k-1}^0) et v_k^1. Finalement, ajoutons deux feuilles x_k^0 et x_k^1 attachées respectivement à v_k^0 et v_k^1. Par exemple, N_3 est illustré en figure 1.23(a).

Pour vérifier que N_k a au moins 2^k enracinements, montrons comment il est possible d'associer à chaque entier $a = \sum_{i=0}^{k-1} a_i 2^i \in [0..2^k - 1]$ un enracinement de N_k : on enracine N_k de telle sorte que la feuille $x_i^{a_i}$ soit enfant d'un sommet hybride, et que la feuille $x_i^{1-a_i}$ soit enfant d'un sommet de spéciation. Ainsi, l'enracinement de N_3 montré en figure 1.23(b) correspond à l'entier $a = 4 = 0 \times 2^0 + 0 \times 2^1 + 1 \times 2^2$ □

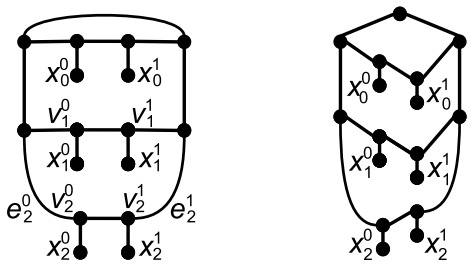


FIGURE 1.23 : Borne inférieure sur le nombre d'enracinements : le réseau non enraciné N_3 de niveau 3 (a) a au moins 2^3 enracinements (b).

Théorème 2 *Tout enracinement d'un réseau non enraciné* N *de niveau* k *produit un réseau* N' *de niveau* k.

Démonstration. A l'étape (iii) du processus d'enracinement, les orientations des arcs sont choisies de telle sorte que si un sommet v peut être atteint par un chemin orienté depuis un sommet u dans N', alors $\sigma(u) < \sigma(v)$. Ainsi, N' est un graphe orienté sans circuit, sinon il contiendrait deux sommets u et v tels que $\sigma(u) < \sigma(v)$ et $\sigma(v) < \sigma(u)$: contradiction.

Vérifions maintenant que N' respecte la condition des degrés des réseaux de niveau k. L'étape (i) du processus d'enracinement assure que N' a une racine. Les étapes (ii) et (iii) garantissent que tout sommet de degré 1 dans N a degré entrant 1 dans N'. Les sommets restants ont degré 3 dans N. Les étapes (ii) et (iii) les forcent à avoir un degré entrant et un degré sortant dans N' tous deux aux moins égaux à 1. Ainsi, selon l'orientation de leur troisième arête incidente, ils deviennent sommets hybrides ou sommets de spéciation dans N'.

Enfin, pour vérifier que N' a au moins k sommets hybrides par blob, concentrons-nous sur un blob B de N tel que la suppression de x arêtes fournisse un arbre T qui connecte tous les sommets de B. On appelle $e(B)$ le nombre d'arêtes et $v(B)$ le nombre de sommets de B après l'enracinement. On appelle h le nombre de sommets hybrides et s le nombre de sommets de spéciation (en excluant la racine) de B après l'enracinement. Alors on a $v(B) = 1 + h + s$. Si l'on considère les arcs dont les sommets de B sont la cible, après l'enracinement, on a $e(B) = 2h + s$. Ainsi, $h = e(B) - v(B) + 1$. Remarquons maintenant que le nombre de sommets avant l'enracinement peut être soit $v(B) - 1$ si la racine de N appartient à B, soit $v(B)$, sinon. Mais comme on a une propriété similaire pour le nombre d'arêtes avant enracinement, le nombre $e(B) - v(B)$ reste constant avant et après l'enracinement, donc considérons ce nombre avant l'enracinement. Comme l'arbre T contient tous les sommets de B, il a $v(B) - 1$ arêtes. On sait également que B a x arêtes de plus que T, donc $e(B) = v(B) - 1 + x$. Finalement, on conclut que $h = e(B) - v(B) + 1 = x$, ce qui prouve que le niveau de N est égal au niveau de son enracinement N'. Ceci conclut la

preuve. □

1.4.5 Autres restrictions de réseaux phylogénétiques explicites

Nous présentons maintenant d'autres restrictions sur les réseaux phylogénétiques explicites orientés (que nous désignerons simplement par le terme "réseaux" dans la suite de cette section), qui ont été introduites en particulier pour obtenir des algorithmes de meilleure complexité comme nous le verrons dans les chapitres suivants. Les relations entre ces diverses restrictions seront précisées en section 1.5.

Un réseau N est **régulier** [Baroni *et al.*, 2004] s'il vérifie les conditions suivantes pour tous sommets u et v :

- $C_N(u) \neq C_N(v)$,
- $C_N(u) \subseteq C_N(v) \Rightarrow u \preceq v$ (c'est-à-dire que v est un ancêtre de u, comme défini page 17),
- N ne contient pas d'**arc de transitivité** de u à v, c'est-à-dire d'arc $a = (u,v)$ tel que $v \preceq_{N-\{a\}} u$ (il existe un chemin de u à v de longueur strictement supérieure à 1).

De façon équivalente, les réseaux réguliers sont ceux qui sont isomorphes au diagramme de Hasse de leurs clades stricts [Baroni *et al.*, 2004].

Un réseau est **sans fratrie hybride** [Nakhleh, 2004; Cardona *et al.*, 2008c] s'il ne contient aucun sommet hybride dont tous les frères (c'est-à-dire les autres enfants d'un de ses parents) sont des sommets hybrides. Il est **sans descendance hybride** [Cardona *et al.*, 2008c] s'il ne contient aucun sommet dont tous les enfants sont des sommets hybrides.

Un réseau est **normal** [Willson, 2007] s'il ne contient pas d'arc de transitivité, et que depuis tout sommet qui n'est pas une feuille, il existe un chemin vers une feuille de X dont tous les sommets, sauf éventuellement le premier, sont des sommets de spéciation. Enfin, un réseau est **unicyclique** s'il contient exactement un sommet hybride.

1.5 Classification des restrictions sur les réseaux phylogénétiques

Les descriptions précédemment proposées dans la littérature de l'ensemble des diverses classes de réseaux phylogénétiques ont pris plusieurs formes. Les catalogues de méthodes (huit illustrées sur le même jeu de données dans [Posada et Crandall, 2001] et sept dans [Makarenkov *et al.*, 2006]) laissent progressivement apparaître la distinction entre réseaux abstraits et explicites [Morrison, 2005, 2010]. Une hiérarchie récapitulative est proposée pour classer divers types de réseaux [Huson et Klöpper, 2005], organisée selon le type de méthodes et de données en entrée. Elle fait apparaître des classes comme les **réseaux d'hybridation** (des réseaux explicites enracinés construits à partir d'arbres en tentant de minimiser le nombre de sommets hybrides), ou les **réseaux de recombinaison** (des

réseaux explicites enracinés construits à partir de séquences binaires indiquant la présence/absence d'un caractère, ou d'un allèle d'un gène) toutes deux contenues dans la classe des **réseaux réticulés**. Ces dénominations laissant planer un doute sur une organisation en fonction des types de réseaux ou des types de données pour leur reconstruction, le premier livre consacré aux réseaux phylogénétiques est organisé de façon explicite selon les types de données en entrée des algorithmes [Huson *et al.*, 2011].

Nous choisissons donc ici de nous focaliser sur les types de réseaux, et proposons à la fin de cette section des diagrammes récapitulatifs des classes de réseaux phylogénétiques basés uniquement sur leurs propriétés topologiques et combinatoires, sans se préoccuper d'éventuels étiquetages ou des données utilisées pour les construire. Ces diagrammes sont des diagrammes de Hasse des classes de réseaux phylogénétiques, et leur intérêt est de faire apparaître distinctement des inclusions qui traduisent que les propriétés combinatoires d'une classe se transmettent à ses sous-classes. Des parallélismes entre les diagrammes permettent aussi de donner un éclairage différent sur des algorithmes existants pour la reconstruction de réseaux abstraits, et de les réutiliser pour construire des réseaux explicites, comme nous le verrons par exemple à la fin de la section 2.2.2.

Quant à la classification en fonction du type de données utilisées en entrée, nous expliquerons dans la section 4.1.2 comment nous avons choisi d'organiser ces informations au sein d'une bibliographie interactive [Gambette, 2010] pour donner un panorama aussi complet et facile à visiter que possible de l'ensemble des méthodes de reconstruction ou de traitement des réseaux phylogénétiques.

Mais avant de présenter les diagrammes récapitulatifs promis, montrons quelques liens entre des sous-classes de réseaux phylogénétiques explicites, et des sous-classes de réseaux abstraits, qui y apparaîtront.

1.5.1 Hiérarchies faibles, pyramides et niveau 1

Dans [Gambette et Huber, 2010], nous montrons que les clades souples d'un réseau N de niveau 1 forment une hiérarchie faible, car ils sont constitués par l'union des clades de deux arbres contenus dans N (l'un obtenu en sélectionnant le parent de gauche pour tout sommet hybride, et l'autre le sommet de droite). Ce résultat, associé à la machinerie permettant de lier distances et hiérarchies faibles présentée au début de la section 1.4.1, donne une façon d'associer canoniquement une distance à un réseau de niveau 1, qu'il est possible de reconstruire à partir de cette distance. Ce problème est étudié de manière plus approfondie d'un point de vue algorithmique par Chan *et al.* [2006].

Nous donnons ici une proposition plus fine sur l'inclusion des clades des réseaux de niveau 1, pour faire également le lien avec les pyramides et prépyramides, et obtenir un ensemble plus riche d'inclusions intermédiaires entre les ensembles de clades d'un réseau de niveau 1 et les hiérarchies faibles.

Proposition 7 *Étant donné un réseau quelconque* N *de niveau 1, tout ensemble de clades* $\mathcal{C} \subseteq \mathcal{S}(N)$ *est une prépyramide, et* $\mathcal{C}(N)$ *et* $\mathcal{S}(N)$ *sont des pyramides.*

Démonstration. Considérons l'ordre σ des feuilles, en partant d'une feuille arbitraire et dans un sens arbitraire, autour d'une représentation planaire du réseau N dont $\mathcal{C}$ est un ensemble de clades souples. Considérons pour chaque clade $C \in \mathcal{C}$ le plus petit (au sens de la descendance) sommet $v(C)$ qui le représente. Outre la racine qui représente le clade X, deux cas se présentent :

- si $v(C)$ est la cible d'un isthme (comme $v(C_1)$ en figure 1.24) alors C contient exactement tous les taxons descendants de $v(C)$, qui forment donc un intervalle de σ.
- sinon, $v(C)$ appartient à un blob B non trivial de N. Deux cas se présentent alors :
  - soit C contient l'ensemble des taxons descendants du sommet hybride de B (comme C_2 dans la figure 1.24). Alors C est l'ensemble des descendants de $v(C)$ qui forme bien un intervalle de σ.
  - soit C ne contient aucun taxon descendant du sommet hybride h de B (comme C_3 dans la figure 1.24). Alors C est l'ensemble des descendants de $v(C)$ qui ne sont pas descendants de h, qui constitue également un intervalle de σ.

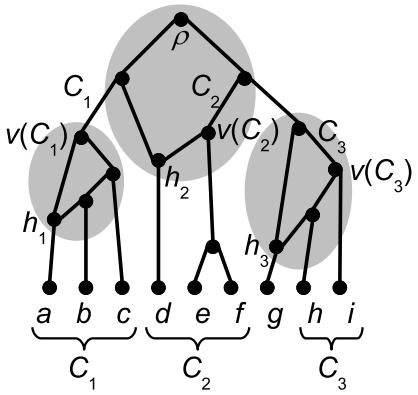


FIGURE 1.24 : Un réseau de niveau 1 et ses clades souples, parmi lesquels $C_1 = \{a, b, c\}$, $C_2 = \{d, e, f\}$ et $C_3 = \{h, i\}$, qui apparaissent comme des intervalles dans l'ordre $\sigma = abcdefghi$ de ses taxons.

On a donc montré que dans tous les cas, un clade contenu dans N apparaît comme un intervalle dans σ, donc tout sous-ensemble de $\mathcal{S}(N)$, et en particulier $\mathcal{C}(N)$, est une prépyramide.

De plus, $\mathcal{C}(N)$ et $\mathcal{S}(N)$ sont clos par intersection non vide. En effet, selon l'analyse de cas ci-dessus, la seule possibilité d'intersection non vide pour deux clades C_1 et C_2 de $\mathcal{C}(N)$ est que $C_1 \subseteq C_2$ (si C_1 est descendant de C_2), $C_2 \subseteq C_1$ (si C_2 descendant de C_1), ou que $v(C_1)$ et $v(C_2)$ appartiennent au même blob B de sommet hybride h sans que l'un

descende de l'autre, auquel cas leur intersection est l'ensemble H de descendants de h, qui est effectivement un clade de $\mathcal{C}(N)$.

En ce qui concerne deux clades C_1 et C_2 de $\mathcal{S}(N)$, il faut considérer une quatrième possibilité : que $v(C_1)$ soit descendant de $v(C_2)$, avec $H \subseteq C_1$ et $H \not\subseteq C_2$ (ou la situation symétrique pour C_1 et C_2). Dans ce cas, $C_1 \cap C_2 = C_1 - H$, c'est-à-dire l'ensemble des taxons descendants de $v(C_1)$ et non descendants de h. Or cet ensemble est également un clade de $\mathcal{S}(N)$, représenté par $v(C_1)$.

Comme de plus les ensembles de clades $\mathcal{C}(N)$ et $\mathcal{S}(N)$ contiennent les singletons et X, ce sont des pyramides. □

D'après la remarque de Bandelt [1992] selon laquelle les prépyramides sont des hiérarchies faibles, un corollaire de cette proposition est que tout ensemble de clades d'un réseau de niveau 1 est une hiérarchie faible.

La proposition 7 ne s'étend pas aux niveaux supérieurs. Par exemple, le réseau N de niveau 2 présenté dans la figure 1.25 montre qu'un résultat analogue à la proposition 7 ne convient pas pour les réseaux de niveau 2, car on peut en construire qui ne sont pas des hiérarchies faibles (et donc pas des prépyramides) : $\{\{a,b,c\},\{a,b,d\},\{b,c,d\}\} \subset \mathcal{S}(N)$ mais $\{a,b,c\} \cap \{a,b,d\} \cap \{b,c,d\} = \{b\}$.

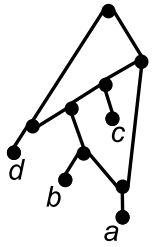


FIGURE 1.25 : Un réseau N de niveau 2 pour lequel $\mathcal{S}(N)$ n'est pas une hiérarchie faible, et a fortiori ni une prépyramide ni une pyramide.

1.5.2 Ensembles circulaires de bipartitions et niveau 1

Dans cette section, nous relions les réseaux non enracinés de niveau 1 et les ensembles circulaires de bipartitions.

Nous introduisons tout d'abord une opération de transformation pour obtenir un réseau simple non enraciné de niveau 1 à partir d'un réseau non enraciné de niveau 1, illustrée en figure 1.8, page 29.

Définition 1.23 *Étant donné un réseau non enraciné* N *de niveau 1, nous appelons* $\mathtt{Simple}(N)$ *un réseau obtenu à partir de* N *de la manière suivante :*

- *comme* N *est planaire extérieur, on considère un ordre* σ *de ses feuilles sur la face extérieure sur une représentation planaire de* N,
- *le graphe* $\mathtt{Simple}(N)$ *est obtenu en attachant les* n *feuilles de* X *adjacentes aux* n *sommets d'un cycle, en respectant l'ordre* σ.

Comme $\mathtt{Simple}(N)$ contient seulement un cycle et des isthmes menant aux feuilles, c'est clairement un réseau simple de niveau 1.

Lemme 2 *Pour tout réseau* $N' = \mathtt{Simple}(N)$, $\mathcal{B}(N) \subseteq \mathcal{B}(N')$

Démonstration. Soit $A|\bar{A}$ une bipartition de N, représentée par une coupe minimale de N, c'est-à-dire soit un isthme x, soit une paire $\{x_1, x_2\}$ d'arêtes d'un même cycle de N. Considérons maintenant la représentation de N utilisée pour construire N', et un ordre σ des feuilles autour de la face extérieure de cette représentation.

Si $A|\bar{A}$ est représentée par une arête e, alors nous dessinons une courbe fermée qui intersecte seulement l'arête x de N, comme en figure 1.26(i). Sinon, nous dessinons une courbe fermée qui intersecte seulement les arêtes x_1 et x_2 de N, comme en figure 1.26(ii). Dans les deux cas, cette courbe sépare la face extérieure de N en deux parties, l'une contenant A et l'autre $\bar{A}$, et l'ensemble des feuilles contenues dans une de ces deux parties apparaît comme un intervalle de σ.

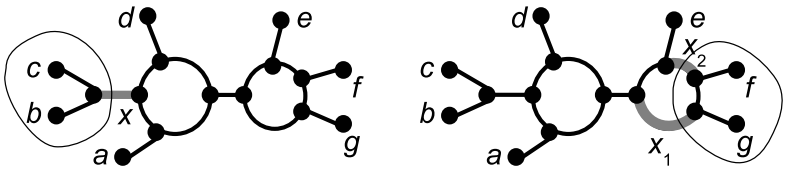


FIGURE 1.26 : Bipartitions d'un réseau non enraciné de niveau 1 dont les feuilles sont dans l'ordre $\sigma = abcdefg$ autour de la face extérieure : $\{b,c\}|\{a,d,e,f,g\}$ est représentée par l'arête x et $\{f,g\}|\{a,b,c,d,e\}$ est représentée par la coupe $\{x_1, x_2\}$.

Ainsi, ces feuilles apparaissent aussi de façon consécutive autour du cycle de N', ce qui implique que $A|\bar{A}$ appartient aussi à $\mathcal{B}(N')$. □

Théorème 3 *Étant donné un ensemble* $\mathcal{B}$ *de bipartitions d'un ensemble* X *de taxons,* $\mathcal{B}$ *est circulaire si et seulement si il existe un réseau simple non enraciné* N *de niveau 1 tel que* $\mathcal{B} \subseteq \mathcal{B}(N)$.

Démonstration. $\Rightarrow$: Comme $\mathcal{B}$ est circulaire, considérons l'ordre σ des taxons de X dans lequel chaque bipartition de $\mathcal{B}$ apparaît comme un intervalle. On construit le réseau

simple non enraciné N de niveau 1 en attachant les feuilles de X à un cycle, en respectant l'ordre σ. Alors, pour toute bipartition $A|\bar{A}$, le sous-graphe de N induit par les sommets de A et leurs voisins est un graphe connexe, qui est connecté par l'intermédiaire de deux arêtes e_1 et e_2 au reste du réseau N, comme montré en figure 1.8(ii), page 29, pour $A = \{b, c, d\}$. Ainsi, $\{e_1, e_2\}$ est une coupe minimale de N qui déconnecte A et $\bar{A}$, donc $A|\bar{A}$ est contenue dans N. Finalement, $\mathcal{B} \subseteq \mathcal{B}(N)$.

$\Leftarrow$: Supposons qu'il existe un réseau non enraciné N de niveau 1 tel que $\mathcal{B} \subseteq \mathcal{B}(N)$. Alors d'après le lemme 2, il existe aussi un réseau simple non enraciné $N' = Simple(N)$ de niveau 1 tel que $\mathcal{B}(N) \subseteq \mathcal{B}(N')$, donc $\mathcal{B}$ est aussi contenu dans N', ce qui prouve que $\mathcal{B}$ est circulaire. □

Nous verrons dans la section 2.2.2 comment utiliser ce théorème pour proposer une heuristique de reconstruction de réseaux phylogénétiques explicites à partir de quadruplets.

1.5.3 Diagrammes récapitulatifs des inclusions de sous-classes

Certaines des inclusions de sous-classes des diagrammes récapitulatifs de la figure 1.27 sont immédiates : les pyramides sont une clôture par intersection d'ensembles non disjoints des prépyramides, tout comme les quasi-hiérarchies par rapport aux hiérarchies faibles. Les k-hiérarchies faibles, ensembles k-compatibles de bipartitions, réseaux médians de dimension k et réseaux de niveau k généralisent trivialement, pour $k \geq 3$, les quasi-hiérarchies (qui sont aussi les 2-hiérarchies faibles), les ensembles 2-compatibles de bipartitions, réseaux médians de dimension 2, et réseaux de niveau 2, respectivement. L'inclusion des prépyramides dans les hiérarchies faibles est notée par Bandelt [1992], et se généralise directement par l'inclusion des pyramides dans les quasi-hiérarchies.

Les ensembles circulaires de bipartitions sont faiblement compatibles (page 73 de Bandelt et Dress [1992a]). Les inclusions entre les classes des réseaux simples non enracinés de niveau 1, des réseaux unicycliques, et des réseaux non enracinés de niveau 1, proviennent directement de leur définition.

Certaines sous-classes sont placées à la même hauteur car elles sont en relation. Par exemple les ensembles de bipartitions 2-compatibles peuvent être représentés par des réseaux médians de dimension 2 [Moulton et Huber, 2005]. Les ensembles de bipartitions faiblement compatibles sont considérés comme les équivalents, dans le contexte non enraciné, des hiérarchies faibles (voir le lemme 5 de Bandelt et Dress [1992a]), et la définition d'un ensemble circulaire de bipartitions est similaire à celle d'une prépyramide. Une autre relation qui n'apparaît pas de façon explicite dans la figure 1.27 est le fait que tout ensemble circulaire de bipartitions peut être représenté par un réseau de bipartitions planaire extérieur [Wetzel, 1995] (voir aussi le théorème 2 de Dress et Huson [2004]).

En complément de la figure 1.27(b), on pourra se référer à la figure 14 de Barthélemy *et al.* [2004] pour visualiser ces restrictions du point de vue des dissimilarités associées aux

ensembles restreints de clades que nous avons évoqués, et d'autres que nous n'avons pas mentionnés ici.

La figure 1.27(d) montre les relations entre classes de réseaux phylogénétiques explicites enracinés binaires. L'étude des diverses définitions possibles des réseaux de niveau 1 dans le cas non binaire - qui sont équivalentes pour des réseaux binaires - est détaillée par Rosselló et Valiente [2009].

L'inclusion de la classe des réseaux normaux dans celle des réseaux réguliers provient du théorème 3.4 de Willson [2010a]. De plus tout réseau normal est un réseau sans descendance hybride [van Iersel *et al.*, 2010b], et les autres inclusions découlent directement des définitions.

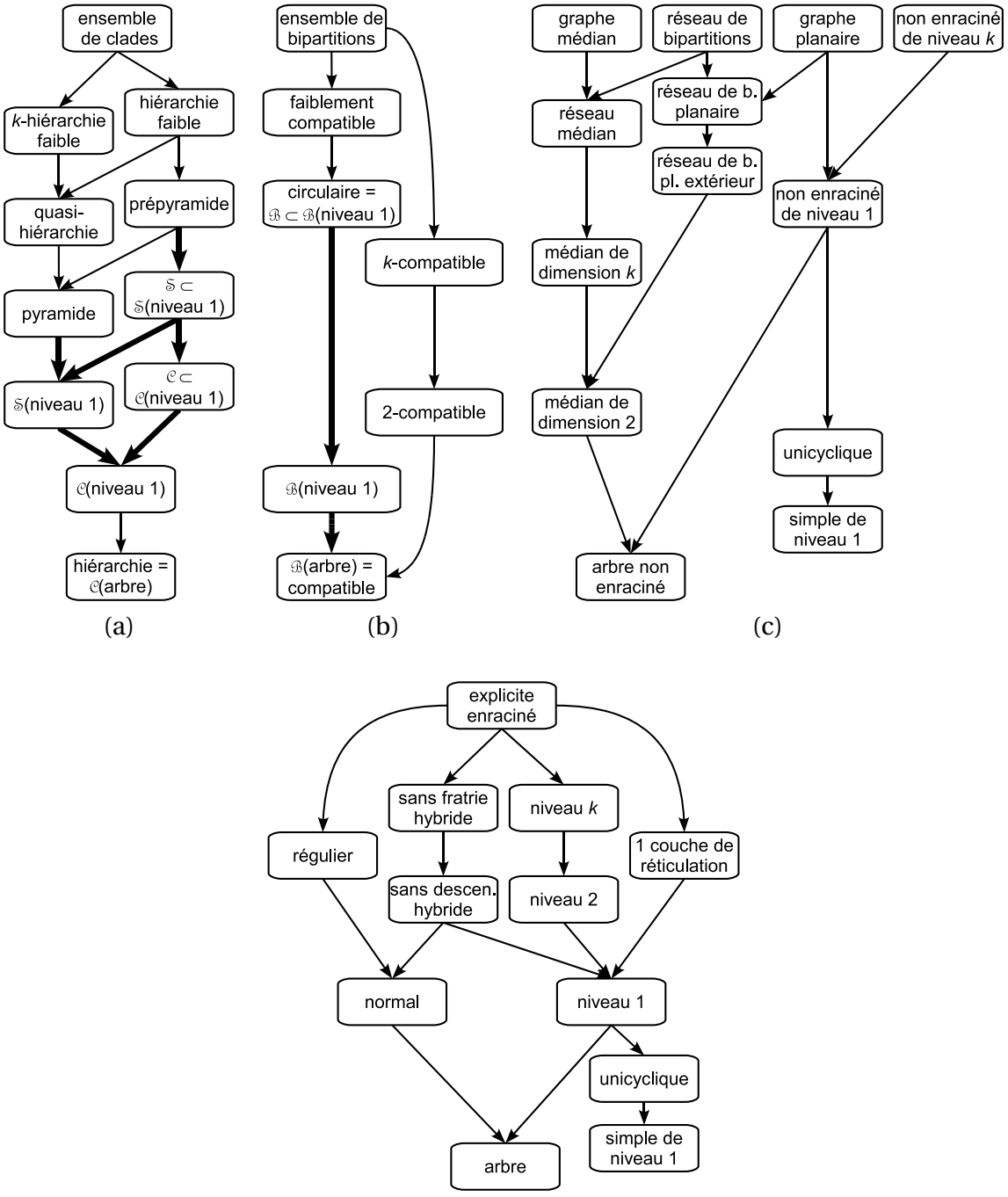


FIGURE 1.27 : Diagrammes récapitulatifs des inclusions de sous-classes de réseaux phylogénétiques : ensembles de clades (a), de bipartitions (b), réseaux non enracinés (c), réseaux phylogénétiques explicites enracinés binaires (d). Les flèches vont d'une classe vers une classe contenue (plus restreinte). Les flèches en gras indiquent des contributions de cette thèse.

2 Algorithmes combinatoires de reconstruction

Après la présentation au chapitre 1 des objets mathématiques au coeur de notre étude, nous nous attaquons maintenant à la reconstruction de réseaux phylogénétiques. Après un état de l'art des diverses méthodes combinatoires existantes, que nous détaillons tout particulièrement pour les méthodes de triplets, nous montrons des nouveaux résultats sur leur généralisation aux quadruplets. Puis nous proposons une nouvelle méthode heuristique de reconstruction de réseaux phylogénétiques explicites enracinés à partir de clades souples.

2.1 Méthodes et algorithmes existants

2.1.1 Panorama des diverses méthodes

a) Reconstruction directe à partir des arbres

Nous allons nous concentrer dans ce panorama sur les méthodes permettant de reconstruire des réseaux à partir d'arbres portant sur les mêmes ensembles de taxons, sans répétition de taxon. Nous détaillerons dans le chapitre 3 le contexte plus général auquel on est de plus en plus confronté en pratique et qui conduit, pour des raisons biologiques ou méthodologiques, à lever ces deux restrictions.

L'utilisation directe des arbres comme données fournies en entrée aux algorithmes de reconstruction de réseaux explicites se heurte à deux obstacles de complexité théorique. Tout d'abord, déterminer si un arbre phylogénétique enraciné est contenu dans un réseau phylogénétique explicite enraciné est un problème **NP-complet** [1] [Kanj *et al.*, 2008], et le reste même pour les réseaux réguliers sans fratrie hybride. Il devient toutefois traitable en temps polynomial pour les réseaux normaux, les réseaux binaires sans descendance hybride, et les réseaux de niveau k [van Iersel *et al.*, 2010b].

Quant au problème de reconstruction, à partir d'arbres enracinés, d'un réseau enraciné avec un nombre minimal de sommets hybrides, appelé **nombre d'hybridation**, il est

1. Un problème NP-complet est **NP-difficile** (c'est-à-dire qu'il n'admet probablement pas d'algorithme exact dont le temps de calcul est polynomial en la taille des données) et **dans NP** (c'est-à-dire qu'on peut vérifier en temps polynomial qu'une solution est correcte).

également NP-complet, et même **APX-difficile**[2], pour deux arbres [Bordewich et Semple, 2007b]. Une approche exacte de reconstruction devra donc se limiter à des données représentables par des réseaux ayant un faible nombre de sommets hybrides, (comme l'admettent les auteurs du logiciel HorizStory[3]).

Pour deux arbres, le problème est toutefois **FPT**[4] en le nombre de sommets hybrides [Bordewich et Semple, 2007a; Bordewich *et al.*, 2007], et un algorithme a été implémenté dans le logiciel HybridNumber [Bordewich *et al.*, 2007] puis de manière optimisée dans un programme plus rapide, HybridInterleaves [Collins *et al.*, 2011]. De plus une résolution par programmation linéaire en nombres entiers est aussi proposée [Wu et Wang, 2010]. Implémentée dans le logiciel SPRDist, elle semble en pratique plus rapide qu'HybridNumber.

Pour plus de deux arbres, d'autres stratégies ont été proposées pour pallier la complexité théorique du calcul du nombre d'hybridation. Il est possible de trouver une borne inférieure et une borne supérieure de ce nombre, bornes qui se révèlent en pratique assez fines quand le nombre d'arbres est petit [Wu, 2010]. La borne supérieure, appelée SIT (Stepwise Insertion of Trees) est obtenue en essayant d'insérer chaque arbre successivement dans le réseau reconstruit, en ajoutant le nombre minimal de sommets hybrides. La borne inférieure, appelée RH (Reticulation Hypercube) est trouvée par programmation linéaire en nombres entiers, elle est également valable dans le cas d'arbres non binaires. L'implémentation du calcul de ces deux bornes, et de la construction du réseau phylogénétique trouvé pour la borne supérieure, sont disponibles dans le logiciel PIRN.

De plus, si les arbres sont contenus dans un réseau de niveau 1 ayant k sommets hybrides, il est possible de résoudre le problème à deux arbres en $O(nk)$ [Nakhleh *et al.*, 2005b], l'algorithme est implémenté sous le nom SpNet. Un meilleur algorithme a été proposé, étendu à t arbres enracinés (non nécessairement binaires) dont un raffinement est contenu dans un réseau de niveau 1 : il fonctionne en $O(t^2n^2)$ [Huynh *et al.*, 2005]. De plus, si les t arbres ne proviennent pas d'un réseau de niveau 1, les mêmes auteurs proposent un algorithme en $O(2^{3td}n^{2t})$ (d étant le degré sortant maximal des arbres en entrée) qui trouve le plus petit ensemble de feuilles $X' \subseteq X$ tel qu'un raffinement de chaque arbre en entrée restreint à $X - X'$ est contenu dans un réseau de niveau 1.

Les auteurs de LatTrans [Hallett et Lagergren, 2001; Addario-Berry *et al.*, 2003] annoncent quant à eux une complexité en temps en $O(2^{4h}n^2)$ pour reconstruire un réseau à h sommets hybrides à partir d'un arbre des espèces et d'arbres de gènes enracinés sous certaines restrictions.

Face à la forte complexité de la reconstruction de réseaux parcimonieux à partir

2. c'est-à-dire qu'il est très improbable qu'on puisse trouver un algorithme en temps polynomial qui pour un paramètre $\epsilon > 0$ fournisse une solution à $k(1+\epsilon)$ sommets hybrides, où k est le nombre minimal de sommets hybrides.

3. "... we are therefore limited to comparing trees that are relatively similar." [MacLeod *et al.*, 2005]

4. c'est-à-dire qu'on peut le résoudre par un algorithme paramétré par un paramètre k, en temps $O(f(k).poly(n))$, où f est une fonction quelconque et n la taille de l'entrée du problème.

d'arbres, d'autres pistes que les algorithmes exacts ont été proposées. Tout d'abord les méthodes de construction de réseaux phylogénétiques abstraits, comme celles des **réseaux de consensus** [Holland et Moulton, 2003], qui fournissent des réseaux médians de dimension au plus k. En effet, en considérant l'ensemble des bipartitions qui apparaissent dans au moins 1 arbre sur $k+1$, parmi tous ceux fournis en entrée, on obtient un ensemble k-compatible de bipartitions, qui peut être représenté par un réseau de consensus.

Une autre possibilité est d'utiliser des heuristiques pour trouver, à partir des arbres fournis en entrée, un réseau avec un faible nombre de sommets hybrides. Ainsi, la méthode RIATA-HGT [Nakhleh *et al.*, 2005a], implémentée dans le logiciel PhyloNet [Than *et al.*, 2008] est basée sur une approche diviser-pour-régner. Des méthodes gloutonnes sont proposées dans le programme EEEP [Beiko et Hamilton, 2006], et le logiciel T-Rex [Boc *et al.*, 2010] : elles proposent divers critères d'optimisation pour choisir le prochain sommet hybride à insérer dans le réseau.

b) Reconstruction à partir des triplets, quadruplets, clades, bipartitions

Une autre piste pour obtenir des algorithmes efficaces est de reconstruire les réseaux phylogénétiques non pas directement à partir de leurs arbres, mais à partir de sous-ensembles de leurs feuilles : triplets ou clades dans le cas enraciné, quadruplets ou bipartitions dans le cas non enraciné. Résoudre ainsi un problème légèrement différent permet d'obtenir des algorithmes de meilleure complexité.

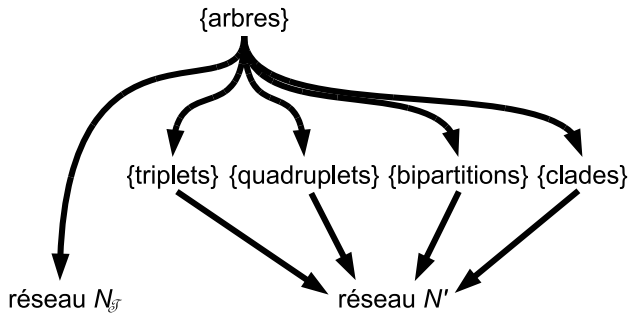
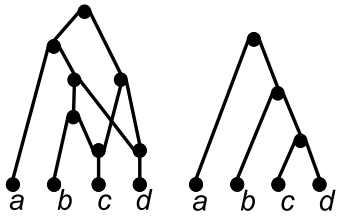


FIGURE 2.1 : Méthodes combinatoires de reconstruction de réseaux phylogénétiques : reconstruire un réseau N' à partir de triplets, de quadruplets, de clades ou de bipartitions extraits des arbres fournis en entrée est parfois plus rapide que reconstruire directement un réseau $N_{\mathcal{T}}$ à partir de ces arbres.

Cependant, cette démarche indirecte illustrée en figure 2.1 donne parfois des résultats non satisfaisants vis-à-vis des arbres fournis en entrée. Il se peut en effet que le réseau reconstruit N' contienne bien tous les triplets ou clades des arbres donnés en entrée, mais pas les arbres eux-mêmes. C'est le cas par exemple dans la figure 2.2(a), où le réseau re-

présenté contient tous les clades, stricts et souples, et tous les triplets de l'arbre T de la figure 2.2(b), mais ne contient pas T lui-même.



(a) (b)

FIGURE 2.2 : Un réseau N de niveau 2 (a) qui contient tous les clades, stricts et souples, et tous les triplets de l'arbre T (b) mais ne contient pas T.

Le réseau N′ obtenu indirectement a toutefois plusieurs intérêts. Tout d'abord, il propose une sorte de borne inférieure de la complexité du réseau reconstruit directement à partir des arbres. Précisons cette affirmation, en supposant que l'on s'intéresse aux réseaux et arbres binaires, et qu'on considère une méthode de reconstruction à partir d'un ensemble $\mathcal{T}$ d'arbres (dont l'union des ensembles de clades est $\mathcal{C}$ et l'union des ensembles de triplets est $\mathcal{R}$), où l'on tente d'obtenir un réseau optimal, c'est-à-dire qui minimise un certain score de complexité $s(N)$ (par exemple le nombre de sommets hybrides, ou le niveau). Appelons $\mathcal{N}_\mathcal{T}$ l'ensemble des réseaux qui contiennent les arbres en entrée, $\mathcal{N}_\mathcal{C}$ et $\mathcal{N}_\mathcal{R}$ l'ensemble des réseaux qui contiennent respectivement tous les clades et tous les triplets de ces arbres. Tout réseau appartenant à $\mathcal{N}_\mathcal{T}$ contient aussi tous les clades de $\mathcal{T}$, donc appartient aussi à $\mathcal{N}_\mathcal{C}$. De plus, tout réseau appartenant à $\mathcal{N}_\mathcal{C}$ contient aussi tous les triplets induits par les clades de $\mathcal{C}$ d'après la proposition 2, donc tous les triplets des arbres de $\mathcal{T}$, donc appartient aussi à $\mathcal{N}_\mathcal{R}$. Ainsi, $\mathcal{N}_\mathcal{T} \subseteq \mathcal{N}_\mathcal{C} \subseteq \mathcal{N}_\mathcal{R}$, donc l'élément de $\mathcal{N}_\mathcal{T}$ de score s minimal, $N_\mathcal{T}$, a un score s inférieur à l'élément N′ de $\mathcal{N}_\mathcal{C}$ (ou de $\mathcal{N}_\mathcal{R}$) de score minimal. Ainsi, le réseau $N_\mathcal{T}$ reconstruit directement à partir des arbres est au moins aussi complexe que celui reconstruit à partir des clades ou des triplets de ces arbres.

D'autre part, les arbres phylogénétiques fournis en entrée sont parfois peu fiables. Dans ce cas, on peut préférer travailler uniquement avec ceux de leurs clades dont la qualité est assurée par exemple par une **valeur de bootstrap** [5] suffisante. De plus, dans certains contextes, comme celui des études populationnelles, il a été montré récemment [Degnan et Rosenberg, 2006] que selon le modèle **coalescent** [6], un arbre reconstuit sur plus de trois

5. La valeur de bootstrap d'un clade indique le niveau de confiance qu'on peut lui apporter : c'est la proportion des arbres reconstruits qui contiennent le clade, si l'on répète la reconstruction un certain nombre de fois en modifiant les données utilisées en entrée, en sélectionnant des portions distinctes des séquences de gènes par exemple [Felsenstein, 2004].

6. Le modèle coalescent [Kingman, 2000], introduit en génétique des populations, consiste à considérer une taille de population constante, et plusieurs générations - apparaissant à intervalles de temps constants

taxons a une forte probabilité d'être différent de l'arbre des espèces, ce qui renforce l'intérêt pour les méthodes fonctionnant à partir de données de triplets. Des premiers résultats indiquent que ces méthodes peuvent effectivement s'avérer plus fiables en pratique que celles qui prennent en compte directement l'ensemble des données en entrée [DeGiorgio et Degnan, 2010].

Historiquement, alors qu'un algorithme de reconstruction d'un arbre à partir de ses triplets avait été introduit dans le contexte des bases de données dès les années 80 [Aho *et al.*, 1981], puis accéléré dans le contexte de la reconstruction phylogénétique [Henzinger *et al.*, 1999; Jansson *et al.*, 2005], les méthodes de reconstruction de réseaux phylogénétiques à base de triplets n'ont été introduites qu'à partir de 2006, suite à l'article fondateur de Jansson et Sung [2006]. Il s'agit principalement de méthodes exactes de complexité polynomiale, que nous présenterons de manière plus détaillée en section 2.1.2.

Contrairement à la reconstruction d'arbres phylogénétiques à partir de triplets, la reconstruction à partir de quadruplets est un problème NP-difficile en toute généralité, quand on ne connaît pas tous les quadruplets de l'arbre à reconstruire [Steel, 1992] et un grand nombre d'approches existent pour résoudre ce problème (voir par exemple [Chor, 1998]). En revanche, pour les triplets comme pour les quadruplets, si l'on sait que l'ensemble fourni en entrée est l'ensemble de tous les triplets (respectivement tous les quadruplets) d'un arbre, alors il est possible de reconstruire cet arbre avec seulement $O(n \log n)$ requêtes sur la présence d'un triplet (respectivement d'un quadruplet) dans cet ensemble [Pearl et Tarsi, 1986; Kannan *et al.*, 1996].

En ce qui concerne la reconstruction de réseaux phylogénétiques à partir de quadruplets, des algorithmes proposent de reconstruire des réseaux phylogénétiques abstraits en passant par l'étape intermédiaire d'inférence d'un ensemble de bipartitions qui induisent ces quadruplets.

Il existe ainsi une caractérisation des ensembles de quadruplets induits par un ensemble faiblement compatible de bipartitions [Bandelt et Dress, 1994], que l'on peut adapter pour obtenir un algorithme de reconstruction de bipartitions pondérées à partir de quadruplets pondérés (en choisissant pour chaque bipartition le poids minimum parmi ceux des quadruplets qu'elle induit dans l'ensemble fourni en entrée) [Berry et Bryant, 1999]. Une heuristique de reconstruction de bipartitions pondérées circulaires à partir d'ensembles de quadruplets pondérés est à la base du logiciel QNet [Grünewald *et al.*, 2007]. L'intérêt de cet algorithme est de fournir un réseau abstrait planaire extérieur, d'après la remarque faite en section 1.5.3, et donc facile à visualiser avec les méthodes de dessin optimisé de réseaux de bipartitions proposées dans le logiciel SplitsTree [Gambette et Huson, 2008].

- dont on reconstruit les relations de parenté en choisissant aléatoirement dans la génération précédente le ou les parents des individus de manière récursive, en commençant par la génération la plus récente. Si l'on autorise un individu à avoir deux parents, on parle de **modèle coalescent avec recombinaison** [Hudson, 1991].

Enfin, en ce qui concerne la reconstruction de réseaux phylogénétiques enracinés à partir de clades, la seule méthode existante permettant de reconstruire des réseaux abstraits à partir de clades stricts [Huson et Rupp, 2008] a été évoquée au début de la section 1.4.1.

2.1.2 Reconstruction à partir de triplets

La première question algorithmique que l'on considère naturellement sur les triplets et les réseaux phylogénétiques explicites est de déterminer l'ensemble des triplets contenus dans un réseau, ce qui est fait en temps optimal $O(n^3)$ pour un réseau à n feuilles, par un algorithme de programmation dynamique [Byrka *et al.*, 2010].

Citons également les algorithmes de reconstruction d'arbres à partir de triplets. Nous avons déjà évoqué l'algorithme de Aho et al. et ses diverses implémentations [Aho *et al.*, 1981; Henzinger *et al.*, 1999; Jansson *et al.*, 2005]. Si l'ensemble de triplets en entrée est dense, un autre résultat intéressant est un algorithme certifiant de reconstruction en temps optimal $O(n^3)$: s'il existe un arbre qui contient les triplets alors on le reconstruit, sinon on exhibe un conflit qui implique un ensemble de triplets portant sur quatre feuilles [Guillemot et Berry, 2010].

La première garantie donnée sur la reconstruction de réseaux phylogénétiques explicites binaires enracinés à partir de triplets est qu'il est possible de reconstruire de tels réseaux pour tout ensemble de triplets [Jansson et Sung, 2006] car il existe un réseau phylogénétique explicite enraciné binaire sur n feuilles, construit à partir d'un réseau de tri de n éléments [Batcher, 1968], qui contient tous les triplets possibles sur ces n feuilles. Toutefois ce réseau a peu d'intérêt biologique puisqu'il contient justement tous les triplets possibles, et a un nombre très important de sommets hybrides.

Le premier algorithme utile d'un point de vue phylogénétique est celui de reconstruction de réseaux de niveau 1 à partir d'un ensemble dense $\mathcal{R}$ de triplets. Il s'exécute en temps $O(n^6)$ [Jansson et Sung, 2006] et se base sur le concept des SN-ensembles : cet algorithme diviser-pour-régner consiste à identifier des sous-ensembles de feuilles $L \subseteq X$, appelés **SN-ensembles**, tels que pour tous $a, b \in L$ et $c \notin L$, $\mathcal{R}_{|\{a,b,c\}} = ab|c$ (le seul triplet sur les feuilles a, b, c est $ab|c$) [7]. Ces SN-ensembles ont la particularité de former une famille laminaire (et donc d'être représentables par un **SN-arbre** enraciné) quand l'ensemble de triplets en entrée est dense. Le calcul du SN-arbre, intialement réalisé par un algorithme en temps $O(n^5)$ [Jansson et Sung, 2006], peut en fait s'effectuer en temps $O(n^3)$ [Jansson *et al.*, 2006].

De plus, dans tout réseau N qui contient les triplets de $\mathcal{R}$, tout SN-ensemble de $\mathcal{R}$ est l'ensemble des feuilles descendantes des cibles d'une union d'isthmes dont les sources appartiennent à un même blob B de N [van Iersel *et al.*, 2009a]. Par exemple, la figure 2.3 montre un réseau de niveau 1 qui contient un ensemble dense de triplets

7. Cette définition équivalente [To et Habib, 2009] n'est pas celle initalement proposée par Jansson et Sung [2006] mais s'avère plus simple à manipuler.

$\mathcal{R} = \{c|ab,d|ab,e|ab,a|cd,e|ac,a|de,b|cd,b|ce,b|de,e|cd\}$, dont le SN-ensemble $\{c,d\}$ est l'ensemble des feuilles sous l'isthme w, $\{a,b\}$ est l'ensemble des feuilles sous u et v, et $\{a,b,c,d,e\}$ est l'ensemble des feuilles sous tous les isthmes de B : u, v, w et x. Ainsi, on peut associer à tout SN-ensemble de $\mathcal{R}$ un blob B du réseau solution. Cette propriété explique le nom donné à ces ensembles : "SN" signifie ici "simple network" car pour tout blob du réseau, si l'on contracte, pour chacun de ses isthmes incidents, tous les sommets descendants de la cible de l'isthme, on obtient un réseau simple.

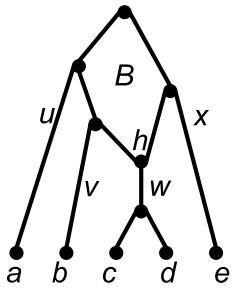


FIGURE 2.3 : Un réseau de niveau 1 qui contient l'ensemble de triplets $\{c|ab,d|ab,e|ab,a|cd,e|ac,a|de,b|cd,b|ce,b|de,e|cd\}$, dont $\{a,b\}$, $\{c,d\}$ et $\{a,b,c,d,e\}$ sont des SN-ensembles.

Mentionnons toutefois que l'association d'un blob à un SN-ensemble n'est pas bijective. Plusieurs SN-ensembles concernent en effet le même blob. Ainsi, pour réaliser la reconstruction du réseau solution, la première étape est de déterminer tous les SN-ensembles associés aux isthmes incidents au blob qui contient la racine du réseau. Si l'on sait faire cette étape, alors on pourra calculer l'ensemble du réseau de manière récursive en reconstruisant progressivement les blobs du réseau "de haut en bas" (pour la relation de descendance).

Les premiers algorithmes de reconstruction de réseaux de niveau 1 [Jansson et Sung, 2006] et 2 [van Iersel *et al.*, 2009a] utilisent le fait qu'il existe toujours un réseau solution où chaque SN-ensemble maximal correspond à l'ensemble des feuilles sous un isthme du blob le plus haut (sauf pour un cas bien identifié de réseaux de niveau 2, où exactement un SN-ensemble correspond à l'union des feuilles sous deux isthmes : il faut donc examiner cette possibilité pour chacun des SN-ensembles maximaux, mais l'algorithme reste polynomial [van Iersel *et al.*, 2009a]). Pour les niveaux k supérieurs, pour k fixé, [To et Habib, 2009] montrent une propriété particulière des SN-ensembles qui correspondent à l'ensemble des feuilles sous exactement un isthme incident au blob le plus haut du réseau solution. Cette propriété implique qu'on peut les trouver en examinant un nombre polynomial (plus précisément, exponentiel en k, mais k est fixé et ne fait pas partie de l'entrée du problème) de possibilités.

Ainsi, nous avons vu que l'identification des SN-ensembles maximaux à partir d'un ensemble dense de triplets en entrée permet d'utiliser une approche diviser-pour-régner

pour se concentrer sur chaque blob, l'un après l'autre. Cette étape de reconstruction de chaque blob s'effectue aussi en temps polynomial (toujours exponentiel en k), grâce au principe suivant : les triplets concernant toutes les feuilles descendantes du blob de niveau k, sauf celles descendantes d'un sommet hybride, sont contenus dans un réseau de niveau $k-1$. Par exemple, dans la figure 2.3, les triplets portant sur l'ensemble des feuilles excepté celles descendant de la cible du sommet hybride h, c'est-à-dire sur l'ensemble $\{a, b, e\}$, sont contenus dans un arbre. Ainsi, le principe de l'algorithme, en temps $O(n^k)$, est de considérer à tour de rôle chacun des $O(n)$ SN-ensembles en supposant qu'il est constitué de l'ensemble L des feuilles sous un sommet hybride, de le supprimer, et de vérifier s'il est effectivement possible de reconstruire un réseau de niveau $k-1$ contenant l'ensemble des triplets restants $\mathcal{R}_{|X-L}$.

Ces deux étapes, l'identification des blobs, et la reconstruction de chaque blob, conduisent à des algorithmes respectivement en temps $O(n^3)$, $O(n^8)$ et $O(n^{\lfloor 9k/2 \rfloor + 4})$ pour reconstruire un réseau respectivement de niveau 1 [Jansson *et al.*, 2006], 2 [van Iersel *et al.*, 2009a] et k [To et Habib, 2009] à partir d'un ensemble dense de triplets. Les algorithmes pour le niveau 1 et le niveau 2 sont disponibles dans le logiciel Level2. Des algorithmes proposent également la reconstruction d'un réseau respectivement de niveau 1, 2, et k, contenant un nombre minimal de sommets hybrides, en temps $O(n^5)$ (implémenté dans les logiciels Marlon et Simplistic [van Iersel et Kelk, 2010]), $O(n^9)$, et $O(n^{6k+2})$ respectivement.

On le voit, ces algorithmes demandent un temps de calcul important en fonction du niveau des réseaux à reconstruire. Or, une autre approche exacte permet de s'accommoder du fait que le réseau à reconstruire a un niveau plus élevé que ceux que nos capacités de calcul nous permettent de reconstruire. Il s'agit de supprimer des feuilles pour baisser la complexité du réseau à reconstruire en obtenant un niveau inférieur. Ainsi, la solution sera moins complète qu'espéré, mais fournira tout de même une information sur l'évolution de certaines des espèces étudiées. Par exemple, pour le calcul d'un arbre contenant un ensemble dense $\mathcal{R}$ de triplets après élimination d'un nombre minimum de l feuilles, des algorithmes de complexité paramétrée résolvent ce problème en temps $O(n^4 + 3.12^l)$ ou $O(4^l n^3)$ [Guillemot et Berry, 2010], bien que la version sans restriction de densité soit NP-difficile, et même **W[2]-difficile** [Berry et Nicolas, 2006] [8].

D'autres résultats négatifs limitent les approches de reconstruction à partir de triplets. Tout d'abord, le problème d'existence d'un réseau de niveau k qui contient un ensemble de triplets en toute généralité (sans la restriction de densité) est NP-complet [Jansson *et al.*, 2006; van Iersel *et al.*, 2009b], même pour les réseaux simples de niveau k, pour tout $k \geq 1$ fixé. Si le niveau k fait partie de l'entrée du problème, l'existence d'un réseau de niveau k qui contient un ensemble dense de triplets fourni en entrée est NP-complet [van Iersel et Kelk, 2011].

8. c'est-à-dire qu'il est improbable qu'un algorithme en temps $O(f(l) \cdot \mathrm{poly}(n))$ puisse résoudre ce problème.

Face à cette complexité, un algorithme d'approximation en temps $O(n|\mathcal{R}|^3)$ permet de fournir un réseau de niveau 1 compatible avec au moins 5/12 (41,66%) des triplets présents dans l'ensemble $\mathcal{R}$ fourni en entrée [Jansson *et al.*, 2006]. Ce résultat est amélioré par Byrka *et al.* [2010] pour obtenir un algorithme qui fournit un réseau compatible avec un ratio d'au moins 48% des triplets en entrée, en temps $O(n^3 + n|\mathcal{R}|)$.

Nous évoquerons au début de la section 3.1 d'autres résultats sur la reconstruction de réseaux contenant un nombre maximum de triplets en entrée, pour prendre en compte la présence de bruit dans les données.

2.2 Reconstruction à partir de quadruplets

Dans cette section, on s'intéresse à la reconstruction de réseaux phylogénétiques non enracinés à partir de quadruplets. De même que pour les algorithmes existants sur les triplets, suite à la remarque de la section 1.3.3, on se concentrera sur la reconstruction de réseaux binaires. Ainsi, tous les réseaux évoqués dans cette section ont degré au plus 3.

L'intérêt de la reconstruction de réseaux à partir de quadruplets, par rapport à celle à partir de triplets, vient du fait que l'enracinement, en phylogénie, est souvent source d'erreurs [Bininda-Emonds *et al.*, 2005]. Ainsi, il sera préférable de ne l'effectuer qu'à la fin de l'algorithme de reconstruction, et donc de commencer par reconstruire un réseau phylogénétique non enraciné.

2.2.1 Extraction des quadruplets d'un réseau

La première question algorithmique à propos des quadruplets et des réseaux phylogénétiques explicites non enracinés est de déterminer comment vérifier qu'un quadruplet est contenu dans un réseau, et par extension déterminer l'ensemble des quadruplets d'un réseau donné.

L'approche de programmation dynamique évoquée dans la section précédente pour calculer l'ensemble des triplets contenus dans un réseau phylogénétique enraciné en $O(n^3)$ [Byrka *et al.*, 2010] ne s'étend pas au cas non enraciné. Toutefois le problème garde une complexité en temps polynomial.

Théorème 4 *L'ensemble des quadruplets d'un réseau phylogénétique non enraciné* N *de niveau* k*, pour tout entier* k*, peut être calculé en temps* $O(n^5(1+\alpha(n,n)))$*, où* α *est l'inverse de la fonction d'Ackermann.*

Démonstration. En utilisant la définition 1.9 des quadruplets d'un réseau non enraciné, on applique simplement le meilleur algorithme connu pour le problème 2-VERTEX-DISJOINT PATHS PROBLEM, qui détermine s'il existe deux chemins à sommets disjoints l'un entre a et b, et l'autre entre c et d, en temps $O(n+n\alpha(n,n))$ [Tholey, 2009], pour chacun des $O(n^4)$

quadruplets $ab|cd$. Ainsi, la complexité totale de l'algorithme qui renvoie l'ensemble des quadruplets de N est $O(n^5(1+\alpha(n,n)))$. □

2.2.2 Difficulté de la reconstruction dans le cas général

Nous nous focalisons maintenant sur la reconstruction d'un réseau non enraciné de niveau k qui contient un ensemble donné de quadruplets.

Problème 1 (LEVEL-k QUARTET CONSISTENCY)
Entrée : *un ensemble Q de quadruplets.*
Sortie : *OUI s'il existe un réseau non enraciné de niveau k qui contient tous les quadruplets de Q, NON sinon.*

Steel a prouvé en 1992 que le problème LEVEL-0 QUARTET CONSISTENCY est NP-complet [Steel, 1992], nous généralisons ci-dessous sa preuve au niveau 1.

Tout d'abord, nous prouvons que pour le niveau 1, ce problème est équivalent si l'on ajoute la restriction que le réseau à reconstruire doit être simple. Dans ce cas nous parlerons du problème SIMPLE LEVEL-1 QUARTET CONSISTENCY.

Lemme 3 *Soit Q un ensemble de quadruplets. S'il existe un réseau non enraciné N de niveau 1 qui contient Q, alors il existe un réseau simple non enraciné N' de niveau 1 qui contient Q.*

Démonstration. Considérons un réseau non enraciné N de niveau 1 qui contient Q, et un réseau simple non enraciné $N' = Simple(N)$ de niveau 1 (voir la définition 1.23, page 56 et la figure 1.8, page 29). Pour tout quadruplet $ad|bc$ de Q, comme Q est contenu dans N alors il existe un arbre non enraciné contenu dans N et qui contient $ad|bc$. En particulier, cet arbre contient la bipartition $A|\bar{A}$ telle que $a,d \in A$ et $b,c \in \bar{A}$. Ainsi, la bipartition $A|\bar{A}$ est contenue dans N, et donc dans N' d'après le lemme 2, donc $ad|bc$ est contenu dans N'. □

Théorème 5 LEVEL-1 QUARTET CONSISTENCY *est un problème NP-complet.*

Démonstration. Comme il est possible de vérifier en temps polynomial qu'un ensemble de quadruplets est bien contenu dans un réseau non enraciné de niveau 1 d'après le théorème 4, le problème LEVEL-1 QUARTET CONSISTENCY est dans NP.

Pour prouver qu'il est NP-difficile, on réduit grâce au lemme 3 le problème SIMPLE LEVEL-1 QUARTET CONSISTENCY, qui est NP-difficile, comme nous allons le prouver par réduction du problème BETWEENNESS, qui est NP-difficile [Opatrny, 1979]. Rappelons que le problème BETWEENNESS consiste à décider, pour une collection C de séquences de trois éléments d'un ensemble A, s'il existe un ordre de A qui respecte les contraintes imposées

par C, où $(a,b,c) \in C$ signifie que $a < b < c$ ou bien $c < b < a$. On peut supposer que tout élément de A apparaît dans au moins une contrainte de C.

Étant donné une instance (A,C) du problème BETWEENNESS, on construit une instance $(X,\mathcal{Q}(C))$ du problème SIMPLE LEVEL-1 QUARTET CONSISTENCY, où X est l'ensemble des feuilles et $\mathcal{Q}(C)$ l'ensemble des quadruplets, de la manière suivante :

- $X = A \cup \{\alpha,\beta,\alpha_1,\beta_1\} \cup \bigcup_{i\in[1..|C|]} X_i$, où $X_i = \{p_i,q_i\}$,
- à chaque contrainte $c_i = (a,b,c) \in C$ d'une instance de BETWEENNESS, pour $a,b,c \in A$, on associe un ensemble de quadruplets $\mathcal{Q}_i = \{\beta p_i|q_i a,\ \alpha q_i|p_i a,\ \beta p_i|q_i b,\ \alpha q_i|p_i b,\ \beta p_i|q_i c,\ \alpha q_i|p_i c,\ p_i b|c q_i,\ p_i a|bc\}$.
- $\mathcal{Q}(C) = \mathcal{Q}'(C) \cup \mathcal{Q}''(C)$, où $\mathcal{Q}'(C) = \bigcup_{i\in[1..|C|]} \mathcal{Q}_i$, et $\mathcal{Q}''(C) = \{\alpha\alpha_1|\beta\beta_1,\ \alpha\beta|\alpha_1\beta_1\} \cup \{\beta\beta_1|x\alpha,\ \alpha\alpha_1|x\beta,\ \alpha_1\beta_1|x\beta,\ \forall x \in A \cup X_i\}$.

$\Rightarrow$: si un ordre σ de A respecte les contraintes de C, alors pour chaque contrainte $c_i = (a,b,c) \in C$:

- si $a <_\sigma b <_\sigma c$, définissons $r_i = p_i$ et $s_i = q_i$,
- si $c <_\sigma b <_\sigma a$, définissons $r_i = q_i$ et $s_i = p_i$.

Considérons le réseau simple non enraciné N de niveau 1 constitué d'un cycle à $4+|A|+2|C|$ sommets, chacun adjacent d'une feuille, tel que les feuilles sont apparaissent dans l'ordre suivant : $\alpha_1\alpha r_1 \ldots r_{|C|}$ puis σ puis $s_1 \ldots s_{|C|}\beta\beta_1$. Ce réseau montré en figure 2.4 contient $\mathcal{Q}_i$ pour tout i de $[1..|C|]$, et il contient tous les quadruplets de $\mathcal{Q}''(C)$, donc il contient $\mathcal{Q}(C)$. Ainsi N est une solution du problème SIMPLE LEVEL-1 QUARTET CONSISTENCY.

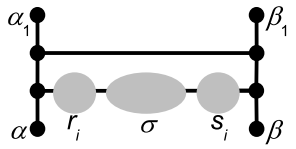


FIGURE 2.4 : Réseau simple de niveau 1 non enraciné construit à partir d'une solution σ de BETWEENNESS.

$\Leftarrow$: inversement, supposons qu'il existe un réseau simple N non enraciné de niveau 1 qui contient $\mathcal{Q}(C)$, alors N est composé d'un cycle de $4+|A|+2|C|$ sommets, chacun adjacent à une feuille. Pour toute feuille x, appelons x' le sommet voisin de x. Comme $\mathcal{Q}(C)$ contient $\{\alpha\alpha_1|\beta\beta_1,\ \alpha\beta|\alpha_1\beta_1\}$, ceci force α', α'_1, β'_1 et β' à apparaître dans cet ordre dans le cycle de N. Nous allons maintenant déterminer les positions des autres sommets dans cet ordre circulaire. Appelons respectivement I_1, I_2, I_3 et I_4 les intervalles de cet ordre circulaire entre α'_1 et β'_1, entre β'_1 et β', entre β' et α', et entre α' et α'_1, comme montré en figure 2.5(i).

Pour toute feuille x de $A \cup \bigcup_{i\in[1..|C|]} X_i$, comme $\beta\beta_1|x\alpha \in \mathcal{Q}(C)$, x' n'appartient pas à l'intervalle I_2. Il n'appartient pas non plus à I_4 à cause de $\alpha\alpha_1|x\beta$, ni à I_1 à cause de $\alpha_1\beta_1|x\beta$.

Le sommet x' est donc dans l'intervalle I_3. Déterminons à présent les ordres possibles de tous les sommets situés dans l'intervalle I_3.

Fixons i et considérons les sommets $\{a', b', c', p_i', q_i'\}$ où $c_i = (a, b, c)$. Fixons les positions de p_i' et q_i' entre α' et β'. Il y a deux possibilités comme montré sur les figures 2.5(ii) et (iii). Dans le premier cas, on appelle respectivement J_1, J_2 et J_3 les intervalles $]\alpha', p_i'[$, $]p_i', q_i'[$, $]q_i', \beta'[$. Dans le deuxième cas, on appelle respectivement J_3, J_2 et J_1 les intervalles $]\alpha', q_i'[$, $]q_i', p_i'[$, $]p_i', \beta'[$. Dans les deux cas, comme $\alpha q_i|p_i a \in \mathcal{Q}(C)$, a' n'est pas dans l'intervalle J_3. Comme $\beta p_i|q_i a \in \mathcal{Q}(C)$, a' n'est pas dans l'intervalle J_1. Toutefois ces deux quadruplets autorisent a' à être placé dans J_2. De même, b' et c' sont aussi dans l'intervalle J_2.

Enfin, étudions les positions relatives de a', b' et c'. Le quadruplet $p_i b|c q_i$ et le fait que b' et c' sont dans l'intervalle J_2 entre p_i' et q_i' forcent p_i', b', c' et q_i' à être placés dans cet ordre le long du cycle de N. De plus, comme $p_i a|bc \in \mathcal{Q}(C)$, p_i', a', b', c' et q_i' sont placés dans cet ordre, comme montré dans les figures 2.5(iv) et (v). Dans tous les cas, b' apparaît toujours placé entre a' et c'. Ainsi, l'ordre dans lequel les feuilles de A sont attachées au cycle de N respecte les contraintes de C. □

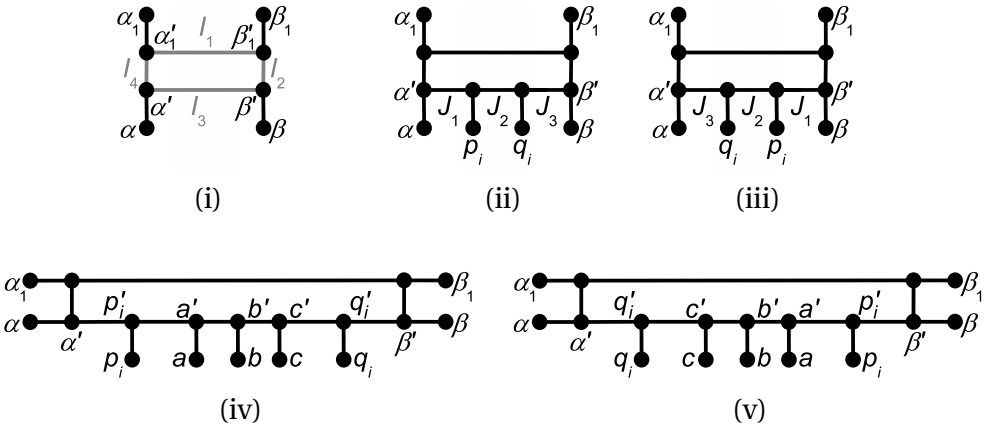


FIGURE 2.5 : Gadgets pour la réduction de BETWEENNESS à SIMPLE LEVEL-1 QUARTET CONSISTENCY : les structures imposées par $\{\alpha\alpha_1|\beta\beta_1, \alpha\beta|\alpha_1\beta_1\}$ (i), en ajoutant $\{\beta\beta_1|x\alpha, \alpha\alpha_1|x\beta, \alpha_1\beta_1|x\beta\}$ pour $x \in X_i$ (ii,iii), et en ajoutant les quadruplets restants de Q_i (iv,v).

Pour reconstruire un réseau non enraciné de niveau 1 à partir de quadruplets, nous pouvons proposer une première approche heuristique. En effet, l'algorithme QNet [Grünewald *et al.*, 2007] est une heuristique de reconstruction d'un ensemble circulaire de bipartitions à partir d'un ensemble de quadruplets pondérés. D'après le théorème 3, il serait donc possible de représenter son résultat par un réseau de niveau 1 qui contient ces bipartitions, ce qui fournirait une première méthode heuristique de reconstruction de réseaux explicites (non enracinés de niveau 1) à partir d'un ensemble de quadruplets.

2.2.3 Structure arborée depuis un ensemble dense de quadruplets

Nous nous concentrons maintenant sur le cas où l'ensemble de quadruplets $\mathcal{Q}$ en entrée est dense. Dans cette section, nous montrons comment trouver les blobs d'un réseau phylogénétique non enraciné N à partir de $\mathcal{Q}$. Pour ce faire, nous introduisons tout d'abord le concept de SN-bipartition, qui est analogue à celui de SN-ensemble [Jansson et Sung, 2006; To et Habib, 2009] évoqué en section 2.1.2.

Précisons que d'après le lemme 3, si nous savons qu'un réseau non enraciné de niveau 1 contient l'ensemble $\mathcal{Q}$ de quadruplets fourni en entrée, alors il existe également un réseau simple (c'est-à-dire à un seul blob non trivial) N', plus parcimonieux en termes de nombre d'arêtes, qui contient $\mathcal{Q}$. Toutefois, N' est moins intéressant en pratique car il n'est pas optimal quant au nombre de quadruplets contenus dans N' mais absents de $\mathcal{Q}$. Ceci explique pourquoi en pratique on commencera par essayer de déduire de $\mathcal{Q}$ les blobs du réseau à reconstruire, puis on se concentrera sur la reconstruction de chaque blob du réseau pour déterminer sa structure interne.

a) Construction des SN-bipartitions

Définition 2.1 *Soit $\mathcal{Q}$ un ensemble de quadruplets sur un ensemble X de taxons, $A \subseteq X$. Une bipartition $A|\bar{A}$ des taxons est une **SN-bipartition** de $\mathcal{Q}$ si c'est soit une bipartition triviale soit une bipartition qui vérifie la propriété suivante : pour tout $x, y \in A$, $z, t \in \bar{A}$, le seul quadruplet sur $\{x, y, z, t\}$, si $\mathcal{Q}$ en contient un, est $xy|zt$.*

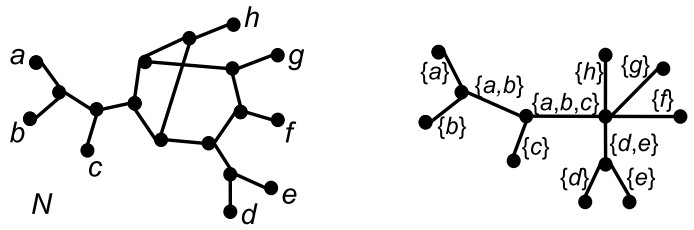
Cette définition d'une SN-bipartition est similaire à la définition des SN-ensembles proposée par To et Habib [2009]. La définition originale des SN-ensembles [Jansson et Sung, 2006] peut aussi être adaptée pour définir les SN-bipartitions dans un contexte non enraciné, comme clôtures par une opération de complétion d'ensembles. Toutefois, nous ne la décrirons pas ici car elle est plus complexe que celle que nous utilisons.

Nous donnons maintenant une propriété importante des SN-bipartitions avant de montrer comment les calculer de manière efficace.

Proposition 8 *Pour un ensemble dense $\mathcal{Q}$ de quadruplets, l'ensemble des SN-bipartitions de $\mathcal{Q}$ est un ensemble compatible de bipartitions.*

Démonstration. Considérons deux bipartitions $B_1 = A_1|A_1'$ et $B_2 = A_2|A_2'$. Supposons par l'absurde qu'aucune des quatre intersections $A_1 \cap A_2$, $A_1 \cap A_2'$, $A_1' \cap A_2$ et $A_1' \cap A_2'$ ne soit vide. Alors $\exists a \in A_1 \cap A_2$, $b \in A_1 \cap A_2'$, $c \in A_1' \cap A_2$ et $d \in A_1' \cap A_2'$. Comme $\mathcal{Q}$ est dense, alors il doit contenir un quadruplet sur $\{a, b, c, d\}$. Comme B_1 est une SN-bipartition, que $a, b \in A_1$ et que $c, d \in \bar{A}_1$, alors ce quadruplet devrait être $ab|cd$. Mais comme B_2 est aussi une SN-bipartition, que $a, c \in A_2$ et que $b, d \in \bar{A}_2$, ce quadruplet devrait être $ac|bd$, ce qui contredit la définition d'une SN-bipartition. □

Nous rappelons la propriété classique mentionnée en section 1.2.2 selon laquelle un ensemble compatible de bipartitions peut être représenté par un arbre non enraciné. On associe donc à l'ensemble des SN-bipartitions un arbre appelé **SN-arbre non enraciné** d'un ensemble dense de quadruplets, dont les arêtes sont étiquetées de façon bijective par les SN-bipartitions, comme illustré en figure 2.6(ii).



(i) (ii)

FIGURE 2.6 : Un réseau N de niveau 2 (i) et le SN-arbre non enraciné de son ensemble $\mathcal{Q}(N)$ de quadruplets (ii), où nous étiquetons par l'ensemble de feuilles A l'arête correspondant à la SN-bipartition $A|\bar{A}$.

Notons que si $\mathcal{Q}$ contient au plus un quadruplet sur chaque ensemble de quatre feuilles, alors le SN-arbre non enraciné de $\mathcal{Q}$ est exactement l'**arbre** T^* défini par Berry et Gascuel [2000], et les SN-bipartitions sont en bijection avec les arêtes de T^*. Nous utiliserons cette remarque pour donner un algorithme efficace de calcul du SN-arbre non enraciné de $\mathcal{Q}$.

Proposition 9 *Pour un ensemble dense $\mathcal{Q}$ de quadruplets, il y a $O(n)$ SN-bipartitions, et le SN-arbre non enraciné de $\mathcal{Q}$ peut être reconstruit en temps $O(n^4)$.*

Démonstration. La proposition 8 implique que le nombre de SN-bipartitions est linéaire en $n = |X|$.

L'algorithme de construction du SN-arbre non enraciné fonctionne de la manière suivante. Nous partitionnons tout d'abord l'ensemble de quadruplets $\mathcal{Q}$ donné en entrée :

- en un ensemble $\mathcal{Q}'$ de quadruplets construit de la manière suivante : pour tout ensemble de quatre feuilles $\{a,b,c,d\} \in X$ on choisit aléatoirement un des quadruplets de $\mathcal{Q}_{|\{a,b,c,d\}}$ (il en existe au moins un car $\mathcal{Q}$ est dense) que l'on ajoute à $\mathcal{Q}'$
- et l'ensemble des quadruplets restants $\mathcal{Q}'' = \mathcal{Q} - \mathcal{Q}'$.

Nous reconstruisons alors le SN-arbre non enraciné de $\mathcal{Q}'$ en temps $O(n^4)$, à l'aide de l'**algorithme** Q^* [Berry et Gascuel, 2000]. Finalement, pour tout quadruplet q de $\mathcal{Q}''$, nous modifions le SN-arbre non enraciné de la manière suivante :

- soit $\{a,b,c,d\}$ l'ensemble des feuilles de q, $ab|cd$ étant le quadruplet de T^* sur les feuilles $\{a,b,c,d\}$, soit u le sommet intersection, dans T^*, des chemins de a à b, de a à c, et de b à d, et soit v le sommet intersection, dans T^*, des chemins de c à d, de c à a, et de d à b,

– contractons alors toutes les arêtes sur le chemin entre u et v. Ceci assure que toutes les arêtes correspondant aux SN-bipartitions séparant $\{a,b\}$ de $\{c,d\}$ sont contractées, et qu'ainsi ces SN-bipartitions de $\mathcal{Q}'$ sont détruites, ce qui est cohérent avec la présence des deux quadruplets q et $ab|cd$ dans $\mathcal{Q}$.

Nous pouvons effectuer cette étape efficacement en temps $O(n^4)$ de la façon suivante : après un prétraitement en temps $O(n)$, il est possible de déterminer en temps constant, pour chacun des $O(n^4)$ quadruplets de $\mathcal{Q}''$, s'ils sont contenus dans T^*. L'astuce consiste à utiliser des requêtes de plus petit ancêtre commun en temps constant [Harel et Tarjan, 1984] sur un enracinement de T^*. Pour chacune des $O(n)$ contractions d'arêtes dans l'arbre T^*, le recalcul d'un enracinement de l'arbre et de la structure de données pour les requêtes de plus petit ancêtre commun s'effectue en temps $O(n)$. Ainsi, la complexité totale en temps est $O(n^4+n^2)=O(n^4)$. □

Un corollaire de cette proposition est que l'ensemble des SN-bipartitions d'un ensemble dense de quadruplets peut être construit en temps $O(n^4)$.

b) Lien entre blobs et SN-bipartitions

Nous prouvons maintenant deux lemmes avant de montrer le lien entre les SN-bipartitions de $\mathcal{Q}$ et les blobs de N.

Définition 2.2 *Pour tout blob B d'un réseau phylogénétique non enraciné N, nous appelons $E(B)$ l'ensemble des isthmes $\{e_1,\ldots,e_t\}$ qui ont un sommet dans B, comme illustré en figure 2.7. Pour toute arête $e_i=b_ic_i\in E(B)$ où $b_i\in B$ et $c_i\notin B$, nous appelons $L_B(e_i)$ l'ensemble des feuilles de la composante connexe de $N-\{e_i\}$ qui contient c_i.*

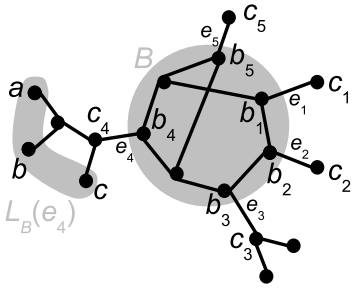


FIGURE 2.7 : Partitionnement des feuilles autour du blob B : $E(B)=\{e_1,e_2,e_3,e_4,e_5\}$.

Lemme 4 *Étant donné un réseau non enraciné N de niveau k, et un ensemble dense $\mathcal{Q}$ de quadruplets contenus dans N, alors pour toute SN-bipartition $A|\bar{A}$ de $\mathcal{Q}$, il existe un*

blob B *de* N *tel que* $E(B)$ *est partitionné en deux ensembles disjoints* E_A *et* $E_{\bar{A}}$ *tels que* $A = \bigcup_{e \in E_A} L_B(e)$ *et* $\bar{A} = \bigcup_{e \in E_{\bar{A}}} L_B(e)$.

Démonstration. Supposons par l'absurde que cette propriété est fausse. Alors aucun des isthmes de N ne sépare A de $\bar{A}$. Donc il existe deux isthmes e_1 et e_2, et deux blobs B_1 et B_2 de N, tels que les deux composantes connexes $L_{B_1}(e_1)$ et $L_{B_2}(e_2)$ de $N - \{e_1, e_2\}$ qui ne contiennent les sommets d'aucun chemin entre e_1 et e_2 sont des sous-ensembles de A. Nous pouvons choisir les isthmes e_1 et e_2 tels que $L_{B_1}(e_1)$ et $L_{B_2}(e_2)$ sont des sous-ensembles maximaux de A.

Ainsi, tout isthme e sur un chemin entre e_1 et e_2 se trouve aussi sur un chemin entre deux feuilles x_1 et x_2 de $\bar{A}$, comme montré en figure 2.8. Notons qu'un tel isthme e existe, sinon il existerait un blob B tel que les isthmes e_1 et e_2 appartiennent tous deux à $E(B)$. Le réseau N contient donc $a_1a_2|x_1x_2$, ce qui contredit le fait que $A|\bar{A}$ est une SN-bipartition.

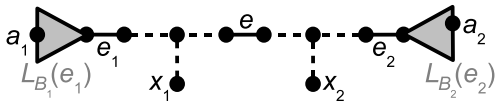


FIGURE 2.8 : Une configuration impossible si $a_1, a_2 \in A$, $x_1, x_2 \in \bar{A}$, et que $A|\bar{A}$ est une SN-bipartition.

□

Lemme 5 *Étant donné un réseau non enraciné* N *de niveau* k*, s'il existe un blob* B*, quatre arêtes distinctes* e_1*,* e_2*,* e_3 *et* $e_4 \in E(B)$*, et quatre feuilles distinctes* $a \in L_B(e_1)$*,* $b \in L_B(e_2)$*,* $c \in L_B(e_3)$*,* $d \in L_B(e_4)$*, alors au moins deux des quadruplets* $ab|cd$*,* $ac|bd$ *et* $ad|bc$ *sont contenus dans* N.

Démonstration. Appelons respectivement a', b', c' et d' les sommets de B incidents à e_1, e_2, e_3 et e_4. D'après le théorème de Menger, comme B ne contient pas d'isthme, il existe deux chemins arête-disjoints, P_1 et P_2, entre les ensembles $\{a', b'\}$ et $\{c', d'\}$, dans B. Disons que P_1 est le chemin qui contient a'.

Si P_1 est un chemin entre a' et c' et que P_2 est en chemin entre b' et d', alors nous appliquons le théorème de Menger dans B entre les ensembles $\{a', c'\}$ et $\{b', d'\}$, et trouvons ainsi deux quadruplets sur les feuilles $\{a, b, c, d\}$.

Si P_1 est un chemin entre a' et d' et que P_2 est un chemin entre b' et c', alors nous appliquons le théorème de Menger entre les ensembles $\{a', d'\}$ et $\{b', c'\}$, et trouvons ainsi deux quadruplets sur les feuilles $\{a, b, c, d\}$. □

Théorème 6 *Soit* N *un réseau non enraciné de niveau* k. *L'ensemble de ses isthmes est en bijection avec les SN-bipartitions de son ensemble de quadruplets* $Q(N)$.

Démonstration. Pour tout isthme $e = uv$ de N, le graphe $N - e$ est constitué de deux composantes connexes, qu'on appelle N_A et $N_{\bar{A}}$, où A est l'ensemble des feuilles de N_A. Comme e est un isthme, alors pour toutes feuilles $a, a' \in A$ et $x, x' \in \bar{A}$, $Q(N)_{|\{a,a',x,x'\}} = aa'|xx'$ et donc la bipartition des feuilles $A|\bar{A}$ induite par e est clairement une SN-bipartition de $Q(N)$.

Inversement, supposons par l'absurde qu'il existe une SN-bipartition $A|\bar{A}$ non représentée par un isthme de N. D'après le lemme 4, il existe un blob B de N tel que $E(B)$ est partitionné en E_A et $E_{\bar{A}}$, où $A = \bigcup_{e \in E_A} L_B(e)$ et $\bar{A} = \bigcup_{e \in E_{\bar{A}}} L_B(e)$. Comme $A|\bar{A}$ n'est pas représentée par un isthme de N, alors $|E_A| \geq 2$ et $|E_{\bar{A}}| \geq 2$, donc il existe quatre isthmes distincts $e_1, e_2 \in E_A$ et $e'_1, e'_2 \in E_{\bar{A}}$, et quatre feuilles a_1 et a_2 dans A, x_1 et x_2 dans $\bar{A}$, telles que $a_1 \in L_B(e_1)$, $a_2 \in L_B(e_2)$, $x_1 \in L_B(e'_1)$ et $x_2 \in L_B(e'_2)$. Alors, d'après le lemme 5, deux quadruplets distincts sur $\{a_1, a_2, x_1, x_2\}$ sont contenus dans N, ce qui contredit le fait que $A|\bar{A}$ est une SN-bipartition de $Q(N)$. □

Grâce à ce théorème, nous pouvons considérer le SN-arbre non enraciné de $Q(N)$ comme un résumé de N, puisque les deux ont le même ensemble d'isthmes, et diffèrent seulement par leurs blobs, qui sont de simples sommets dans le SN-arbre, et des graphes sans isthmes dans N.

2.2.4 Reconstruction dans des cas restreints

a) Depuis l'ensemble de tous les quadruplets

Étant donné l'ensemble $Q(N)$ de tous les quadruplets d'un réseau N de niveau 1, il est possible de reconstruire N en temps linéaire en la taille de l'entrée $O(|Q(N)|) = O(n^4)$. Nous montrons d'abord ceci pour des réseaux simples de niveau 1, après avoir introduit le graphe d'ordre des quadruplets.

Définition 2.3 *Étant donné un ensemble* Q *de quadruplets sur un ensemble* X *de taxons, nous définissons le* ***graphe d'ordre des quadruplets*** $G(Q) = (\{\{a, b\}, a \neq b \in X\}, \{\{a, b\}\{b, c\}, \forall d \in X, ac|bd \notin Q\})$. *Pour toute arête* $\{a, b\}\{b, c\}$ *de ce graphe, nous l'étiquetons par la feuille* b.

Cette définition est illustrée en figure 2.9(ii). Notons que $G(Q)$ est un graphe non orienté car a et c ont un rôle symétrique dans la définition des arêtes de $E(G(Q))$.

Lemme 6 *Pour un ensemble* Q *de quadruplets sur un ensemble* X *de taxons, il est possible de décider en temps polynomial s'il existe un réseau simple non enraciné* N *de niveau 1 tel*

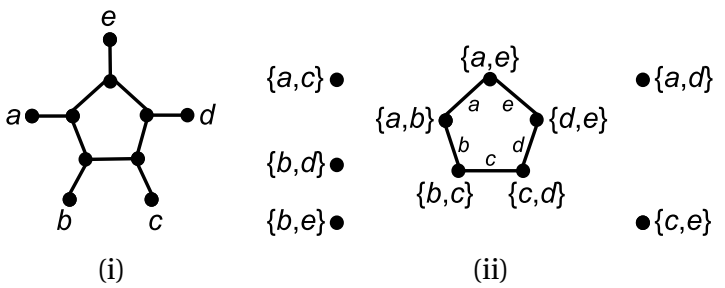


FIGURE 2.9 : Un réseau simple non enraciné N de niveau 1 (i) et le graphe d'ordre de ses quadruplets $G(\mathcal{Q}(N))$ (ii).

que $\mathcal{Q} = \mathcal{Q}(N)$*, c'est-à-dire que l'ensemble des quadruplets de* N *est exactement* $\mathcal{Q}$*. De plus, pour les instances positives, un tel réseau peut être reconstruit en temps* $O(n^4)$.

Démonstration. Dans un réseau simple non enraciné N de niveau 1, les n feuilles sont accrochées à un cycle. Appelons ces feuilles $[1..n]$ selon l'ordre de leurs positions autour du cycle à partir d'un sommet arbitraire et dans un sens arbitraire. Notre but est de trouver cet ordre étant donné $\mathcal{Q}(N)$.

Des feuilles a, b et c sont consécutives s'il n'existe pas d'autre feuille accrochée entre a et c sur le même côté du cycle que b, ce qui est équivalent au fait que $ac|bd \notin \mathcal{Q}$ pour toute feuille d de X.

Ainsi, le graphe d'ordre des quadruplets $G(\mathcal{Q}(N))$ est constitué d'un cycle de longueur n ainsi que de sommets isolés, donc l'ordre des feuilles autour du cycle de N correspond à l'ordre des étiquettes du cycle $\{a,b\}\{b,c\},\ldots,\{x,y\}\{y,a\},\{y,a\}\{a,b\}$ de $G(\mathcal{Q}(N))$.

Pour trouver cet ordre, on exécute l'algorithme *SimpleUnrootedLevel1* décrit en pseudo-code en figure 2.10. Ainsi, on construit le graphe d'ordre des quadruplets $G(\mathcal{Q}(N))$ en temps $O(n^4)$ (pour tout ensemble de trois feuilles $\{a,b,c\}$ on teste en temps $O(n)$ si une arête doit être ajoutée entre $\{a,b\}$ et $\{b,c\}$). Puis on extrait l'ordre en temps $O(n)$ par un parcours de $G(\mathcal{Q}(N))$ partant d'une arête quelconque de $G(\mathcal{Q}(N))$: si l'on n'obtient pas un cycle de longueur n alors on répond NON. Finalement, on vérifie que l'ensemble des quadruplets $\mathcal{Q}$ fourni en entrée est effectivement égal à $\mathcal{Q}(N)$, et dans ce cas on répond OUI et on renvoie l'ordre des étiquettes le long du cycle de $G(\mathcal{Q}(N))$, sinon on répond NON.

□

Théorème 7 *Pour un ensemble* $\mathcal{Q}$ *de quadruplets sur un ensemble* X *de taxons, il est possible de décider en temps polynomial s'il existe un réseau non enraciné* N *de niveau 1 tel que* $\mathcal{Q} = \mathcal{Q}(N)$*, c'est-à-dire que l'ensemble des quadruplets de* N *est exactement* $\mathcal{Q}$*. De plus, pour les instances positives, un tel réseau peut être reconstruit en temps* $O(n^4)$.

SimpleUnrootedLevel1($\mathcal{Q}$: ensemble de quadruplets, n : nombre de feuilles de $\mathcal{Q}$)
1. $G(\mathcal{Q}) \leftarrow$ nouveauGrapheStable($n(n-1)/2$ sommets numérotés de 1 à $n(n-1)/2$) ;
2. **Pour** i de 1 à n **faire**
3. **Pour** j de $i+1$ à n **faire**
4. **Pour** k de $j+1$ à n **faire**
5. arete_ij_jk $\leftarrow$ VRAI ;
6. arete_ik_kj $\leftarrow$ VRAI ;
7. arete_ki_ij $\leftarrow$ VRAI ;
8. **Pour** l de 1 à n **faire**
9. **Si** $l \neq i$ et $l \neq j$ et $l \neq k$ **alors**
10. **Si** $ik|jl \in \mathcal{Q}$ **alors**
11. arete_ij_jk $\leftarrow$ FAUX ;
12. **Si** $ij|kl \in \mathcal{Q}$ **alors**
13. arete_ik_kj $\leftarrow$ FAUX ;
14. **Si** $jk|il \in \mathcal{Q}$ **alors**
15. arete_ij_ik $\leftarrow$ FAUX ;
16. **Si** arete_ij_jk **alors**
17. $G(\mathcal{Q})$.ajouteArête($\frac{(j-1)(j-2)}{2}+i, \frac{(k-1)(k-2)}{2}+j$) ;
18. **Si** arete_ik_kj **alors**
19. $G(\mathcal{Q})$.ajouteArête($\frac{(k-1)(k-2)}{2}+i, \frac{(k-1)(k-2)}{2}+j$) ;
20. **Si** arete_ki_ij **alors**
21. $G(\mathcal{Q})$.ajouteArête($\frac{(j-1)(j-2)}{2}+i, \frac{(k-1)(k-2)}{2}+i$) ;
22. $\{a,b\} \leftarrow G(\mathcal{Q})$.arêteQuelconque ; chercheCycle $\leftarrow$ VRAI ;
23. ordre = pileVide ; ordre.empile(a) ; ordre.empile(b) ; feuillesTrouvées $\leftarrow 2$;
24. **Tant que** feuillesTrouvées $< n$ et chercheCycle **faire**
25. $c \leftarrow 0$; continue $\leftarrow$ VRAI ;
26. **Tant que** continue et $c < n$ **faire**
27. $c \leftarrow c+1$;
28. $i \leftarrow \min(a,b)$; $j \leftarrow \max(a,b)$; $k \leftarrow \min(b,c)$; $l \leftarrow \max(b,c)$;
29. **Si** $G(\mathcal{Q})$.contientArête($\frac{(j-1)(j-2)}{2}+i, \frac{(l-1)(l-2)}{2}+k$)) **alors**
30. ordre.empile(c) ; feuillesTrouvées $\leftarrow$ feuillesTrouvées + 1 ; continue $\leftarrow$ FAUX ;
31. $a \leftarrow b$; $b \leftarrow c$;
32. **Sinon** chercheCycle $\leftarrow$ FAUX ;
33. $N \leftarrow$ cycle auquel sont accrochées les feuilles dans l'ordre ordre ;
34. **Si** $\mathcal{Q} = \mathcal{Q}(N)$ **alors**
35. **Renvoyer** N ;
36. **Sinon** Pas de solution

FIGURE 2.10 : L'algorithme *SimpleUnrootedLevel1* renvoie un réseau simple non enraciné de niveau 1, s'il en existe un, dont l'ensemble de quadruplets est exactement $\mathcal{Q}$. Le graphe d'ordre des quadruplets $G(\mathcal{Q})$ est codé de la manière suivante : les feuilles étant numérotées de 1 à n, le sommet $\{i,j\}$ de $G(\mathcal{Q})$, pour $i < j$, est numéroté $\frac{j(j-1)}{2}+i$.

Démonstration. Nous construisons tout d'abord le SN-arbre non enraciné de $\mathcal{Q}$ en nous appuyant sur le résultat de la proposition 9. D'après le théorème 6, les isthmes de toute solution N sont en bijection avec les arêtes du SN-arbre non enraciné de $\mathcal{Q}$. Ainsi, comme les sommets de N ont degré au plus 3, tout sommet de degré au moins quatre dans le SN-arbre non enraciné correspond à un blob B de N avec au moins quatre sommets. Soit u un sommet de degré au moins quatre du SN-arbre non enraciné de $\mathcal{Q}$, et $B(u)$ le blob associé. Alors pour tout ensemble d'isthmes distincts e_a, e_b, e_c et e_d de $E(B(u))$, pour toutes feuilles $l_a \in L_B(e_a)$, $l_b \in L_B(e_b)$, $l_c \in L_B(e_c)$ et $l_d \in L_B(e_d)$, $\mathcal{Q}_{|\{l_a,l_b,l_c,l_d\}}$ doit être contenu dans le réseau reconstruit pour le blob $B(u)$. Ainsi, pour tout sommet u de degré $d \geq 4$, on construit en temps $O(d^4)$ le réseau simple non enraciné de niveau 1 qui contient $\mathcal{Q}_{|\{l_1,\ldots,l_t\}}$, à l'aide de l'algorithme *SimpleUnrootedLevel1* du lemme 6. Si cette étape échoue pour l'un des sommets de degré au moins quatre du SN-arbre, alors on répond NON. Sinon on construit N en remplaçant ces sommets par le réseau simple reconstruit pour l'ensemble de quadruplets $\mathcal{Q}_{|\{l_1,\ldots,l_t\}}$. Finalement, on vérifie si $\mathcal{Q} = \mathcal{Q}(N)$. Si c'est le cas, on répond OUI en renvoyant N, sinon on répond NON.

La complexité totale de cet algorithme est en temps $O(n^4)$. □

b) Depuis un ensemble dense de quadruplets

Nous montrons dans cette section que dans le cas où un ensemble dense de quadruplets est contenu dans un réseau de niveau 1, même si l'on ne connaît pas tous les quadruplets de ce réseau, il reste possible de séparer les blobs à partir des quadruplets, comme en section 2.2.3.

Lemme 7 *Si un ensemble dense $\mathcal{Q}$ de quadruplets est contenu dans un réseau non enraciné N de niveau 1, alors il est contenu dans un réseau non enraciné N' de niveau 1 dont les isthmes sont en bijection avec les SN-bipartitions de $\mathcal{Q}$.*

Démonstration. Pour tout isthme e de N, de même que dans la démonstration du théorème 6, la bipartition des feuilles induite par e est clairement une SN-bipartition de $\mathcal{Q}$. Supposons maintenant par l'absurde qu'il existe une SN-bipartition $A|\bar{A}$ de $\mathcal{Q}$ qui ne correspond à aucun isthme de N.

D'après le lemme 4, nous savons qu'il existe un blob C de N (en fait, un cycle, puisque N est de niveau 1) tel que $E(B)$ est partitionné en E_A et $E_{\bar{A}}$, où $A = \bigcup_{e \in E_A} L_B(e)$ et $\bar{A} = \bigcup_{e \in E_{\bar{A}}} L_B(e)$. Étiquetons par X tout sommet de B incident à un isthme de E_X, avec $X \in \{A, \bar{A}\}$, comme montré sur la figure 2.11.

Prouvons que les sommets étiquetés par A apparaissent consécutivement autour de C, c'est-à-dire que le sous-graphe de C induit par les sommets étiquetés par A est un chemin. Supposons que ce n'est pas le cas, alors on peut trouver deux sommets a'_1 et a'_2 étiquetés par A tels que sur les deux chemins entre eux dans C, il existe un sommet étiqueté par $\bar{A}$.

On appelle ces deux sommets intercalés x'_1 et x'_2, comme illustré en figure 2.11. On appelle e_1, e_2, e'_1 et e'_2 les isthmes incidents respectivement à a'_1, a'_2, x'_1 et x'_2. Alors il existe quatre feuilles a_1, $a_2 \in A$, x_1, $x_2 \in \bar{A}$ telles que $a_1 \in L_C(e_1)$, $a_2 \in L_C(e_2)$, $x_1 \in L_C(e'_1)$ et $x_2 \in L_C(e'_2)$. Ceci est impossible car dans ce cas, $a_1a_2|x_1x_2$ n'appartient pas à $\mathcal{Q}$, et donc $A|\bar{A}$ n'est pas une SN-bipartition de $\mathcal{Q}$.

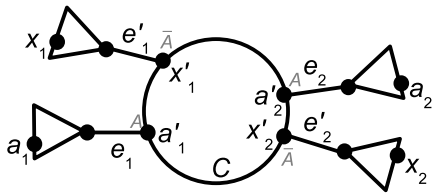


FIGURE 2.11 : Une configuration impossible dans un réseau non enraciné de niveau 1 si $a_1, a_2 \in A$, $x_1, x_2 \in \bar{A}$ et que $A|\bar{A}$ est une SN-bipartition.

Montrons maintenant que comme l'ensemble des sommets étiquetés par A est contigu, alors il est possible de transformer N en un autre réseau N' de niveau 1 qui contient toujours $\mathcal{Q}$, par une opération de **découpage de cycles**. Comme montré en figure 2.12(b), on découpe le cycle en deux cycles reliés par un isthme. Sur un des cycles appelé C_A, on attache, dans le même ordre que dans C, les isthmes incidents aux sommets de C ayant un sommet étiqueté par A, et sur l'autre cycle, $C_{\bar{A}}$, on attache les isthmes incidents aux sommets de C ayant un sommet étiqueté par $\bar{A}$, dans le même ordre que dans C. Si l'un de ces deux cycles a moins de quatre sommets, alors on le contracte en un sommet, comme $C_{\bar{A}}$ en figure 2.12(b).

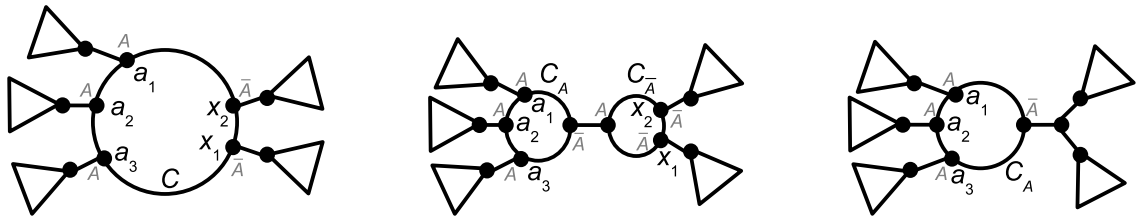


FIGURE 2.12 : L'opération de découpage des cycles : l'unique cycle du réseau non enraciné N de niveau 1, dont les sommets sont étiquetés, en gris, à la fois par A et $\bar{A}$ (a), peut être découpé en deux (b) pour faire représenter la SN-bipartition $A|\bar{A}$ de $\mathcal{Q}$ par un isthme, puis le cycle $C_{\bar{A}}$, qui a moins de quatre sommets, est contracté (c).

Nous pouvons maintenant vérifier que le réseau obtenu, dans lequel $A|\bar{A}$ est représenté par un isthme, contient bien $\mathcal{Q}$. En effet, l'opération de découpage de cycles n'a aucune conséquence sur les quadruplets qui contiennent zéro ou une feuille de A, ou,

symétriquement, de $\bar{A}$, car l'ordre des sommets le long du cycle a été conservé. Pour les quadruplets ayant deux feuilles a_1, a_2 dans A et deux feuilles x_1, x_2 dans $\bar{A}$, nous savons qu'ils doivent être en fait de la forme $a_1a_2|x_1x_2$ puisque $A|\bar{A}$ est une SN-bipartition. Ces quadruplets sont bien contenus dans le réseau après l'opération de découpage de cycles.

Ainsi, après avoir appliqué l'opération de découpage de cycles à N pour toute SN-bipartition qui n'y est pas représentée par un isthme, on obtient le réseau non enraciné N' de niveau 1, qui a bien les propriétés voulues. □

c) Réseaux simples de niveau 1

Même s'il est possible de déduire la structure globale (l'arbre de blobs) d'un réseau non enraciné de niveau 1 qui contient un ensemble dense $\mathcal{Q}$ de quadruplets - s'il en existe un - la complexité de la reconstruction d'un réseau non enraciné de niveau 1 à partir de $\mathcal{Q}$ reste un problème ouvert. Nous donnons ci-dessous des propriétés qui montrent que ce problème pourrait être algorithmiquement difficile car certains ensembles denses de quadruplets sont contenus dans des réseaux simples de niveau 1 aux structures très différentes. Ainsi, la restriction de densité pourrait bien être trop faible pour la reconstruction des réseaux non enracinés de niveau 1, car elle ne fixe pas nécessairement la structure des feuilles à l'intérieur des cycles du réseau.

Notons que le problème de reconstruction d'un réseau simple non enraciné de niveau 1 à partir d'un ensemble $\mathcal{Q}$ de quadruplets est équivalent au problème suivant : trouver un ordre σ des feuilles telles que pour tout quadruplet $ab|cd \in \mathcal{Q}$ tel que $\sigma(a) < \sigma(b)$ et $\sigma(c) < \sigma(d)$, ni $\sigma(a) < \sigma(c) < \sigma(b) < \sigma(d)$, ni $\sigma(c) < \sigma(a) < \sigma(d) < \sigma(b)$. Cette formulation est similaire à celle du problème NON-BETWEENNESS qui est NP-complet dans le cas général [Guttmann et Maucher, 2006], mais dont la complexité n'est pas connue dans le cas dense.

Proposition 10 *Pour tous réseaux simples non enracinés N_1 et N_2 de niveau 1, il existe un ensemble dense $\mathcal{Q}$ de quadruplets tel que $\mathcal{Q} \subseteq \mathcal{Q}(N_1)$ et $\mathcal{Q} \subseteq \mathcal{Q}(N_2)$.*

Démonstration. Pour tout ensemble de quatre feuilles $\{a, b, c, d\}$, N_1 et N_2 contiennent, tous deux, deux des trois quadruplets sur $\{a, b, c, d\}$. Ainsi, ils partagent au moins un quadruplet. □

Il est même possible de construire un ensemble dense de quadruplets qui est contenu dans un nombre exponentiel (par rapport au nombre de feuilles) de réseaux simples non enracinés de niveau 1.

Proposition 11 *Pour tout entier $n \geq 3$, il existe un ensemble dense de quadruplets sur $2n$ feuilles qui est contenu dans 2^n réseaux simples non enracinés de niveau 1 non isomorphes.*

Démonstration. Considérons l'ensemble de feuilles $\{x_i, i \in [1..2n]\}$. Définissons des "blocs" $B_i = \{x_{2i-1}, x_{2i}\}$, et le réseau simple non enraciné N de niveau 1 obtenu en accrochant les feuilles $x_1 \dots x_{2n}$ autour d'un cycle, dans l'ordre $1, 2, \dots 2n$.

Considérons maintenant l'ensemble Q de quadruplets défini de la manière suivante, contenu dans N. Pour tout ensemble de quatre feuilles $a, b, c, d \in [1..2n]$:

- cas 1) si les quatre feuilles appartiennent à différents blocs B_i, alors, en considérant que $a < b < c < d$, on ajoute $ab|cd$ et $bc|ad$ dans Q.
- cas 2) si exactement deux feuilles (disons a et b) appartiennent à un même bloc B_i, alors on ajoute $ab|cd$ dans Q.
- cas 3) sinon, deux feuilles (disons a et b) appartiennent à un même bloc B_i, et deux autres (c et d) appartiennent à un bloc B_j avec $i < j$, alors on ajoute $ab|cd$ dans Q.

Notons que dans cette construction, deux feuilles appartenant à un même bloc B_i, ont une position symétrique dans les quadruplets de Q. Ainsi, tout autre réseau obtenu en accrochant les feuilles dans un ordre identique à celui de N, modulo des transpositions à l'intérieur des blocs B_i, contiendra toujours Q. Comme il y a n blocs B_i de taille 2, il y a n transpositions possibles, et donc 2^n réseaux simples non enracinés de niveau 1 qui contiennent Q. □

En complément de ces résultats, nous avons étudié une approche par **obstructions** pour déterminer si un ensemble dense Q de quadruplets est contenu dans un réseau de niveau 1. Cette approche consiste à identifier un ensemble fini de configurations de quadruplets interdites : les obstructions. Plus formellement, il s'agit de trouver un ensemble $\mathfrak{Q}_{\text{interdits}}$ d'ensembles de quadruplets sur un nombre constant de feuilles tel que Q est contenu dans un réseau de niveau 1 si et seulement si aucun élément de $\mathfrak{Q}_{\text{interdits}}$, quel que soit le renommage des feuilles, n'est sous-ensemble Q.

Pour terminer sur les propriétés non intuitives des ensembles de quadruplets contenus dans un réseau simple de niveau 1, la table 2.1 énumère tous les ensembles **minimalement denses** de quadruplets (c'est-à-dire tels qu'il existe exactement un quadruplet sur chaque ensemble de quatre feuilles) sur cinq feuilles, à isomorphisme près. On remarque que tous sont contenus dans au moins un réseau simple non enraciné de niveau 1. Ainsi, d'éventuelles obstructions (nous aborderons plus précisément ce concept en section 3.1.1) de taille inférieure ou égale à 5 sur les ensembles de quadruplets à la reconstruction de réseaux non enracinés de niveau 1 contiendront nécessairement au moins deux quadruplets sur un même ensemble de feuilles.

Enfin, un résultat négatif de Grünewald *et al.* [2009] peut être mis en relation avec cette recherche d'obstructions de taille fixe. Nous avons prouvé dans le théorème 3 que l'ensemble de bipartitions d'un réseau simple de niveau 1 est circulaire. Ainsi, l'ensemble des ensembles de bipartitions des réseaux simples de niveau 1 est inclus dans l'ensemble des ensembles de bipartitions circulaires. Ce dernier est toutefois caractérisable uniquement par des obstructions de taille non bornée. Ce résultat est même donné dans un cas plus gé-

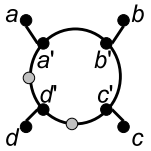
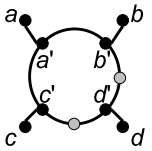
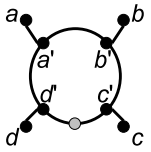
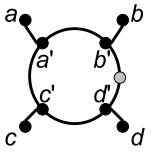
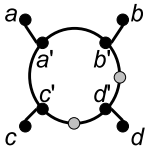
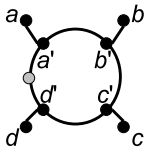
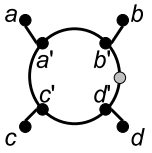
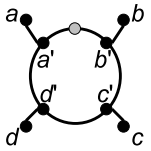
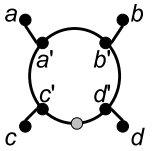
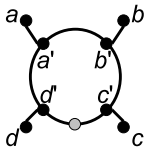
ensemble de quadruplets	ordre a', b', c', d'	ordre a', b', d', c'
$ab\|cd, ab\|ce, ab\|de, ac\|de, bc\|de$		
$ab\|cd, ab\|ce, ac\|de, ad\|be, bc\|de$		
$ab\|cd, ab\|de, ac\|be, ae\|de, bc\|de$	$\emptyset$	
$ab\|cd, ab\|de, ae\|bc, ae\|cd, bc\|de$		$\emptyset$
$ab\|cd, ac\|be, ac\|de, ad\|be, ab\|de$	$\emptyset$	
$ab\|cd, ac\|be, ad\|ce, ae\|bd, bc\|de$		
$ab\|cd, ac\|de, ad\|be, ae\|bc, bc\|de$		$\emptyset$

TABLE 2.1 : Les 7 ensembles de quadruplets minimalement denses (à isomorphisme près) et les ensembles de réseaux simples de niveau 1 qui les contiennent. Les sommets gris correspondent aux emplacements possibles du voisin e' de la feuille e. Le quadruplet $ab|cd$ correspond à deux ordres possibles, le long du cycle, pour les voisins respectifs a', b', c' et d' de a, b, c et d : a', b', c', d' et a', b', d', c'.

néral où des poids sont donnés aux bipartitions. Grünewald et al. étendent ce résultat aux **fonctions de poids des quadruplets** (qui à chaque quadruplet $ab|cd$ associent la somme des poids des bipartitions qui le contiennent), en montrant que les fonctions de poids des quadruplets ne peuvent être caractérisées par des obstructions de taille bornée.

2.3 Reconstruction à partir de clades

Dans cette section, nous nous intéressons à la question de la reconstruction d'un réseau phylogénétique explicite à partir de son ensemble de clades souples. Ce problème de reconstruction, si l'on cherche une solution avec un nombre minimal de sommets hybrides, est appelé MINRETCLUSTERS, et il est NP-difficile et même APX-difficile [van Iersel et Kelk, 2011].

Nous nous intéressons donc à sa restriction à une classe particulière de réseaux enracinés, les réseaux à une couche de réticulation, et proposons une approche heuristique en deux étapes. Bien que cette approche soit heuristique, elle consiste à résoudre successivement deux problèmes combinatoires difficiles de manière exacte, ces deux problèmes étant choisis pour obtenir une solution raisonnable d'un point de vue biologique (en commençant par construire un arbre sur la portion des données qui ne contient pas de conflits entre clades, puis en ajoutant un nombre minimal d'arcs d'hybridation).

Mentionnons que le problème de reconstruction de réseaux de niveau k à partir d'un ensemble $\mathcal{C}$ de clades souples sur un ensemble X de taxons est également NP-difficile et APX-difficile si l'on cherche à minimiser le niveau k [van Iersel et Kelk, 2011]. Le problème de reconstruction peut toutefois être résolu en temps $O(|X|^{3k+2}|\mathcal{C}|)$ pour $k \geq 2$ [van Iersel *et al.*, 2010a]. Cet algorithme, appelé CASS, qui parvient plus généralement à construire, avec la même complexité en temps, un réseau de niveau k contenant les clades souples donnés (sans garantie que ce k soit alors minimal, cette propriété est seulement conjecturée par ses auteurs), est implémenté dans le logiciel Dendroscope sous le nom "Minimum Network".

Dans les solutions trouvées par CASS, le nombre de sommets hybrides est généralement moins important que dans les celles trouvées avec l'algorithme que nous présentons ci-dessous [Huson *et al.*, 2009] (également implémenté dans Dendroscope sous le nom "Galled Network"), mais au prix de temps de calcul beaucoup plus élevé sur certaines instances.

2.3.1 Test de compatibilité

Tout comme pour les triplets et quadruplets, la première question à étudier, pour une méthode combinatoire de reconstruction phylogénétique, est de vérifier que le réseau obtenu correspond bien aux données en entrée.

Cependant, contrairement aux triplets et aux quadruplets, pour lesquels cette vérification peut s'effectuer en temps polynomial, pour les clades souples ce problème est NP-complet en toute généralité. En effet, étant donnés un réseau phylogénétique explicite enraciné N sur un ensemble X de taxons et un sous-ensemble $\mathcal{C}$ de X, le problème CLUSTER CONTAINMENT, qui consiste à déterminer si N contient un clade souple $C \subseteq X$, est un problème NP-complet [Kanj *et al.*, 2008]. Il reste NP-complet pour les réseaux phylogénétiques réguliers, et sans fratrie hybride [van Iersel *et al.*, 2010b].

En revanche, dans la classe des réseaux normaux et des réseaux de niveau k (pour k fixé), le problème peut être résolu en temps polynomial [van Iersel *et al.*, 2010b], tout comme dans la classe des réseaux sans descendance hybride [Huson *et al.*, 2011]. Nous prouvons la même propriété pour les réseaux à une couche de réticulation.

Proposition 12 *Le problème* CLUSTER CONTAINMENT *peut être résolu en temps polynomial sur les réseaux à une couche de réticulation.*

Démonstration. Montrons comment déterminer si un arc $x = (v,w)$ représente un clade souple C dans un réseau N à une couche de réticulation. Soit L_x l'ensemble des feuilles f telles qu'il existe un chemin orienté de w à f, constitué uniquement d'arcs de spéciation, comme illustré en figure 2.13. Soit R_x l'ensemble des sommets hybrides r tels qu'il existe un chemin orienté de w à r, constitué éventuellement d'arcs de spéciation suivis d'exactement un arc d'hybridation. On partitionne R_x en deux sous-ensembles R_x^1 (constitué des sommets hybrides dont tous les parents sont descendants de w) et R_x^2 (constitué des autres sommets). On appelle Y_r l'ensemble des descendants d'un sommet r de R_x.

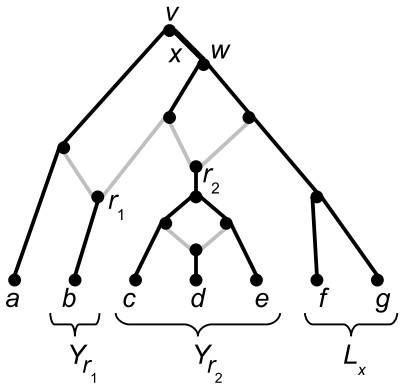


FIGURE 2.13 : Un réseau N à une couche de réticulation dont les arcs d'hybridation sont indiqués en gris. Les clades souples représentés par e contiennent tous L_x. Ils contiennent tous également Y_{r_2} car $r_2 \in R_x^1$, mais contiennent Y_{r_1} de manière optionnelle car $r_1 \in R_x^2$: ainsi, $\mathcal{S}_N(w) = \{L_x \cup Y_{r_2}, L_x \cup Y_{r_1} \cup Y_{r_2}\}$.

Les taxons de L_x, et les Y_r pour $r \in R^1_x$, sont contenus dans tous les clades souples représentés par x. Comme tout sommet hybride r d'un réseau à une couche de réticulation déconnecte tous ses descendants des autres sommets du réseau, deux cas se présentent pour chaque sommet r de R^2_x. En effet, l'ensemble Y_r des taxons descendants de r est optionnel pour les clades souples représentés par r, au sens où ces derniers contiennent soit tout l'ensemble Y_r, soit aucun taxon de Y_r.

Finalement, l'arc x représente le clade C si et seulement si il existe un ensemble $R \subseteq R^2_x$ tel que $C = L_x \cup \bigcup_{r \in R^1_x} Y_r \cup \bigcup_{r \in R} Y_r$.

Ainsi, pour déterminer si un sous-ensemble de taxons C est un clade souple de N, on détermine, pour chacun des $O(m)$ arcs, les ensembles L_x et Y_r en temps $O(m)$. Cette étape en temps $O(m^2)$ peut être effectuée en pré-traitement pour être réutilisée pour d'autres tests de présence de clades souples. Puis, on applique l'algorithme donné en pseudo-code en figure 2.14. Pour tout arc x de N, on vérifie si L_x et tous les Y_r, pour $r \in R^1_x$, sont inclus dans C. Si ce n'est pas le cas, on répond NON. Si c'est le cas, on considère successivement chacun des Y_r, pour $r \in R^2_x$. Si l'un d'eux chevauche C alors on répond NON. Sinon, tous les Y_r sont chacun soit inclus dans C, soit disjoints de C. C est donc bien un clade souple de N. La complexité en temps de cet algorithme est $O(m^2)$, en implémentant de manière efficace les tests de chevauchement grâce à un pré-étiquetage en temps $O(n)$ des feuilles appartenant à C (pour pouvoir ensuite répondre en temps constant à un test de présence dans C d'une feuille donnée). □

2.3.2 Décomposition des réseaux phylogénétiques

Soit $\mathcal{C}$ un ensemble de clades sur l'ensemble X de taxons. On définit le **graphe d'incompatibilité** de $\mathcal{C}$ comme le graphe $IG(\mathcal{C}) = (\mathcal{C}, \{\{A,B\}, A-B \neq \emptyset, B-A \neq \emptyset, A \cap B \neq \emptyset\})$, c'est-à-dire que deux sommets sont adjacents si les clades qu'ils représentent se chevauchent, autrement dit ne peuvent être tous deux contenus dans un même arbre phylogénétique. Un exemple de graphe d'incompatibilité est donné en figure 2.18(a) à la page 95.

Soit N un réseau à une couche de réticulation qui contient un ensemble $\mathcal{C}$ de clades souples. Comme un clade souple $\mathcal{C}$ peut être représenté par plus d'un arc dans N, on définit une **assignation d'arcs** ϵ comme une bijection entre chaque clade souple $C \in \mathcal{C}$ et un arc de spéciation $\epsilon(C)$ de N qui le représente.

On dit que N est une **représentation décomposable** de $\mathcal{C}$, ou plus simplement que N est **décomposable**, s'il existe une assignation d'arcs ϵ telle que pour toute paire de clades souples $A, B \in \mathcal{C}$, les deux arcs $\epsilon(A)$ et $\epsilon(B)$ sont situés dans le même blob de N si et seulement si A et B sont dans la même composante connexe du graphe d'incompatibilité $IG(\mathcal{C})$.

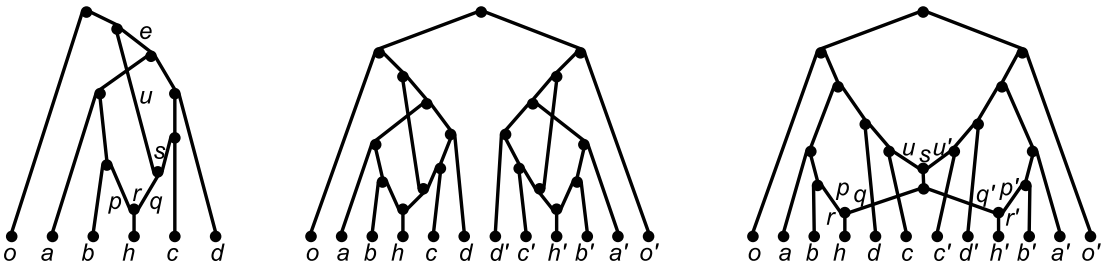
Par définition, les réseaux de clades stricts [Huson et Rupp, 2008] et les réseaux de bipartitions [Bandelt et Dress, 1992a] sont décomposables. Toutefois, un exemple de Gusfield *et al.* [2007] illustré en figure 2.15 montre que les réseaux de clades souples ne le sont

```
GalledClusterContainment(N : réseau à une couche de réticulation sur X, C : clade de X)
 1. Pour tout élément f de X faire
 2.     dansC(f) ←FAUX;
 3. Pour tout élément f de C faire
 4.     dansC(f) ←VRAI;
 5. appartient ← FAUX;
 6. Pour tout arc x de N faire
 7.     inclusion ← VRAI;
 8.     Pour toute feuille f de L_x faire
 9.         Si NON(dansC(f)) alors
10.             inclusion ← FAUX;
11.     Pour tout sommet r de R^1_x faire
12.         Pour toute feuille f de L_x faire
13.             Si NON(dansC(f)) alors
14.                 inclusion ← FAUX;
15.     Si inclusion alors
16.         testYr ← VRAI;
17.         Pour tout sommet r de R^2_x faire
18.             inclus ← VRAI;
19.             Pour toute feuille f de Y_r faire
20.                 Si NON(dansC(f)) alors
21.                     inclus ← FAUX;
22.             disjoint ← VRAI;
23.             Pour toute feuille f de Y_r faire
24.                 Si dansC(f) alors
25.                     disjoint ← FAUX;
26.             Si NON(inclus ou disjoint) alors
27.                 testYr ← FAUX;
28.         Si testYr alors
29.             appartient ← VRAI;
30. Renvoyer appartient;
```

FIGURE 2.14 : L'algorithme *GalledClusterContainment* qui détermine si $\mathcal{C}$ est un clade souple d'un réseau phylogénétique N à une couche de réticulation. Les ensembles L_x, R^1_x, R^2_x, et Y_r se calculent chacun en temps $O(m)$ par un simple parcours du graphe orienté N.

pas nécessairement. Il est en effet possible d'économiser parfois un sommet hybride en remplaçant deux blobs, chacun avec deux sommets hybrides, comme dans le réseau N_1 de la figure 2.15(ii), par un seul blob avec trois sommets hybrides, comme dans le réseau N_2 de la figure 2.15(iii). Toutefois, même si le réseau N_2 est plus parcimonieux en termes de nombre de sommets hybrides, N_1 peut sembler plus approprié du point de vue biologique en séparant les feuilles de $S = \{o, a, b, c, d\}$ de celles de $S' = \{o', a', b', c', d'\}$ dans deux blobs distincts. En effet, N_2 renforce la proximité entre les feuilles étiquetées x et celles

étiquetées x', alors qu'aucun des clades parmi ceux fournis en entrée ne fait apparaître ensemble des éléments de S et de S'.



(i) (ii) (iii)

FIGURE 2.15 : Un réseau minimum N qui contient les clades souples $\mathcal{C} = \big\{\{a\}, \{b\}, \{c\}, \{d\}, \{o\}, \{h\}, \{a,b\}, \{a,b,h\}, \{b,h\}, \{c,d\}, \{c,d,h\}, \{a,b,c,d,h\}, \{a,b,c,d\}\big\}$ en utilisant deux sommets hybrides r et s (i). Notons que le rôle de l'arc d'hybridation u est de permettre que le clade souple $\{a,b,c,d\}$ soit représenté par l'arc e. Deux copies de N combinées en un réseau décomposable N_1 (ii) nécessitant 4 sommets hybrides pour contenir tous les clades souples de $\mathcal{C}$, ainsi qu'un second ensemble similaire $\mathcal{C}'$ sur les feuilles de $S' = \{o', a', b', c', d', h'\}$. Un autre réseau N_2 (iii), non décomposable, qui contient aussi les clades souples de $\mathcal{C} \cup \mathcal{C}'$, mais n'a que 3 sommets hybrides. Toutefois, il est de niveau 3 et relie des feuilles assez éloignées, alors que le réseau N_1 est de niveau 2.

Ainsi, à la fois pour éviter ce problème, et pour des raisons algorithmiques, nous proposons de nous limiter à des représentations décomposables. Nous allons donc décomposer la reconstruction de la façon suivante [Gusfield et Bansal, 2005; Huson *et al.*, 2005]. Supposons qu'un ensemble $\mathcal{C}$ de clades de X est donné en entrée. Nous calculons tout d'abord les composantes connexes de $IG(\mathcal{C})$ (soit de façon naïve en temps $O(|\mathcal{C}|^2)$ en construisant le graphe d'incompatibilité, soit directement en temps sous-quadratique [Charbit *et al.*, 2008]).

Puis, pour toute composante connexe non triviale $\mathcal{C}' \subseteq \mathcal{C}$ de $IG(\mathcal{C})$ on calcule un réseau phylogénétique enraciné N' pour $\mathcal{C}'$. Finalement tous ces réseaux sont connectés pour devenir les blobs d'un réseau N pour $\mathcal{C}$.

On dit que deux taxons $x, y \in X$ sont **séparés** dans $\mathcal{C}$ s'il existe un clade $A \in \mathcal{C}$ tel que $|\{x,y\} \cap A| = 1$. En considérant chaque sous-problème $\mathcal{C}'$, on identifie, en les contractant en un seul taxon-représentant, les ensembles de taxons qui ne sont pas séparés par un clade de $\mathcal{C}'$. A la fin de reconstruction du réseau phylogénétique, on remplacera la feuille correspondant à chaque taxon-représentant par une multifurcation adjacente à l'ensemble des taxons non séparés qu'il représente.

Ainsi, dans la suite, on considérera qu'un ensemble $\mathcal{C}$ de clades sur X a les propriétés suivantes :

(P1) Le graphe d'incompatibilité $IG(\mathcal{C})$ n'a qu'une composante connexe.

(P2) Toute paire de taxons de X est séparée dans $\mathcal{C}$.

En effet, notre algorithme traite successivement chaque composante connexe du graphe d'incompatibilité.

Comme expliqué en section 1.4.2, on peut considérer les réseaux à une couche de réticulation comme un arbre phylogénétique auquel on ajoute des réticulations.

Ainsi, nous allons maintenant procéder en deux étapes pour reconstruire le réseau :

- déterminer un ensemble minimal R de feuilles à enlever pour éliminer les conflits dans les clades, c'est le problème MAXIMUM COMPATIBLE SUBSET,
- attacher avec un nombre minimal d'arcs les taxons impliqués dans des conflits à l'arbre reconstruit sur les clades sans conflits, c'est le problème MINIMUM ATTACHMENT.

La première étape sera détaillée en section 2.3.3 et la seconde en section 2.3.4.

2.3.3 Recherche d'un ensemble maximum de taxons compatibles

a) Complexité du problème

Problème 2 (MAXIMUM COMPATIBLE SUBSET (MCS))
Entrée : *un ensemble $\mathcal{C}$ de clades sur un ensemble X de taxons.*
Sortie : *le plus petit ensemble R de taxons de X tel que l'ensemble de clades $\mathcal{C}_{|X-R}$ ne contienne pas de chevauchement.*

Pour un ensemble quelconque de clades, ce problème est équivalent au problème MAXIMUM COMPATIBLE TREE[9] (MCT)qui est NP-complet [Steel et Hamel, 1996]. Nous appellerons **MCS-r** (MCS-restreint) le problème MCS pour lequel les instances sont réduites aux ensembles de clades $\mathcal{C}$ sur X pour lesquels les propriétés (P1) et (P2) sont vérifiées. Nous montrons maintenant que ce problème MCS-r est NP-complet.

Théorème 8 (NP-complétude de MCS-r) *Soit $\mathcal{C}$ un ensemble de clades de X avec les propriétés (P1) et (P2). Résoudre MCS-r sur ce type d'instances est un problème NP-complet.*

Démonstration. Réduisons le problème MCS au problème MCS-r.

Soit X un ensemble de taxons et $\mathcal{C}$ une instance du problème MCS. Soient $X' = X \cup \{o\}$, où o est un nouveau taxon distinct de ceux de X, et $\mathcal{C}_1$ l'ensemble de tous les clades triviaux. Définissons $\mathcal{C}' = (\mathcal{C} \setminus \mathcal{C}_1) \cup \{X\} \cup \{\{o, x\} \mid x \in X\}$. Notons que par construction, l'ensemble de clades $\mathcal{C}'$ de X' a les propriétés (P1) et (P2). Montrons qu'il existe une solution R' de taille $k+1$ au problème MCS-r pour $\mathcal{C}'$ si et seulement si il existe une solution R de taille k au problème MCS pour $\mathcal{C}$.

9. Problème MCT (MAXIMUM COMPATIBLE TREE) : étant donné un ensemble $\mathcal{T}$ d'arbres en entrée, recherche du plus grand sous-ensemble de feuilles $S \subseteq X$ tel que pour tous les arbres $T \in \mathcal{T}$, $T_{|S}$ ont un raffinement commun.

$\Rightarrow$: soit R' une solution au problème MCS-r pour $\mathcal{C}'$ de taille $k+1$, deux cas se présentent.

Cas (1) : $o \in R'$. Alors $R = R' \setminus \{o\}$ est de taille k et supprime tous les chevauchements de $\mathcal{C}$.

Cas (2) : $o \notin R'$. Dans ce cas, l'ensemble R' doit contenir tous les éléments de X sauf un, c'est-à-dire $|R'| = k+1 = |X|-1$, car sinon il resterait deux clades de la forme $\{o,x\}$ et $\{o,y\}$, qui se chevaucheraient. Alors tout sous-ensemble $R \subseteq X$ de taille $|X|-2 = k$ convient pour $\mathcal{C}$, parce qu'une collection de clades sur seulement deux taxons ne peut contenir de chevauchements.

$\Leftarrow$: Soit R une solution du problème MCS-r pour $\mathcal{C}$, de taille k. Considérons $R' = R \cup \{o\}$. Cet ensemble a pour taille $k+1$, supprime tous les chevauchements entre les clades de type $\{o,x\}$ car il supprime le taxon o. Tous les autres chevauchements sont supprimés car $R \subseteq R'$. □

b) Algorithme de complexité paramétrée pour MCS-r

Il existe un algorithme FPT en le nombre de conflits pour résoudre le problème MCT pour un ensemble d'arbres phylogénétiques enracinés sur X [Berry et Nicolas, 2006]. Cet algorithme peut également être utilisé pour résoudre le problème MCS-r en codant chaque clade C par un arbre phylogénétique qui le contient.

Nous présentons maintenant l'algorithme FPT implémenté dans Dendroscope en septembre 2008 pour résoudre *directement* le problème MCS-r. Cet algorithme, appelé *SeedGrowing* fonctionne très bien en pratique.

Pour toute paire de clades chevauchants $A, B \in \mathcal{C}$, on définit la **déclaration d'incompatibilité** $(A-B, A \cap B, B-A)$. On note L la liste de toutes ces déclarations d'incompatibilité pour l'ensemble de clades $\mathcal{C}$ (voir figure 2.18(b) page 95).

Pour résoudre le problème MCS-r, on doit trouver un ensemble minimum de taxons $R \subseteq X$ qui **résout** toutes les déclarations d'incompatibilité dans L, c'est-à-dire que pour chacune d'entre elles, au moins un de ses trois termes est contenu dans R.

L'algorithme *SeedGrowing*, décrit en pseudo-code en figure 2.16, fonctionne en construisant un ensemble S de solutions candidates, appelées **graines**, dont il essaie d'étendre celle de taille minimale qui résout le plus de déclarations d'incompatibilité, jusqu'à en trouver une qui résout toutes les déclarations d'incompatibilités.

Théorème 9 (Complexité de l'algorithme *SeedGrowing*) *S'il existe une solution de taille* k *au problème MCS-r pour un ensemble* H *de déclarations d'incompatibilités, alors l'algorithme* SeedGrowing *la trouvera en considérant au plus* 3^{k+1} *graines. Sa complexité dans le pire cas est en temps* $O(k|H|3^k)$.

SeedGrowing(L : liste de déclarations d'incompatibilité à 3 membres)
1. $S \leftarrow \{L[1][1], L[1][2], L[1][3]\}$
2. **Pour** tout élément s de S **faire**
3. $\quad \text{rang}(s) = 1$
4. $S_{min} =$ éléments de S de taille minimale
5. $s^* =$ élément de S_{min} de rang maximal
6. **Tant que** $\text{rang}(s^*) < |L|$ **faire**
7. $\quad$ **Si** s^* résout la déclaration d'incompatibilité $L[\text{rang}(s^*)+1]$ **alors**
8. $\quad\quad \text{rang}(s^*) = \text{rang}(s^*)+1$
9. $\quad$ **Sinon**
10. $\quad\quad S \leftarrow S \cup \{s^* \cup L[\text{rang}(s^*)+1][1],\ s^* \cup L[\text{rang}(s^*)+1][2],\ s^* \cup L[\text{rang}(s^*)+1][3]\}$
11. $\quad\quad S \leftarrow S - \{s^*\}$
12. $\quad\quad S_{min} =$ éléments de S de taille minimale
13. $\quad\quad s^* =$ élément de S_{min} de rang maximal
14. **Renvoyer** s^*

FIGURE 2.16 : L'algorithme *SeedGrowing* qui recherche un ensemble de taille minimale qui résout toutes les déclarations d'incompatibilité de L.

Démonstration. Démontrons que si un élément minimum $s \in S$ contient k taxons, alors S contient au plus 3^{k+1} ensembles. Considérons l'arbre d'énumération des graines généré par l'algorithme. La **profondeur** d'un sommet v de cet arbre est le nombre d'arcs dans le chemin depuis la racine vers v.

Au début d'une itération, l'algorithme choisit une graine s de taille minimale. Par construction, la profondeur du sommet correspondant sera d'au plus $|s|$. Si s résout la déclaration d'incompatibilité suivante, alors aucun sommet n'est ajouté à l'arbre d'énumération. Dans le cas contraire, trois sommets sont ajoutés, chacun représentant une graine dont la taille est strictement plus grande que $|s|$. Ainsi, l'arbre d'énumération aura une profondeur d'au plus $k+1$, et il y aura au plus trois fois plus de sommets à la profondeur i qu'à la profondeur $i-1$. Ceci implique que le nombre de graines considérées dans S est au plus 3^{k+1}. □

c) Codage de MCS-r comme problème d'édition d'un graphe sans M

Il est également possible de coder le problème MCS-r comme celui de l'édition d'un graphe. Ceci permet de résoudre le problème avec une meilleure complexité théorique en temps, en utilisant le meilleur algorithme connu pour le problème HITTING SET.

Définition 2.4 (Graphe des caractères) *Soit* $C = \{C_1, \ldots, C_r\}$ *un ensemble de clades sur un ensemble* X *de* n *taxons. Le* ***graphe des caractères*** *de* $\mathcal{C}$ *est le graphe biparti* $G(\mathcal{C}) =$

$(\mathcal{C} \cup X, E)$, *où* $E = \{\{x, C_i\} | x \in C_i\}$. *Les sommets de* $G(\mathcal{C})$ *associés à* X *sont appelés* ***sommets-taxons****.*

Définition 2.5 (Graphe sans M) *Un graphe biparti* $G = (X \cup Y, E)$*, où* X *et* Y *sont des ensembles indépendants, est un* ***graphe sans M*** *s'il ne contient pas de graphe M, c'est-à-dire de chemin de longueur 5 avec ses deux extrémités dans* Y *comme sous-graphe induit.*

Notons que le chevauchemment de deux clades C_1 et C_2 de $\mathcal{C}$ implique l'existence de trois taxons x, y et z tels que $x \in C_1 - C_2$, $y \in C_1 \cap C_2$, $z \in C_2 - C_1$, ce qui correspond précisément à la présence d'un graphe M, comme montré en figure 2.17 à la page 2.17.

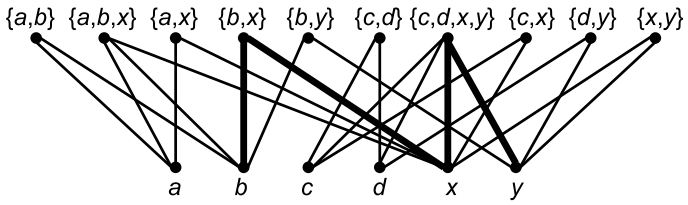


FIGURE 2.17 : Le graphe de caractères de $\mathcal{C} = \{\{a, b\}, \{a, b, x\}, \{a, x\}, \{b, x\}, \{b, y\}, \{c, d\}, \{c, d, x, y\}, \{c, x\}, \{d, y\}, \{x, y\}\}$. Les clades $\{b, x\}$ et $\{c, d, x, y\}$ étant incompatibles, ils correspondent à la présence d'un graphe M (par exemple celui sur les sommets-taxons $\{b, x, y\}$ montré en gras) dans le graphe des caractères.

Théorème 10 *Étant donné un ensemble* $\mathcal{C}$ *de clades, résoudre le problème MCS-r sur l'ensemble* $\mathcal{C}$ *est équivalent à supprimer le plus petit nombre* t *de sommets-taxons dans le graphe de caractères* $G(\mathcal{C})$ *afin qu'il devienne un graphe sans M.*

Ainsi, un simple arbre de recherche bornée sur les graphes M pour éliminer leurs sommets-taxons donne, comme l'algorithme *SeedGrowing*, une complexité en temps de $O^*(3^k)$ (pour tout graphe M, essayer de supprimer l'un de ses trois sommets-taxons) [10]. Le problème peut aussi être résolu en appliquant un algorithme FPT [11] de résolution de 3-HITTING SET sur les ensembles de trois sommets-taxons des graphes M.

Précisons que cette approche par suppression de sommets pour obtenir un graphe sans M est similaire à celle utilisée pour résoudre le problème MINIMUM FLIP CONSENSUS [Chen *et al.*, 2006] qui consiste à déterminer le nombre minimum de modifications (addition ou suppression de taxons) pour qu'un ensemble de clades ne contienne plus de chevauchement. Ce problème, qui revient à modifier le plus petit nombre d'arêtes dans le graphe de caractères afin qu'il devienne un graphe sans M, est également NP-complet et des algorithmes FPT, entre autres, ont été proposés pour le résoudre [Chen *et al.*, 2006; Böcker *et al.*, 2008; Komusiewicz et Uhlmann, 2008].

10. c'est-à-dire $O(\mathrm{poly}(n)3^k)$ où poly est une fonction polynomiale.
11. en $O^*(2.076^k)$ d'après http://fpt.wikidot.com/fpt-races

2.3.4 Ajout des réticulations

Présentons maintenant la deuxième partie de l'algorithme qui consiste à ajouter les réticulations entre la partie du réseau reconstruite sur les taxons sans conflit trouvés à la section précédente, et le reste des taxons. Un exemple est donné en figure 2.18.

Soit $\mathcal{C}$ un ensemble de clades sur X et $R \subsetneq X$ un ensemble minimum de taxons tel que la restriction $\mathcal{C}_{|X\setminus R}$ de $\mathcal{C}$ à $X \setminus R$ ne contient pas de chevauchements. Soit T un arbre phylogénétique enraciné sur $X\setminus R$ qui contient $\mathcal{C}_{|X\setminus R}$, et soit $L(e)$ le clade de $\mathcal{C}_{|X\setminus R}$ représenté par un arc e de T. Pour tout arc de spéciation e de T, soit $\mathcal{C}(e) = \{C \in \mathcal{C} \mid C - R = L(e)\}$ l'ensemble de tous les clades de $\mathcal{C}$ dont la restriction à $X\setminus R$ est $L(e)$, et soit $R(e) = \{r \in R \cap C \mid C \in \mathcal{C}(e)\}$. $R(e)$ est constitué des sommets de R qui seront contenus dans les clades souples représentés par l'arc e dans le réseau reconstruit. Nous appellerons T (l'arbre ainsi que les étiquettes $R(e)$ pour tout arc e) la **partie haute**.

Soit $\mathcal{C}_{|R}$ la restriction de $\mathcal{C}$ à l'ensemble de taxons R et soit $\widehat{\mathcal{C}}_{|R}$ les clades maximaux (pour l'inclusion) de $\mathcal{C}_{|R}$. Définissons maintenant le graphe B associé à $\widehat{\mathcal{C}}_{|R}$, de la manière suivante : chaque clade $C \in \widehat{\mathcal{C}}_{|R}$ est représenté par un sommet $\nu(C)$, chaque taxon $r \in R$ est représenté par un sommet $\nu(r)$, et on place un arc de $\nu(C)$ vers $\nu(r)$ pour tous les taxons r contenus dans le clade C. On appellera B la **partie basse**.

Dans l'exemple de la figure 2.18, où $R = \{x, y\}$, $\mathcal{C}_{|R} = \{\{x\}, \{y\}, \{x, y\}\}$, et $\widehat{\mathcal{C}}_{|R} = \{\{x, y\}\}$. Les sommets $\nu(x)$ et $\nu(y)$ sont étiquetés respectivement par "x" et "y", et le sommet $\nu(\{x, y\})$ est étiqueté par "xy".

Il s'agit alors de trouver un ensemble d'**arcs-attaches** depuis les sommets de la partie haute T vers les sommets de la partie basse B de telle manière que le réseau résultant représente l'ensemble $\mathcal{C}$ des clades fournis en entrée.

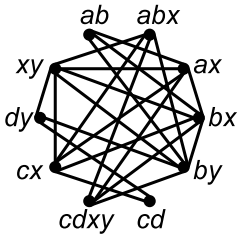
Plus précisément, notre objectif est de représenter tous les clades de $\mathcal{C}(e)$ par l'arc e de T, et tous les clades de $\mathcal{C}_{|R}$ par les arcs entrants des sommets de type $\nu(C)$, où $C \in \widehat{\mathcal{C}}_{|R}$. Pour assurer cela, les arcs-attaches doivent vérifier les propriétés suivantes :

(A1) Pour tout arc e de T et tout taxon $r \in R(e)$ il existe un arc-attache d'un descendant dans T depuis la cible de e soit vers $\nu(r)$, soit vers un sommet de la forme $\nu(C)$ dans B, où $C \in \widehat{\mathcal{C}}_{|R}$ contient r.

(A2) Pour tout sommet de type $\nu(C)$ dans B, où $C \in \widehat{\mathcal{C}}_{|R}$, il existe exactement un arc-attache depuis un sommet dans T vers $\nu(C)$.

(A3) Pour tout arc e de T et $r \in R$ tels que $\mathcal{C}(e)$ contient un clade $C \in \mathcal{C}$ qui ne contient pas r, il existe un chemin orienté depuis un sommet de T non-descendant de la cible de e, vers $\nu(r)$.

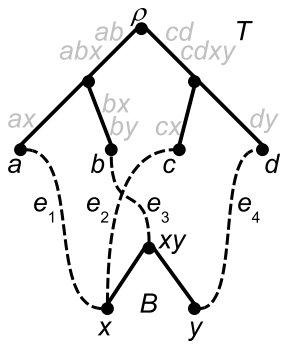
La propriété (A1) assure qu'on peut atteindre $\nu(r)$ par un chemin orienté depuis la cible de tout arc e de T qui représente un clade $C \in \mathcal{C}(e)$ qui contient r. Par exemple, dans la figure 2.18(c), le sommet étiqueté "a" est la cible d'un arc de T qui représente le clade $\{a, x\}$ qui contient x, et on peut effectivement atteindre $\nu(x)$ par un chemin orienté passant par l'arc-attache e_1, en partant de ce sommet étiqueté "a".



(a)

({b},{a},{x})
({a},{b},{x})
({a},{b},{y})
({a,x},{b},{y})
({a,b},{x},{c,d,y})
({a,b},{x},{c})
({a,b},{x},{y})
({a},{x},{b})
({a},{x},{c,d,y})
({a},{x},{c})
({a},{x},{y})
({x},{b},{y})
({b},{x},{c,d,y})
({b},{x},{c})
({b},{x},{y})
({b},{y},{c,d,x})
({b},{y},{d})
({b},{y},{x})
({d},{c},{x})
({c},{d},{y})
({c},{x},{y})
({d},{y},{x})

(b)



(c)

FIGURE 2.18 : Un ensemble de clades $\mathcal{C} = \big\{\{a,b\}, \{a,b,x\}, \{a,x\}, \{b,x\}, \{b,y\}, \{c,d\}, \{c,d,x,y\}, \{c,x\}, \{d,y\}, \{x,y\}\big\}$ sur $X = \{a,b,c,x,y\}$, le graphe d'incompatibilité $IG(\mathcal{C})$ (a), la liste correspondante L de déclarations d'incompatibilité (b), résolues en choisissant $\{x,y\}$ comme solution du problème MCS-r, et un réseau N à une couche de réticulation obtenu en résolvant le problème MINIMUM ATTACHMENT (c). Le réseau effectivement reconstruit par l'algorithme, où les feuilles ont degré entrant 1, est déduit directement à partir de N de manière triviale et montré en figure 2.21. Ici, la partie haute et la partie basse sont étiquetées respectivement par T et B. Les arcs de la partie haute sont étiquetés par les clades non-triviaux qu'ils représentent, et les feuilles par leurs taxons. Les arcs-attaches sont montrés en pointillés.

La propriété (A2) assure que tous les sommets de la forme $v(C)$ de B reçoivent un arc entrant. On ne leur autorise pas un degré entrant supérieur à 1 pour assurer que seuls les sommets de la forme $v(r)$, pour $r \in R$, seront des sommets hybrides, et que la solution sera bien un réseau à une couche de réticulation. Par exemple, dans la figure 2.18(c), le sommet $v(\{x,y\})$, étiqueté "xy", a un degré entrant égal à 1, grâce à l'arc-attache e_3.

Enfin, la propriété (A3) assure que tout clade ne contenant pas $v(r)$ puisse effectivement être contenu en tant que clade souple dans le réseau. Par exemple, dans la figure 2.18(c), si l'arc e_4 n'existait pas, alors le clade souple $\{b,x\}$ ne pourrait être représenté par l'arc dont la cible est le sommet étiqueté "b", qui représenterait uniquement les clades souples $\{b,y\}$ et $\{b,x,y\}$.

Ainsi nous cherchons à résoudre le problème suivant.

Problème 3 (MINIMUM ATTACHMENT)
Entrée : *un ensemble $\mathcal{C}$ de clades, une partie haute T et une partie basse B.*
Sortie : *un ensemble de taille minimale d'**arcs-attaches** depuis les sommets de T vers les sommets de B qui vérifie les propriétés (A1)–(A3).*

Pour montrer que ce problème est NP-complet, nous allons réduire le problème SET

COVER [12] restreint aux instances ne contenant pas d'inclusions d'ensembles, que nous appelons NO INCLUSION SET COVER.

Théorème 11 (NP-complétude de NO INCLUSION SET COVER) *Étant donnés une collection $\mathcal{C}$ d'ensembles, sans relation d'inclusion, sur un ensemble X, et un entier k, déterminer s'il existe un sous-ensemble de taille k de $\mathcal{C}$ qui couvre X est NP-complet.*

Démonstration. Nous procédons par une réduction de la version générale du problème SET COVER.

Étant donnée une instance de SET COVER $\mathcal{C} = \{C_1, \ldots, C_m\}$ sur un ensemble d'éléments $X = \{x_1, \ldots x_n\}$, construisons l'instance $\mathcal{C}'$ suivante de NO INCLUSION SET COVER, sur l'ensemble d'éléments $X' = X \cup \{x_{n+1}, \ldots, x_{n+m+1}\}$: $\mathcal{C}' = \left\{C'_i = C_i \cup \{x_{n+i}\}\right\} \cup \left\{C'_{m+1} = \{x_{n+1}, \ldots, x_{n+m+1}\}\right\}$. Ainsi, on a ajouté un nouvel élément x_{n+i} à chaque ensemble C_i, ainsi qu'un nouvel élément x_{n+m+1}, et un ensemble C'_{m+1} qui contient tous ces $m+1$ nouveaux éléments. Cette construction est illustrée en figure 2.19. Notons qu'aucun ensemble C'_i de $\mathcal{C}'$ n'est inclus dans un autre, car pour tout $i \in [1..m+1]$, x_i n'appartient qu'à l'ensemble C'_{n+i}.

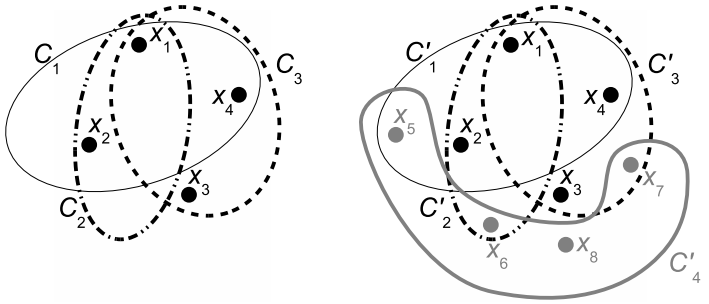


FIGURE 2.19 : Illustration de la réduction de SET COVER à NO INCLUSION SET COVER : une instance de SET COVER avec une solution de taille 2 montrée en gras (i), et l'instance de NO INCLUSION SET COVER correspondante avec une solution de taille 3 également montrée en gras (ii).

Supposons qu'il existe une solution $\{C_i \mid i \in I \subset [1..m]\}$ à k ensembles pour l'instance $\mathcal{C}$ de SET COVER, alors les k ensembles de $\{C'_i \mid i \in I\}$ couvrent également tous les sommets x_j pour $j \in [1..n+m]$, donc les $k+1$ ensembles de $\{C'_i \mid i \in I\} \cup \{C'_{m+1}\}$ couvrent tous les sommets de X'.

12. Problème SET COVER : étant donnée une collection $\mathcal{C}$ d'ensembles sur un ensemble X et un entier k, il s'agit de trouver un sous-ensemble $\mathcal{C}' \subseteq \mathcal{C}$ de taille k qui couvre X, c'est-à-dire tel que $\forall x \in X, \exists C \in \mathcal{C}'$ tel que $x \in C$.

Inversement, supposons qu'il existe une solution $\{C'_i \mid i \in I \subset [1..m+1]\}$ à $k+1$ ensembles pour l'instance $\mathcal{C}'$ de NO INCLUSION SET COVER. Parmi les ensembles de $\mathcal{C}'$, seul C'_{m+1} contient x_{n+m+1}, donc il appartient nécessairement à la solution. Les k ensembles restants C'_i de la solution couvrent nécessairement tous les sommets de $X' \setminus C'_{m+1}$, c'est-à-dire tous les sommets de X, donc les k ensembles de $\{C_i \mid i \in I \setminus \{m+1\}\}$ couvrent également tous les sommets de X.

Finalement, comme il est possible de vérifier en temps polynomial qu'une collection d'ensembles est bien solution, le problème NO INCLUSION SET COVER est NP-complet. □

Théorème 12 (NP-complétude de MINIMUM ATTACHMENT) *Étant donnés une instance du problème* MINIMUM ATTACHMENT *et un entier* k*, déterminer s'il existe une solution de ce problème à* k *arcs-attaches est NP-complet.*

Démonstration. Nous réduisons le problème NO INCLUSION SET COVER au problème MINIMUM ATTACHMENT.

Étant donnée une instance $\mathcal{C}$ de NO INCLUSION SET COVER à m ensembles, construisons une instance de MINIMUM ATTACHMENT qui a une solution avec $m+k$ arcs-attaches si et seulement si NO INCLUSION SET COVER a une solution de taille k, comme illustré en figure 2.20. On peut se restreindre aux instances $\mathcal{C}$ de NO INCLUSION SET COVER telles que tout élément de X est contenu dans au moins deux clades de $\mathcal{C}$ car on peut réduire le cas général à ce cas restreint (tout clade contenant un élément de X contenu par aucun autre clade est forcément dans la solution du problème).

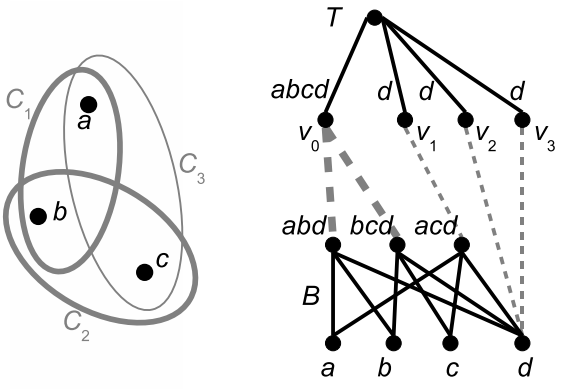
Soit d un nouveau taxon, n'appartenant pas à X. On va chercher à construire par le problème MINIMUM ATTACHMENT un réseau phylogénétique T de racine ρ avec $m+1$ feuilles.

L'entrée de ce problème sera constituée de la partie haute suivante : une racine incidente à $m+1$ arcs (ρ, v_i) pour $0 \leq i \leq m$ tels que $R((\rho, v_0)) = X \cup \{d\}$ et $R((\rho, v_i)) = \{d\}$. Sa partie basse est constituée d'un sommet $v(C \cup \{d\})$ pour tout ensemble $C \in \mathcal{C}$ ($\mathcal{C}$ ne contenant pas d'ensembles inclus l'un dans l'autre, nous sommes assurés que tous les éléments $C \cup \{d\}$ sont maximaux pour l'inclusion), d'un sommet hybride $v(r)$ pour tout $r \in X \cup \{d\}$, et d'un arc $(v(C \cup \{d\}), v(r))$ pour tout $r \in \mathcal{C}$, pour tout $C \in \mathcal{C}$.

Le fait qu'on se soit restreint à des instances de NO INCLUSION SET COVER où tout élément de X est présent dans au moins deux ensembles de $\mathcal{C}$ assure que chaque sommet $v(r)$ a deux parents, c'est-à-dire que la propriété (A3) est respectée pour les arcs (ρ, v_i) $(1 \leq i \leq m)$.

Nous allons maintenant prouver l'affirmation suivante : s'il existe une solution de cette instance de MINIMUM ATTACHMENT avec $m+k$ arcs-attaches, alors il en existe une dans laquelle le sommet v_0 est parent seulement de sommets de la forme $v(C \cup \{d\})$, avec $C \in \mathcal{C}$.

Il découle des propriétés (A1) et (A2) qu'il y a k arcs-attaches incidents à v_0 et à des sommets de B, et m arcs-attaches incidents aux sommets $v_1, \ldots, v_m$ et à des sommets de B.



(i) (ii)

FIGURE 2.20 : Illustration de la réduction de NO INCLUSION SET COVER à MINIMUM ATTACHMENT : une instance de NO INCLUSION SET COVER (i) et l'instance de MINIMUM ATTACHMENT correspondante (ii). Dans cette dernière, l'ajout des arcs-attaches en tirets gris permet de satisfaire les contraintes induites par les trois propriétés (A1), (A2) et (A3) qui s'expriment en particulier par les étiquettes $R(e)$ des arcs de la partie haute. Les étiquettes des sommets de la partie basse enfants de v_0 correspondent à la solution de NO INCLUSION SET COVER.

Comme tous les arcs-attaches partant de v_0 conduisent aux sommets-clades de B, alors cet ensemble de sommets clades définit un sous-ensemble $\mathcal{S}$ de clades qui couvre X', et donc également X, fournissant une solution du problème NO INCLUSION SET COVER de taille k. Maintenant, afin de prouver l'affirmation, supposons qu'un des arcs-attaches incidents à v_0 est également incident à un sommet de type $\nu(r)$ dans B, pour un taxon $r \in X'$. Si un sommet-clade $\nu(C \cup \{d\})$ est enfant de v_0 avec $x \in C \cup \{d\}$, alors on peut supprimer l'arc-attache de v_0 à $\nu(r)$ car il est superflu. Sinon, il existe un sommet-clade $\nu(C \cup \{d\})$ qui est enfant d'un des sommets v_i de T, avec $i > 0$. Dans ce cas, on modifie ainsi la solution : rediriger l'arc-attache de v_0 vers $\nu(r)$ de telle sorte qu'il soit incident à $\nu(C \cup \{d\})$, et rediriger l'arc-attache de v_i vers $\nu(C \cup \{d\})$ de telle sorte qu'il soit incident à $\nu(r)$. Cette opération est répétée jusqu'à ce que tous les enfants de v_0 soient des sommets-clades.

Inversement, il n'est pas difficile de voir que toute solution à k ensembles d'une instance du problème NO INCLUSION SET COVER mène à une solution à $m + k$ arcs-attaches de ce cas simplifié du problème MINIMUM ATTACHMENT. □

Les instances de ce problème que nous devons traiter en pratique sont généralement de taille réduite, et une approche de séparation et évaluation [Lawler et Wood, 1966] suffit pour les résoudre. A partir de la solution obtenue pour le problème MINIMUM ATTACHMENT, il suffit d'ajouter des arcs menant aux feuilles pour obtenir un réseau phylogénétique dont les feuilles ont degré entrant 1, comme le réseau de la figure 2.21.

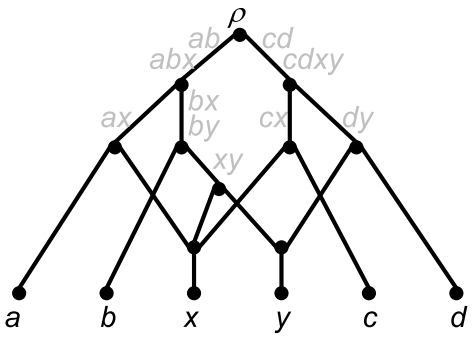


FIGURE 2.21 : Réseau phylogénétique obtenu à partir de la solution du problème MINIMUM ATTACHMENT sur les clades de l'exemple en figure 2.18. Les clades sont indiqués en gris sur les arcs qui les représentent.

Ajoutons qu'il est également possible de le formuler comme problème de programmation linéaire en nombres entiers, de la façon suivante. Pour tout sommet u de la partie haute et tout sommet v de la partie basse, la variable binaire $a_{u,v}$ indique s'il existe un arc-attache de u à v (1 si oui, 0 sinon). Les propriétés (A1)-(A3) se traduisent par les inégalités suivantes :

– (A1) : pour tout arc $e = (x, y)$ de T, pour tout taxon $r \in R(e)$,

$$\sum_{u \in T \,|\, u \preceq_T y} a_{u,v(r)} + \sum_{u \in T,\, v(C) \in B \,|\, u \preceq_T y,\, r \in C} a_{u,v(C)} \geq 1$$

– (A2) : pour tout $C \in \hat{\mathcal{C}}_{|R}$, $\sum_{u \in T} a_{u,v(C)} = 1$

– (A3) : pour tout arc $e = (x, y)$ de T et pour tout $r \in R$ tel que $\mathcal{C}(e)$ contient un clade $C \in \mathcal{C}$ qui ne contient pas r,

$$\sum_{u \in T \,|\, u \not\preceq_T y} \sum_{v \in B \,|\, v(r) \preceq_B v} a_{u,v} = \sum_{u \in T \,|\, u \not\preceq_T y} a_{u,v(r)} + \sum_{u \in T,\, C' \in \hat{\mathcal{C}}_{|R} \,|\, u \not\preceq_T y,\, r \in C'} a_{u,v(C')} \geq 1.$$

Il s'agit alors de minimiser la somme $\sum_{u \in T, v \in B} a_{u,v}$ en respectant ces contraintes.

Deuxième partie

Utilisation pratique des méthodes combinatoires

Préambule

Les méthodes combinatoires proposées dans la partie I fonctionnent sur des données considérées comme exactes, et demandent parfois une certaine exhaustivité. De plus, leur temps de calcul dépend de la complexité des réseaux reconstruits.

Nous abordons dans cette partie II les contraintes biologiques sur les données, et comment nous pouvons les prendre en compte pour utiliser, et en utilisant, des méthodes combinatoires. En évoquant les limites que constituent le bruit et le silence dans les données biologiques disponibles, nous proposons une solution pour corriger des triplets erronés, et présentons les méthodes de clôture qui permettent d'inférer des données manquantes.

Enfin, nous montrons plus concrètement comment utiliser les algorithmes de reconstruction combinatoire de réseaux phylogénétiques, avec une sélection appropriée des données, et de la méthode de reconstruction. Nous appliquons alors diverses méthodes combinatoires pour créer des réseaux phylogénétiques, visualisés avec le logiciel Dendroscope, à partir de données de la base Hogenom [Dufayard et al.*, 2005]. Cette base de données d'arbres de gènes est accessible librement sur internet et concerne 513 espèces et contient plus de 70 000 arbres de gènes.*

3 Limites des méthodes combinatoires

Les arbres de gènes disponibles dans les bases de données de phylomes, comme de nombreuses données biologiques, peuvent contenir des erreurs de topologie des arbres, qui se traduisent en erreurs sur les quadruplets, triplets ou clades que nous utilisons en entrée des algorithmes de reconstruction de réseaux phylogénétiques. Les données sont donc **bruitées**. Il est possible aussi que certaines données soient manquantes, si un gène n'a pas été identifié chez une des espèces, par exemple. Il faut donc aussi s'attaquer au problème du **silence** des données. Nous proposons de résoudre ces deux problèmes à l'aide de méthodes combinatoires, développées dans le cadre de cette thèse ou déjà connues. Nous évoquons aussi d'autres limites des méthodes combinatoires de reconstruction en donnant des éléments concrets sur l'explosion de complexité en fonction du niveau, et en identifiant certains problèmes de fiabilité.

Dans tout ce chapitre, nous nous limitons aux réseaux les plus informatifs possibles, c'est-à-dire les réseaux binaires.

3.1 Bruit et silence dans les données

3.1.1 Bruit et corrections d'erreurs sur les triplets

a) Méthodes existantes

Une façon de prendre en compte les possibilités d'erreurs dans les données en entrée est de construire un résultat qui ne contient pas absolument tous les triplets en entrée mais un nombre maximal d'entre eux.

Toutefois la complexité de ce problème est importante même lorsque l'on cherche à reconstruire des arbres. En effet, le problème MAXIMUM COMPATIBLE SUBSET OF ROOTED TRIPLES, qui consiste à déterminer s'il existe un arbre qui contient r triplets parmi un ensemble $\mathcal{R}$ fourni en entrée, est NP-complet [Bryant, 1997; Jansson, 2001; Wu, 2004]. Ce résultat est renforcé et généralisé par van Iersel *et al.* [2009b], qui prouvent que le problème suivant est également NP-complet : déterminer, pour $k \geq 0$, s'il existe un réseau de niveau k qui contient r triplets parmi un ensemble dense $\mathcal{R}$ fourni en entrée.

Des algorithmes exponentiels ont toutefois été proposés pour résoudre ces problèmes, en temps $O((|\mathcal{R}|+n^2)3^n)$ [Wu, 2004] pour celui sur les arbres. Un premier algorithme exponentiel de programmation dynamique pour reconstruire un réseau de niveau 1 contenant le maximum de triplets de $\mathcal{R}$ a été proposé, en temps $O(|\mathcal{R}|4^n)$ et en espace $(n3^n)$ [van

Iersel *et al.*, 2009b]. Une heuristique pour ce problème a été implémentée dans le logiciel Lev1athan [Huber *et al.*, 2010].

Enfin, dans le contexte non enraciné, citons une approche similaire à celle que nous développons dans cette section : un algorithme de complexité paramétrée en $O(4^k n + n^4)$ [Gramm et Niedermeier, 2003], permet de reconstruire, s'il existe, un arbre phylogénétique binaire non enraciné contenant $|Q - k|$ quadruplets parmi un ensemble minimalement dense Q. Cet algorithme, qui a ensuite été amélioré pour obtenir des complexités en temps en $O(3.0446^k n + n^4)$, $O(2.0162^k n^3 + n^5)$, et même $O^*((1+\epsilon)^k)$ pour toute constante $\epsilon > 0$ [Chang *et al.*, 2010], se fonde sur des obstructions initialement trouvées par Bandelt et Dress [1986], c'est-à-dire de sous-ensembles de quadruplets dont la présence dans un ensemble dense Q de quadruplets empêche de reconstruire un arbre contenant Q. Inversement, à partir de tout ensemble de quadruplets ne contenant aucune obstruction, il est possible de reconstruire un arbre. Précisons également que de nombreuses approches autres que les algorithmes de complexité paramétrée existent pour ce problème. En particulier, s'il est possible d'obtenir un arbre contenant l'ensemble minimalement dense de quadruplets en entrée après suppression de $O(n)$ d'entre eux, alors le problème se résout en temps polynomial [Berry *et al.*, 1999; Wu *et al.*, 2006].

Notre approche dans cette section consiste à commencer par déterminer un ensemble explicite d'obstructions de triplets pour la reconstruction d'arbres. Nous en déduisons un algorithme de complexité paramétrée pour la reconstruction d'un arbre en supprimant un nombre minimal de triplets parmi un ensemble dense fourni en entrée, permettant ainsi de corriger un éventuel bruit dans les données. Puis nous tentons de généraliser cette approche au niveau 1, en obtenant quelques premiers résultats qui seront utilisés en section 3.3.

b) Obstructions sur les triplets pour reconstruire un arbre

L'idée de base de notre approche de reconstruction d'arbres à partir d'un ensemble dense de triplets, et de découverte d'obstructions si c'est impossible, est de savoir où placer une nouvelle feuille x en considérant un triplet $a|bc$ qui a été placé.

Définition 3.1 *Soit un triplet* $a|bc$*, une feuille* x*, et un ensemble dense de triplets* $\mathcal{R}$*. Nous définissons les cinq zones et feuilles suivantes (voir figure 3.1) en fonction de* $\mathcal{R}_x = \mathcal{R}_{|\{a,b,c,x\}}$ *:*

- *zone A :* $L_A = \{x \mid \mathcal{R}_x = \{a|bc, b|ax, c|ax, x|bc\}\}$*, feuille* $\alpha_A = a$*,*
- *zone B :* $L_B = \{x \mid \mathcal{R}_x = \{a|bc, a|bx, a|cx, c|bx\}\}$*, feuille* $\alpha_B = b$*,*
- *zone C :* $L_C = \{x \mid \mathcal{R}_x = \{a|bc, a|bx, a|cx, b|cx\}\}$*, feuille* $\alpha_C = c$*,*
- *zone D :* $L_D = \{x \mid \mathcal{R}_x = \{x|ab, x|ac, x|bc, a|bc\}\}$*, feuille* $\alpha_D = b$*,*
- *zone E :* $L_E = \{x \mid \mathcal{R}_x = \{a|bc, a|bx, a|cx, x|bc\}\}$*, feuille* $\alpha_E = b$*,*

et $\mathcal{R}_X = \mathcal{R}_{|L_X \cup \{\alpha_X\}}$ *pour* $X \in \{A, B, C, D, E\}$.

Théorème 13 *Pour un ensemble dense de triplets* $\mathcal{R}$*, les propriétés suivantes sont équivalentes :*

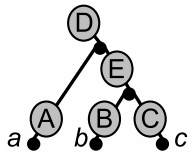


FIGURE 3.1 : Zones définies par les différents cas possibles pour les ensembles de triplets sur quatre feuilles.

(i) $\mathcal{R}$ *est contenu dans un arbre.*

(ii) $\mathcal{R}$ *contient exactement un triplet sur chaque ensemble de trois feuilles, et pour tout ensemble de triplets sur quatre feuilles* $\{a,b,c,d\}$*, d'une part* $ab|d, bc|d \in \mathcal{R} \Rightarrow ac|d$*, et d'autre part* $ab|c, bc|d \in \mathcal{R} \Rightarrow ac|d$ *[Dress, 1997],*

(iii) $\mathcal{R}$ *contient exactement un triplet sur chaque ensemble de trois feuilles, et pour tout ensemble de triplets sur quatre feuilles* $\{a,b,c,d\}$*,* $ab|c, bc|d \in \mathcal{R} \Rightarrow ab|d, ac|d \in \mathcal{R}$ *[Guillemot et Berry, 2010],*

(iv) $\mathcal{R}$ *contient exactement un triplet sur chaque ensemble de trois feuilles, et tout ensemble de triplet sur quatre feuilles est isomorphe soit à* $\{x_1|x_2x_3, x_1|x_2x_4, x_1|x_3x_4, x_2|x_3x_4\}$ *(cas 1), soit à* $\{x_1|x_2x_3, x_2|x_1x_4, x_3|x_1x_4, x_4|x_2x_3\}$ *(cas 2),*

(v) $\mathcal{R}$ *ne contient aucun ensemble de triplets isomorphe à ces quatre obstructions :* ① $\{a|bc, c|ab\}$, ② $\{a|bc, c|bd, d|ab\}$, ③ $\{a|bc, c|bd, d|ac\}$, ④ $\{a|bc, a|bd, d|ac\}$.

Démonstration. $i \Leftrightarrow ii$ et $i \Leftrightarrow iii$ ont été prouvées de façon indépendante par Dress [1997] et Guillemot et Berry [2010]. Ces deux équivalences pourraient être utilisées pour prouver que $i \Leftrightarrow iv$. Toutefois, nous donnons une preuve directe dont certaines idées sont reprises à la fin de cette section pour fournir des résultats préliminaires pour le niveau 1.

$i \Rightarrow iv$: il n'y a que deux formes possibles d'arbres binaires enracinés sur quatre feuilles : la chenille correspond au cas 1 et l'arbre équilibré au cas 2. Aucun n'autorise l'existence de deux triplets distincts sur un même ensemble de trois feuilles.

$iv \Rightarrow i$: procédons par récurrence sur le nombre de feuilles. Le résultat est direct sur quatre feuilles. Considérons un ensemble de triplets $\mathcal{T}$ sur $n > 4$ feuilles. Commençons avec un triplet $a|bc$. Pour toute autre feuille x, nous n'avons que les cas 1 et 2 pour les ensembles non isomorphes de triplets sur quatre feuilles. Par l'ensemble des bijections possibles $f : \{x_1, x_2, x_3, x_4\} \to \{a, b, c, x\}$, on obtient cinq possibilités en fonction de la valeur de $f^{-1}(x)$, ce qui correspond exactement aux cinq zones de la définition 3.1.

Ainsi, à toute feuille x, on peut associer une zone X. Les ensembles $\mathcal{R}_X$ sont denses, et leur taille est strictement plus petite que celle de l'ensemble original de triplets, donc on peut appliquer l'hypothèse de récurrence et s'assurer que chacun d'entre eux correspond à un arbre qui peut être intégré dans la structure de la figure 3.1 pour obtenir T. Il faut

également vérifier que tous les triplets qui ne sont pas dans $\mathcal{R}_A \cup \mathcal{R}_B \cup \mathcal{R}_C \cup \mathcal{R}_D \cup \mathcal{R}_E$ sont corrects, c'est-à-dire contenus dans T.

Feuilles dans des zones différentes :
Dans la suite de la preuve, quand nous utilisons la lettre capitale X, elle représente n'importe quelle feuille de L_X. Les triplets avec des feuilles dans des zones différentes peuvent être de différents types :

- sur les feuilles $\{x, y, Z\}$, où $\{x, y\} \in \{a, b, c\}$, $Z \in \{A, B, C, D, E\}$. Par définition des zones, ces triplets sont corrects.
- sur les feuilles $\{x, Y, Z\}$, où :
  - $x = a$ et :
    - $Y = A$ et :
      - $Z = B$: on sait qu'on a les triplets $b|aA$ et $a|bB$, donc on doit être dans le cas 2, donc on a aussi le triplet $B|aA$, qui est correct, et $A|bB$.
      - $Z = C$: symétrique au cas précédent ($C \leftrightarrow B, c \leftrightarrow b$).
      - $Z = D$: on a les triplets $b|aA$ et $D|ab$, donc on doit être dans le cas 1, et on a aussi le triplet $D|aA$, qui est correct, et $D|Ab$.
      - $Z = E$: on a les triplets $b|aA$ et $a|bE$, donc on doit être dans le cas 2, et on a aussi $E|aA$, qui est correct, et $A|bE$.
    - $Y = B$ et :
      - $Z = C$: on a $a|bB$ et $a|bC$, donc on est dans le cas 1, et on a aussi $a|BC$, qui est correct.
      - $Z = D$: on a $a|bB$ et $D|ab$, donc on est dans le cas 1, et on a aussi $D|aB$, qui est correct, et $D|bB$.
      - $Z = E$: on a $a|bB$ et $a|bE$, donc on est dans le cas 1, et on a aussi $a|BE$, qui est correct.
    - $Y = C$: symétrique au cas $Y = B$ ($C \leftrightarrow B, c \leftrightarrow b$).
    - $Y = D$ et $Z = E$: on a $D|ab$ et $a|bE$, donc on est dans le cas 1, et on a aussi $D|aE$, qui est correct, et $D|bE$.
  - $x = b$ et :
    - $Y = A$ et :
      - $Z = B$: on a déjà montré qu'on a $A|bB$, qui est correct.
      - $Z = C$: on sait qu'on a $b|aA$ et $a|bC$, donc on est dans le cas 2, donc on a aussi $A|bC$, qui est correct.
      - $Z = D$: on a déjà montré qu'on a $D|Ab$, qui est correct.
      - $Z = E$: on a déjà montré qu'on a $A|bE$, qui est correct.
    - $Y = B$ et :
      - $Z = C$: on sait qu'on a les triplets $b|cC$ et $c|bB$, donc on est dans le cas 2, donc on a aussi $C|bB$, qui est correct, et $B|cC$.
      - $Z = D$: on a déjà montré qu'on a $D|bB$, qui est correct.

- $Z = E$: on a $c|bB$ et $E|bc$, donc on est dans le cas 1, donc on a aussi $E|bB$, qui est correct, et $E|Bc$.
- $Y = C$ et :
  - $Z = D$: on a $a|bC$ et $D|ab$, donc on est dans le cas 2, donc on a aussi $D|bC$, qui est correct.
  - $Z = E$: on a $a|bC$ et $a|bE$, donc on est dans le cas 1, donc on a aussi $a|EC$, qui est correct.
- $Y = D$ et $Z = E$: on a déjà montré qu'on a $D|bE$, qui est correct.

– $x = c$: symétrique au cas précédent où $x = b$ ($c \leftrightarrow b, C \leftrightarrow B$).

– sur les feuilles $\{X, Y, Z\}$, où :
- $X = A$ et :
  - $Y = B$ et :
    - $Z = C$: on a montré qu'on a les triplets $a|BC$, $B|aA$ et $C|aA$, donc on est dans le cas 2 et on a aussi le triplet $A|BC$, qui est correct.
    - $Z = D$: on a montré qu'on a $A|Bc$, $D|Ac$ et $D|Bc$, donc on est dans le cas 1 et on a aussi $D|AB$, qui est correct.
    - $Z = E$: on a montré qu'on a $B|aA$, $E|aA$ et $a|BE$, donc on est dans le cas 2 et on a aussi $A|BE$, qui est correct.
  - $Y = C$: symétrique au cas précédent où $Y = B$ ($c \leftrightarrow b, C \leftrightarrow B$).
  - $Y = D$ et $Z = E$: on a montré qu'on a $D|aA$, $D|aE$ et $E|aA$, donc on est dans le cas 1 et on a aussi $D|AE$, qui est correct.
- $X = B$ et :
  - $Y = C$ et :
    - $Z = D$: on a montré qu'on a $D|Bb$, $D|Cb$ et $B|Cb$, donc on est dans le cas 1 et on a aussi $D|BC$, qui est correct.
    - $Z = E$: symétrique au cas précédent où $Z = D$ ($E \leftrightarrow D$).
  - $Y = D$ et $Z = E$: on a montré qu'on a $D|aA$, $D|aE$ et $E|aA$, donc on est dans le cas 1 et on a aussi $D|AE$, qui est correct.
- $X = C$: symétrique au cas précédent où $X = B$ ($c \leftrightarrow b, C \leftrightarrow B$).

Deux feuilles dans la même zone :
On considère les triplets sur deux feuilles dans la zone X et une dans une zone distincte Y. Une remarque importante est que les ensembles $\mathcal{R}_X$ ont été choisis de telle sorte que les triplets qui ne sont pas dans $\mathcal{R}_X$ ne donnent pas d'autre information sur la position des feuilles dans X que celle donnée par $\mathcal{R}_X$.

Deux cas apparaissent :

– soit la zone Y est **sous** la zone X (i.e. il existe un chemin orienté de la racine vers la zone X qui passe à travers la zone Y). Alors les feuilles de Y peuvent influencer la localisation de feuilles sous X. Nous montrons ci-dessous que dans ce cas cette information est cohérente pour toutes les feuilles de Y, et cohérente avec l'information donnée par la feuille α_X.

– sinon (si X est sous Y ou bien qu'aucun ne peut être atteint par un chemin orienté passant à travers l'autre, depuis la racine), nous montrons ci-dessous que les triplets sont contenus dans l'arbre reconstruit.

Cas où Y est sous X : prouvons que pour toutes feuilles x_1 et x_2 dans la zone X, et y_1 dans la zone Y_1 en dessous, le triplet sur $\{x_1, x_2, y_1\}$ est cohérent avec le triplet sur $\{x_1, x_2, \alpha_X\}$ (i.e. $x_1|x_2y_1 \in \mathcal{R} \Rightarrow x_1|x_2\alpha_X \in \mathcal{R}$, $x_2|x_1y_1 \in \mathcal{R} \Rightarrow x_2|x_1\alpha_X \in \mathcal{R}$, et $y_1|x_1x_2 \in \mathcal{R} \Rightarrow \alpha_X|x_1x_2 \in \mathcal{R}$).

Commençons avec la zone E, en rappelant que $\alpha_E = b$. Considérons deux feuilles $e_1, e_2 \in L_E$. Dans la première partie de la démonstration (*feuilles dans des zones différentes*), on a prouvé qu'on a toujours les triplets $e_1|bx$ et $e_2|bx$, pour $x \in B \cup C \cup \{c\}$ (B et C sont les deux zones plus basses que E). Le triplet sur les feuilles $\{b, e_1, e_2\}$ peut être :

– soit $b|e_1e_2$: alors on doit être dans le cas 1 pour l'ensemble de triplets sur les feuilles $\{b, e_1, e_2, x\}$, donc on a le triplet $x|e_1e_2$,
– soit $e_1|be_2$: alors on doit être dans le cas 2 pour l'ensemble de triplets sur les feuilles $\{b, e_1, e_2, x\}$, et on a le triplet $e_1|xe_2$,
– soit $e_2|be_1$, ce qui est symétrique au cas précédent et force la présence du triplet $e_2|xe_1$.

Dans tous les cas, le triplet sur les feuilles $\{b, e_1, e_2\}$ est cohérent avec celui sur les feuilles $\{x, e_1, e_2\}$.

Les mêmes arguments tiennent quand on considère les feuilles $d_1, d_2 \in L_D$ et une feuille $x \in A \cup B \cup C \cup E \cup \{a, c\}$, ce qui termine la démonstration.

Cas où Y n'est pas sous X :

– Si $X = A$, les feuilles sont appelées a_1, a_2 : si $Y = B$, alors on a prouvé dans la première partie de la démonstration qu'on avait $B|aa_1$ et $B|aa_2$ donc on est dans le cas 1 et on a $B|a_1a_2$. Les mêmes arguments fonctionnent pour $Y \in \{C, D, E, b, c\}$.
– Si $X = B$, les feuilles sont appelées b_1, b_2 : si $Y = A$, alors on a prouvé dans la première partie de la démonstration qu'on avait $A|bb_1$ et $A|bb_2$ donc on est dans le cas 1 et on a $A|b_1b_2$. Les même arguments fonctionnent pour $Y \in \{C, D, E, b, c\}$.
– Si $X = C$, le cas est symétrique à celui où $X = B$.
– Si $X = D$, aucune zone n'est sous D.
– Si $X = E$, les feuilles sont appelées e_1, e_2. Si $Y = A$, on a démontré que $A|be_1$ et $A|be_2$ sont présent, donc on est dans le cas 1, et on a $A|e_1e_2$.

Trois feuilles dans la même zone :
En utilisant l'hypothèse de récurrence, on sait que le triplet est correct.

Finalement, on a montré que tous les triplets de $\mathcal{R}$ sont contenus dans T.

$iii \Rightarrow iv$: tout ensemble de triplets sur quatre feuilles est d'un certain type (cas 1 ou 2), qui ne contient aucune des obstructions ② to ④, et l'obstruction ① est explicitement interdite en forçant la présence d'exactement un triplet sur chaque ensemble de trois feuilles.

$\mathrm{iv} \Rightarrow \mathrm{iii}$: on considère tout ensemble de triplets sur quatre feuilles. À cause de l'obstruction ①, il ne contient pas deux triplets différents sur un même ensemble de feuilles.

On énumère donc tous les ensembles possibles de 4 triplets sur 4 feuilles, c'est-à-dire, à isomorphisme près :

- $\mathcal{R}_1 : \{x_1|x_2x_3, x_1|x_2x_4, x_1|x_3x_4, x_2|x_3x_4\}$
- $\mathcal{R}_2 : \{x_1|x_2x_3, x_2|x_1x_4, x_3|x_1x_4, x_4|x_2x_3\}$
- $\{x_1|x_2x_3, x_2|x_3x_4, x_3|x_1x_4, x_4|x_1x_2\}$: impossible car on peut trouver l'obstruction ③ dans cet ensemble de triplets ($x_1 \to d, x_2 \to a, x_3 \to c, x_4 \to b$)
- $\{x_1|x_2x_3, x_1|x_3x_4, x_4|x_1x_2, x_4|x_2x_3\}$: impossible car on peut trouver l'obstruction ④ dans cet ensemble de triplets ($x_1 \to a, x_2 \to c, x_3 \to b, x_4 \to d$)
- $\{x_1|x_2x_3, x_1|x_2x_4, x_3|x_1x_4, x_2|x_3x_4\}$: impossible car on peut trouver l'obstruction ④ dans cet ensemble de triplets ($x_1 \to a, x_2 \to b, x_3 \to d, x_4 \to c$)
- $\{x_1|x_2x_3, x_1|x_2x_4, x_3|x_1x_4, x_4|x_2x_3\}$: impossible car on peut trouver l'obstruction ④ dans cet ensemble de triplets ($x_1 \to a, x_2 \to b, x_3 \to d, x_4 \to c$)

Finalement, tous les ensembles de triplets de taille 4 sont de type $\mathcal{R}_1$ ou $\mathcal{R}_2$, qui sont exactement les cas 1 et 2. □

La liste des obstructions proposée ici a été améliorée par la proposition 1 de Guillemot et Mnich [2010]. En effet, l'obstruction ② n'est pas nécessaire car si l'on complète par le triplet manquant pour avoir un ensemble dense, on obtient forcément une des autres obstructions.

c) Algorithmes de correction d'erreurs

Ces obstructions induisent directement, dans le cas dense, un algorithme de complexité paramétrée en le nombre de triplets, appelé *MaxTripletSubset*, pour le problème MAXIMUM COMPATIBLE SUBSET OF ROOTED TRIPLES, dont nous rappelons plus formellement la définition ci-dessous [Bryant, 1997].

Problème 4 (MAXIMUM COMPATIBLE SUBSET OF ROOTED TRIPLETS (MCSRT))
Entrée : *Un ensemble $\mathcal{R}$ de triplets, et $t \in [0, |\mathcal{R}|]$.*
Sortie : *OUI s'il existe un ensemble $\mathcal{R}'$ de $\mathcal{R}$ tel que $\mathcal{R}'$ peut être contenu dans un arbre et $|\mathcal{R}'| \geq |\mathcal{R}| - t$, NON sinon.*

Ce problème peut être vu comme un problème de minimisation, où l'on veut éditer au plus t triplets pour obtenir un ensemble qui peut être contenu dans un arbre. Notons que contrairement à l'algorithme existant pour le problème similaire pour les quadruplets [Gramm et Niedermeier, 2003], qui se restreint aux ensembles minimalement denses de quadruplets, celui que nous présentons maintenant s'applique pour tout ensemble dense de triplets, on autorise donc l'ensemble de triplets fournis en entrée à contenir plus d'un triplet sur un ensemble de trois feuilles.

Théorème 14 *Pour un ensemble dense de triplets, le problème MCSRT peut être résolu avec une complexité en temps en* $O(6^t n + n^4)$.

Démonstration. L'algorithme *MaxTripletSubset* procède avec un arbre de recherche bornée : initialement, aucun triplet n'est marqué, et on trouve l'ensemble S_O des obstructions en temps $O(n^4)$.

Pour chaque sommet de l'arbre de recherche de profondeur au plus t :
- on considère la première obstruction dans S_O, il y a alors 3 triplets possibles à changer (les non marqués, car il est inutile de considérer les triplets déjà changés)
- pour chacun d'entre eux, on essaie les deux autres possibilités, donc 6 branches sont créées dans l'arbre de recherche.
- pour toute branche créée, on met à jour l'ensemble S_O des obstructions en temps $O(n)$ en considérant, pour le triplet édité $a|bc$, si une obstruction a été créée ou enlevée sur les feuilles $\{a, b, c, x\}$ pour toute feuille x.

Finalement, s'il existe un sommet de profondeur au plus t tel que $S_O = \emptyset$, alors il existe un arbre qui contient au moins $|\mathcal{R}| - t$ triplets de $\mathcal{R}$. Sinon, l'instance n'admet aucune solution. L'arbre de recherche a $O(6^t)$ sommets, donc la complexité en temps totale de l'algorithme est $O(6^t n + n^4)$. □

La complexité de cet algorithme a été améliorée par Guillemot et Mnich [2010] pour obtenir un complexité en temps de $2^{O(p^{1/3}\log p)} + O(n^4)$. Ces obstructions ou conflits de taille bornée sont aussi à l'origine des algorithmes mentionnés dans la section 2.1.2, qui permettent de trouver un arbre contenant un ensemble de triplets, après suppression d'un nombre minimal de feuilles.

d) Vers les obstructions du niveau 1

Nous proposons une approche similaire pour généraliser ce résultat au niveau 1. Même si elle n'a pas encore abouti, elle débouche sur de premiers résultats intéressants que nous donnerons à la fin de ce chapitre.

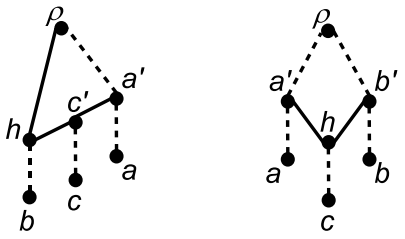
De la même manière que pour les arbres, où un triplet de référence $a|bc$ permettait de fixer une configuration locale et d'étudier le placement des autres feuilles dans cette configuration, nous allons également partir d'un triplet de référence, en exigeant toutefois de trouver deux triplets sur le même ensemble de feuilles, ce qui s'explique par le lemme suivant.

Lemme 8 *Soit* $\mathcal{R}$ *un ensemble dense de triplets sur un ensemble* X. *S'il existe un réseau* N *strictement de niveau 1 tel que* $\mathcal{R} = \mathcal{R}(N)$, *alors* X *contient trois feuilles* a, b *et* c *telles que* $\{a|bc, b|ac\} \subseteq \mathcal{R}$, *et* $c|ab \notin \mathcal{R}$.

Démonstration. Comme N est strictement de niveau 1, alors il contient au moins un blob dont le graphe non orienté sous-jacent est un cycle. Appelons h le sommet hybride contenu dans ce blob.

Si comme dans la figure 3.2(i), un des parents de h est la racine ρ du blob, alors comme ce blob a au moins quatre sommets, h a au moins un parent c' et un grand-parent a' distincts de ρ. Soit b une feuille de N descendante de h, c une feuille descendante de c' mais pas de h, et a' une feuille descendante de a mais pas de c'. Alors $a|bc$ et $b|ac$ sont contenus dans N, mais $c|ab$ ne l'est pas.

Sinon, comme dans la figure 3.2(ii), aucun parent de h n'est la racine ρ du blob, alors appelons a' et b' ses parents. Soit c une feuille descendante de h, a une feuille descendante de a' mais pas de h, et b une feuille descendante de b' mais pas de h. Alors $a|bc$ et $b|ac$ sont contenus dans N, mais $c|ab$ ne l'est pas.



(i) (ii)

FIGURE 3.2 : Configurations possibles pour un sommet hybride dans un réseau de niveau 1. Les arcs en pointillés indiquent qu'il existe un chemin allant de la source vers la cible.

Ainsi, dans tous les cas, on a trouvé trois feuilles vérifiant les propriétés voulues. □

De façon similaire à la définition 3.1, étant donné un ensemble de feuilles sur lequel deux triplets sont présents dans $\mathcal{R}$, on définit huit zones pour chacune des configurations possibles pour le niveau 1, qui correspondent à toutes les manières d'ajouter une feuille x dans un réseau dont l'ensemble des triplets est $\{a|bc, b|ac\}$.

Définition 3.2 *Considérons un ensemble extrêmement dense de triplets $\mathcal{R}$ sur un ensemble X de feuilles, et quatre feuilles $a, b, c, x \in X$, telles que $\{a|bc, b|ac\} \subset \mathcal{R}$ et $c|ab \notin \mathcal{R}$. On définit les zones suivantes (voir figure 3.3) en fonction de $\mathcal{R}_x = \mathcal{R}_{|\{a,b,c,x\}}$:*

- *zone A :* $L_A = \{x \mid \mathcal{R}_x = \{a|bc, b|ac, b|ax, b|cx, c|ax, x|bc\}\}$, $\alpha_A = a$,
- *zone B :* $L_B = \{x \mid \mathcal{R}_x = \{a|bc, a|bx, a|cx, b|ac, c|bx, x|ac\}\}$, $\alpha_B = b$,
- *zone C :* $L_C = \{x \mid \mathcal{R}_x = \{a|bc, a|bx, a|cx, b|ac, b|ax, b|cx\}\}$, $\alpha_C = c$,
- *zone H :* $L_H = \{x \mid \mathcal{R}_x = \{a|bc, c|ac, x|ab, x|ac, x|bc\}\}$, $\alpha_H = c$,
- *zone* D1 *:* $L_{D1} = \{x \mid \mathcal{R}_x = \{a|bc, b|ac, a|bx, a|cx, x|ac, b|cx, c|bx\}\}$, $\alpha_{D1} = c$,
- *zone* E1 *:* $L_{E1} = \{x \mid \mathcal{R}_x = \{a|bc, b|ac, a|bx, a|cx, x|ac, c|bx, x|bc\}\}$, $\alpha_{E1} = c$,

- *zone* F1 : $L_{F1} = \{x \mid \mathcal{R}_x = \{a|bc, b|ac, b|ax, a|cx, c|ax, b|cx, x|bc\}\}$, $\alpha_{F1} = c$,
- *zone* G1 : $L_{G1} = \{x \mid \mathcal{R}_x = \{a|bc, b|ac, b|ax, c|ax, x|ac, b|cx, x|bc\}\}$, $\alpha_{G1} = c$,
- *zone* D2 : $L_{D2} = \{x \mid \mathcal{R}_x = \{a|bc, b|ac, a|bx, b|ax, a|cx, b|cx, c|bx\}\}$, $\alpha_{D2} = b$,
- *zone* E2 : $L_{E2} = \{x \mid \mathcal{R}_x = \{a|bc, b|ac, a|bx, x|ab, x|ac, c|bx, x|bc\}\}$, $\alpha_{E2} = b$,
- *zone* F2 : $L_{F2} = \{x \mid \mathcal{R}_x = \{a|bc, b|ac, a|bx, b|ax, a|cx, b|cx, x|bc\}\}$, $\alpha_{F2} = b$,
- *zone* G2 : $L_{G2} = \{x \mid \mathcal{R}_x = \{a|bc, b|ac, b|ax, x|ab, x|ac, b|cx, x|bc\}\}$, $\alpha_{G2} = b$,
- *zone* D3 : $L_{D3} = \{x \mid \mathcal{R}_x = \{a|bc, b|ac, a|bx, b|ax, a|cx, c|ax, b|cx\}\}$, $\alpha_{D3} = a$,
- *zone* E3 : $L_{E3} = \{x \mid \mathcal{R}_x = \{a|bc, b|ac, b|ax, c|ax, x|ab, x|ac, x|bc\}\}$, $\alpha_{E3} = a$,
- *zone* F3 : $L_{F3} = \{x \mid \mathcal{R}_x = \{a|bc, b|ac, a|bx, b|ax, a|cx, x|ac, b|cx\}\}$, $\alpha_{F3} = a$,
- *zone* G3 : $L_{G3} = \{x \mid \mathcal{R}_x = \{a|bc, b|ac, a|bx, x|ab, a|cx, x|ac, x|bc\}\}$, $\alpha_{G3} = a$.

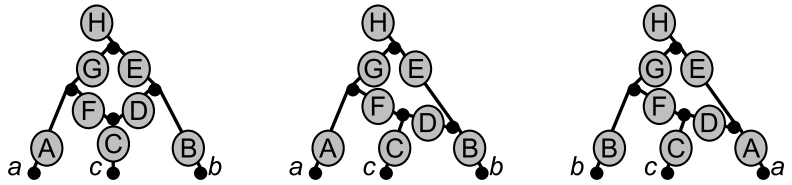


FIGURE 3.3 : Les huit zones définies par les différents cas possibles pour les ensembles de triplets sur quatre feuilles, pour chacune des trois configurations possibles au niveau 1.

Cette définition fournit une piste pour l'extension du théorème 13. De plus, nous l'utilisons dans la preuve du lemme 9 de la page 122.

3.1.2 Silence et inférence des données manquantes

Comme nous l'avons vu dans le chapitre 2, de nombreuses méthodes combinatoires de reconstruction de réseaux phylogénétiques exigent des données assez complètes. Les ensembles de triplets doivent être denses pour obtenir des algorithmes polynomiaux, et quand on reconstruit des réseaux à partir de clades, ils doivent être obtenus à partir d'arbres concernant les mêmes espèces.

Ce n'est pas le cas des arbres phylogénétiques des bases de données de phylomes qui ne portent pas en général sur les mêmes ensembles d'espèces, par exemple en raison d'une incapacité à détecter le gène concerné par l'arbre phylogénétique chez certaines des espèces voulues, soit pour des raisons méthodologiques (la recherche d'une séquence similaire à la séquence du gène d'intérêt a échoué dans ces espèces), soit pour des raisons biologiques (le gène a subi une **suppression**, c'est-à-dire que suite à l'évolution, il a été perdu dans ces espèces). On dira qu'on cherche alors à reconstuire des **super-réseaux**.

Pour s'accommoder de ce silence dans les données, une première approche consiste à inférer les données manquantes. Pour les triplets, on peut utiliser par exemple une mé-

thode de super-arbre pour inférer des triplets qui ne seront contradictoires avec aucun de ceux présents en entrée du problème [Ranwez *et al.*, 2007].

Pour les bipartitions, plusieurs méthodes ont été proposées spécifiquement pour traiter ce problème. La Z-clôture [Meacham, 1983; Huson *et al.*, 2004] tout d'abord, illustrée en figure 3.4, consiste à inférer les deux bipartitions $A_1|B_1 \cup B_2$ et $A_1 \cup A_2|B_2$ à partir des bipartitions $A_1|B_1$ et $A_2|B_2$ telles que $A_1 \cap A_2 \neq \emptyset$, $A_2 \cap B_1 \neq \emptyset$, $B_1 \cap B_2 \neq \emptyset$ et $A_1 \cap B_2 = \emptyset$. Pour un ensemble de b bipartitions sur n taxons, le fait d'appliquer la Z-clôture autant de fois que possible peut être réalisé par un algorithme en temps $O(nb^3)$ et en espace $O(nb)$ pour obtenir au plus $n+b$ bipartitions. Toutefois, selon l'ordre d'application des Z-clôtures dans cet algorithme, on obtiendra un résultat différent.

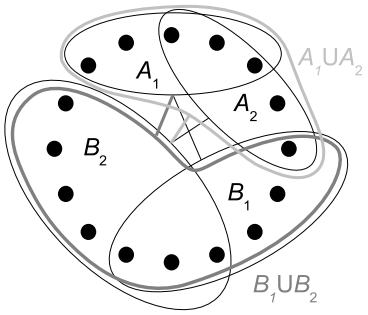


FIGURE 3.4 : Opération de Z-clôture : les points représentants les feuilles, à partir des bipartitions $A_1|B_1$ et $A_2|B_2$ (où A_1, B_1, A_2 et B_2 sont les ovales noirs), on obtient les bipartitions $A_1|B_1 \cup B_2$ et $A_1 \cup A_2|B_2$

Une autre méthode consistant à générer les quadruplets induits par les bipartitions d'arbres fournis en entrée, puis à compléter ces arbres en plaçant les taxons manquants aux positions les plus probables, est appelée **Q-imputation** [Holland *et al.*, 2007]. Enfin, deux autres opérations de clôtures de bipartitions ont été proposées plus récemment, la **M-clôture** (règle binaire comme la Z-clôture) et la **Y-clôture** (règle ternaire) [Grünewald *et al.*, 2008], qui préservent toutes deux l'éventuel caractère circulaire des bipartitions.

Toutefois, ces méthodes ont un temps de calcul élevé, et donnent peu de garanties sur leur résultat. Il serait donc préférable d'utiliser des données les plus complètes possibles a priori, plutôt que de tenter de les compléter algorithmiquement, problème que nous aborderons en section 4.1.3.

3.2 Explosion de complexité en fonction du niveau

D'après la section 1.4.3, les réseaux de niveau k peuvent être vus de manière simplifiée comme des arbres de générateurs, qui sont les briques de base que l'on peut assembler pour créer de tels réseaux. Les algorithmes de reconstruction évoqués en section 2.1.2 ex-

ploitent cette structure en tentant de commencer par trouver les isthmes du réseau qui correspondent aux arcs de cet arbre.

Toutefois, cette vision simplifiée cache pleinement l'ampleur de la complexité du réseau à l'intérieur des blobs. Dans cette section, afin d'étudier cette complexité, nous évaluons tout d'abord le nombre de générateurs de niveau fixé, et montrons qu'il est en fait exponentiel en fonction du niveau. Puis nous estimons le niveau de réseaux phylogénétiques explicites générés par simulation, afin de montrer que dans les modèles testés, le niveau augmente de façon à peu près aussi élevée que le nombre total de réticulations du réseau.

3.2.1 Bornes sur le nombre de générateurs

Les règles R1 et R2 de construction des générateurs de niveau k à partir de ceux de niveau $(k-1)$, introduites en section 1.4.3, peuvent être utilisées pour obtenir des bornes inférieures et supérieures sur le nombre de générateurs de niveau k.

Proposition 13 *Le nombre* g_k *de générateurs de niveau* k *est au moins égal à* 2^{k-1}.

Démonstration. La propriété est vraie pour $k = 0$, fixons $k \geq 1$. On définit une injection G_k entre l'ensemble des entiers $[0..2^{k-1}-1]$ et l'ensemble de générateurs de niveau k. Le générateur $G_k(a)$ est construit à partir de la représentation binaire de l'entier a en utilisant uniquement la règle R1.

Le processus de construction est illustré en figure 3.5. Exprimons a sous la forme $\sum_{i=0}^{k-2} a_i 2^i \in [0..2^{k-1}-1]$ tel que $a_i \in \{0,1\}$. Commençons à partir du générateur $\mathcal{G}^1$ de niveau 1, puis pour i variant de 0 à $k-2$:

- soit h_i un sommet hybride minimal pour la relation de descendance dans le générateur G construit à cette étape.
- soit e_i l'arc de source le plus grand parent de h_i (une simple récurrence montre qu'il existe toujours un parent de h_i strictement plus grand que l'autre).
- remplacer G par $R1(G, e_i, h_i)$ si a_i=1, et par $R1(G, e_i, e_i)$ si $a_i = 0$.

Ainsi, nous obtenons un graphe orienté $G_k(a)$ dont la structure est une chaîne de cycles qui code la représentation binaire de a. La proposition 5 assure que $G_k(a)$ est un générateur de niveau k. On peut donc, pour chaque k, construire un ensemble $\{G_k(a), a \in [0..2^{k-1}-1]\}$ de 2^{k-1} générateurs de niveau k. Étant chacun composés d'une chaîne spécifique constituée de deux types de "maillons", ils sont clairement non isomorphes. □

Proposition 14 *Pour* $k \geq 1$*, un générateur de niveau* k *a au plus* $3k-1$ *sommets et au plus* $4k-2$ *arcs.*

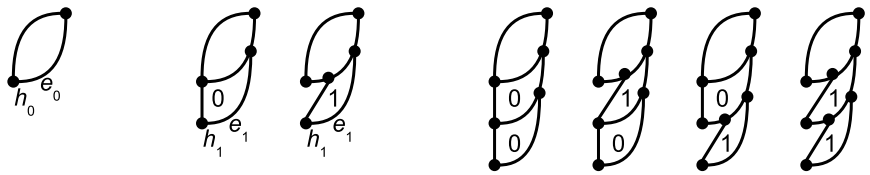


FIGURE 3.5 : Construction de 2^{k-1} générateurs non isomorphes de niveau k : on part du générateur $\mathcal{G}^1$ (a) et on applique $R1(\mathcal{G}^1, e_0, h_0)$ pour obtenir $G_2(0)$ (b), $G_2(1) = R1(\mathcal{G}^1, e_0, e_0)$ (c), $G_3(0) = R1(G_2(0), e_1, h_1)$ (d), $G_3(1) = R1(G_2(1), e_1, h_1)$ (e), $G_3(2) = R1(G_2(0), e_1, e_1)$ (f), $G_3(3) = R1(G_2(1), e_1, e_1)$ (g).

Démonstration. L'unique générateur de niveau 1 a deux sommets et deux arcs. Par le corollaire 1.20, chaque générateur de niveau k est obtenu par application de la règle R1 ou R2 à un générateur de niveau $(k-1)$, donc par k applications des règles R1 ou R2 au total. Nous remarquons alors que chaque application de règle R1 ou R2 ajoute au plus trois sommets et quatre arcs. Ces bornes sont atteintes quand la règle R2 est utilisée de façon répétée sur deux arcs différents comme dans la figure 1.19(e), page 44. □

Proposition 15 *Le nombre* g_k *de générateurs de niveau* k *est au plus égal à* $k!^2 50^k$.

Démonstration. La proposition 14 assure que le nombre d'arcs d'un générateur de niveau k est au plus $4k$, et que son nombre de sommets hybrides est k, donc son nombre de côtés est au plus $5k$. En appliquant pour la k-ième fois la règle R1 ou R2, on choisit une paire de côtés, c'est-à-dire de sommets hybrides ou d'arcs, donc il y a moins de $(5k)^2$ possibilités. Ainsi, $g_{k+1} \leq 2(5k)^2 g_k < 50k^2 g_k$, et finalement $g_k < k!^2 50^k$. □

Notons que même si ces bornes sont loin d'être fines, elles donnent des informations utiles sur les générateurs de niveau k. La borne inférieure montre qu'il existe un nombre exponentiel de générateurs de niveau k, ce qui implique, par le théorème de décomposition de la section 1.4.3, une complexité importante à l'intérieur des blobs de haut niveau. La borne supérieure pour la valeur de g_{k+1} en fonction de g_k, et le fait que $g_3 = 65$ [Kelk, 2008], montrent qu'il semble réaliste de construire automatiquement l'ensemble des générateurs de niveau 4 et 5 au moins. Nous proposons donc maintenant un algorithme incrémental de construction des générateurs de niveau k à partir des générateurs de niveau $(k-1)$, polynomial en la taille des données en entrée.

3.2.2 Algorithme de construction des générateurs de niveau k

Nous étudions maintenant la mise en application des règles R1 et R2 pour construire les générateurs de niveau $(k+1)$ connaissant l'ensemble de ceux de niveau k. Notons que deux séquences différentes de règles sont susceptibles de produire des générateurs de niveau k isomorphes. Ainsi, les générateurs de niveau k isomorphes doivent être détectés et supprimés lors de ce processus de construction.

Théorème 15 *Il existe un algorithme qui prend en entrée l'ensemble S_k^* des générateurs de niveau k, et renvoie en temps polynomial en $|S_k^*|$ l'ensemble des générateurs de niveau $(k+1)$.*

Démonstration. L'algorithme, *BuildGenerators*, est décrit dans la figure 3.7.

D'après la preuve de la proposition 15, les règles R1 et R2 sont appliquées au plus $50k^2|S_k^*|$ fois dans l'algorithme. La proposition 13 assure que $|S_k^*| \geq 2^{k-1}$, donc $k = o(|S_k^*|)$, et par la proposition 14, la taille d'un générateur est aussi polynomiale en $|S_k^*|$. Le nombre de tests d'isomorphisme est donc polynomial en $|S_k^*|$.

Pour prouver que cet algorithme est polynomial en la taille de l'entrée S_k^*, il reste donc à prouver que le test d'isomorphisme peut être effectué en temps polynomial en k.

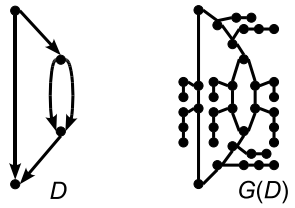
L'isomorphisme de graphes orientés est GRAPH ISOMORPHISM-complet [Zemlyachenko *et al.*, 1985], ce qui implique qu'aucun algorithme polynomial n'est actuellement connu pour résoudre ce problème dans le cas général. Toutefois, on se restreint ici au cas particulier d'instances où les graphes *orientés* ont degré maximal 3, et il est possible de décider en temps polynomial de l'isomorphisme de graphes *non orientés* de degré borné [Luks, 1982]. Montrons comment réduire le problème de l'isomorphisme de graphes orientés de degré maximal 3 et de degré sortant et entrant maximal 2 au problème de l'isomorphisme de graphes de degré borné.

Pour tout graphe orienté D dont les sommets ont degré au plus 3, degré sortant et entrant 2, et d'éventuels arcs multiples, nous utilisons le gadget introduit par Miller [1979] pour construire, comme illustré en figure 3.6, un graphe $G(D)$ de la manière suivante :

- tous les sommets de D sont des sommets de $G(D)$,
- tout arc (u,v) de D est transformé en un chemin de longueur 4 $u-u'-v'-v$, complété par un chemin de longueur 2 attaché à u' et un chemin de longueur 3 attaché à v'.

Cette construction, est faite en temps polynomial en la taille du graphe et fournit un graphe de degré maximal 3. Elle assure de plus que D_1 est isomorphe à D_2 si et seulement si $G(D_1)$ est isomorphe à $G(D_2)$. □

Bien que l'isomorphisme de graphes soit décidable en temps polynomial pour des graphes de degré maximum borné, il n'existe aucune implémentation de l'algorithme de Luks, qui semble difficile à utiliser en pratique [Kaibel et Schwartz, 2003].



(a) (b)

FIGURE 3.6 : Un graphe orienté D (a) et le graphe G(D) associé (b) par la transformation proposée par Miller [Miller, 1979] qui préserve l'isomorphisme et le degré borné.

BuildGenerators(S^*_k : ensemble de générateurs de niveau k)
1. $S \leftarrow \emptyset$
2. **Pour** tout générateur g de niveau k dans S^*_k **faire**
3. **Pour** toute paire (X,Y) de côtés de g **faire**
4. **Si** la règle R1 peut être appliquée sur les côtés X et Y **alors**
5. $g' \leftarrow R1(g,X,Y)$
6. **Pour** tout générateur h de niveau $(k+1)$ dans S **faire**
7. **Si** g' n'est pas isomorphe à h **alors** $S \leftarrow S \cup \{g'\}$
8. **Si** la règle R2 peut être appliquée sur les côtés X et Y **alors**
9. $g' \leftarrow R2(g,X,Y)$
10. **Pour** tout générateur h de niveau $(k+1)$ dans S **faire**
11. **Si** g' n'est pas isomorphe à h **alors** $S \leftarrow S \cup \{g'\}$
12. **Renvoyer** S

FIGURE 3.7 : L'algorithme *BuildGenerators* construit l'ensemble S des générateurs de niveau $(k+1)$ à partir de l'ensemble S^*_k des générateurs de niveau k, en temps polynomial.

Pour construire les générateurs de niveau 4 à partir de ceux de niveau 3, nous avons donc opté pour un algorithme exponentiel qui parcourt les deux générateurs en parallèle, depuis la racine, en essayant d'identifier les sommets communs pour détecter l'isomorphisme. Parmi les 8501 générateurs de niveau 4 construits en appliquant les règles R1 ou R2, un total de 1993 ne sont pas isomorphes. La liste des 91454 générateurs non isomorphes de niveau 5 a également été construite en une dizaine d'heures de calcul sur un ordinateur portable, avec une implémentation de l'algorithme 3.7. Les listes de générateurs de niveau au plus 5, le programme pour les construire, son code source en Java et des détails d'implémentation sont disponibles à l'adresse http://generators.gambette.com. On notera que la séquence d'entiers 1, 4, 65, 1993, 91454 n'est pas présente dans l'Encyclopédie en ligne des séquences d'entiers connues [Sloane, 2010].

3.2.3 Niveau élevé de réseaux simulés

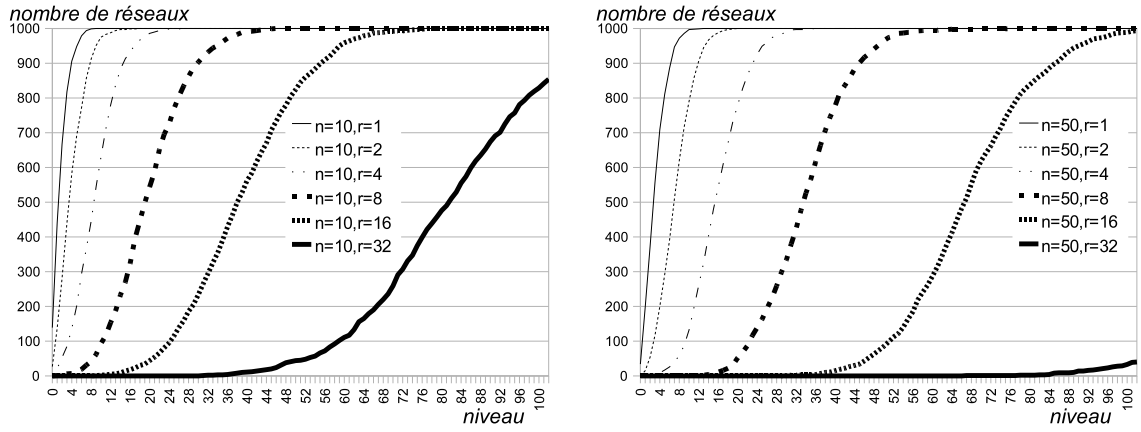
Arenas *et al.* [2008] ont étudié les propriétés des ensembles de réseaux phylogénétiques simulés selon le modèle coalescent avec recombinaison, en mesurant la proportion parmi ces réseaux de ceux appartenant à certaines sous-classes, en particulier les arbres et les réseaux de niveau 1. Nous avons prolongé leur étude, grâce aux données de simulation qu'ils nous ont transmises, en calculant le niveau de tous les réseaux phylogénétiques générés par leur simulation réalisée avec le programme Recodon [Arenas et Posada, 2008]. L'implémentation en Java d'un algorithme basique de décomposition en blobs pour calculer le niveau est également disponible à l'adresse http://generators.gambette.com.

Pour de petites valeurs du niveau, les résultats obtenus sont présentés dans la table 3.1 et un aperçu pour les niveaux supérieurs est donné en figure 3.8. Nous observons que les réseaux phylogénétiques avec un petit niveau, comme les autres restrictions étudiées par Arenas *et al.* [2008], ne couvrent qu'une portion réduite des réseaux phylogénétiques correspondant au modèle coalescent avec de forts taux de recombinaison. En fait, les réseaux simulés selon ce modèle n'ont pas vraiment la structure arborée exprimée dans le théorème 1, mais sont le plus souvent constitués d'un gros blob qui contient tous les sommets hybrides. Ce phénomène apparaît même pour de faibles taux de recombinaison, comme montré en figure 3.9.

n	r	arbre	niveau 1	niveau 2	niveau 3	niveau 4	niveau 5
10	0	1000	1000	1000	1000	1000	1000
10	1	139	440	667	818	906	948
10	2	27	137	281	440	582	691
10	4	1	21	53	85	136	201
10	8	0	1	1	6	7	12
10	16	0	0	0	0	0	0
50	0	1000	1000	1000	1000	1000	1000
50	1	34	198	373	557	709	811
50	2	0	15	54	117	200	292
50	4	0	1	1	2	9	17
50	8, 16, 32	0	0	0	0	0	0

TABLE 3.1 : Nombre de réseaux à $n = 10$ et 50 feuilles, ayant niveau 0, 1, 2, 3, 4, 5, en fonction du taux de recombinaison $r = 0$, 1, 2, 4, 8, 16 (simulation de 1000 réseaux selon le modèle coalescent avec recombinaison).

Ainsi, dans ce contexte, de nouvelles structures et techniques algorithmiques doivent être étudiées. Mentionnons toutefois que ce modèle ne convient pas pour décrire tous les cas d'évolution réticulée, et que d'autres peuvent être plus appropriés, comme celui qui insère des transferts horizontaux selon une loi de Poisson [Galtier, 2007], ou ceux utilisés pour la simulation de réseaux phylogénétiques dans NetGen [Morin et Moret, 2006].



(a) (b)

FIGURE 3.8 : Nombre de réseaux de niveau k à 10 (a) et 50 feuilles (b), en fonction de k, à différents taux de recombinaison r =1, 2, 4, 8, 16, 32 (simulation de 1000 réseaux selon le modèle coalescent avec recombinaison).

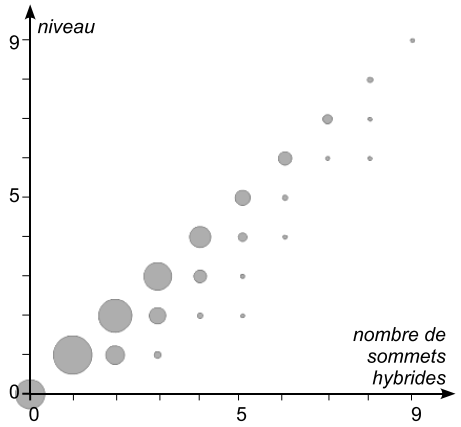


FIGURE 3.9 : Nombre de sommets hybrides et niveau des réseaux à 10 feuilles pour un taux de recombinaison $r = 1$: la taille d'un point de coordonnées (x, y) représente le nombre de réseaux strictement de niveau x contenant exactement y sommets hybrides (simulation de 1000 réseaux selon le modèle coalescent avec recombinaison).

3.3 Fiabilité des réseaux obtenus par les méthodes combinatoires

Nous nous intéressons maintenant à l'ensemble des solutions possibles à partir des données en entrée. La question de l'unicité du réseau solution est particulièrement inté-

ressante car si elle n'est pas garantie, il y aura ambiguïté de la solution, qui ne pourra alors pas être considérée comme fiable.

Toutefois ce problème semble complexe et peu de résultats existent sur l'unicité des réseaux reconstruits à partir d'ensembles de triplets. Cette question est présentée comme un problème ouvert par van Iersel *et al.* [2009a], et un premier élément de réponse est donné par van Iersel *et al.* [2009b], avec la construction, pour tout niveau $k \geq 2$, d'un réseau N^k de niveau k qui est le seul réseau de niveau k qui contient tous les triplets de $\mathcal{R}(N^k)$.

Précisons que dans cette section on interdit, comme van Iersel *et al.* [2009b], la présence de feuilles de degré entrant strictement supérieur à 1 dans les réseaux de niveau k. En effet, les autoriser revient à autoriser des réseaux qui n'apportent aucune information topologique supplémentaire et empêchent toute unicité.

Nous étudions ici un problème légèrement différent, celui de l'**encodage** des réseaux de niveau k par leur ensemble de triplets : il consiste à déterminer, si pour un réseau N donné de niveau k, il existe un autre réseau N' de niveau k tel que $\mathcal{R}(N) = \mathcal{R}(N')$. Ainsi, cela correspond à un problème d'unicité d'un réseau en fonction de l'ensemble de tous les triplets qu'il contient.

Nous résolvons ci-dessous cette question pour les réseaux de niveau 1 en caractérisant les réseaux encodables et non encodables. De plus, nous répondons également à cette question pour l'encodage par les clades souples, et les arbres contenus.

Mentionnons que des résultats similaires existent pour une autre classe restreinte de réseaux. En effet, si $\mathcal{T}(N)$ est l'ensemble des arbres contenus dans un réseau régulier N, alors il n'existe aucun autre réseau N' régulier tel que $\mathcal{T}(N') = \mathcal{T}(N)$ [Willson, 2010b]. L'algorithme de reconstruction d'un réseau à partir de son ensemble d'arbres contenus est de plus présenté par son auteur comme utilisable comme heuristique de reconstruction de réseaux phylogénétiques à partir d'arbres enracinés.

3.3.1 Encodage des réseaux simples de niveau 1

Etudions tout d'abord le cas de l'encodage des réseaux simples de niveau 1 par leur ensemble de triplets. Une analyse de cas montre que ceux dont le blob a quatre sommets ne sont pas encodables, puisque les trois configurations $\mathcal{G}_1^1$, $\mathcal{G}_2^1$ et $\mathcal{G}_3^1$, illustrées en figure 3.10, correspondent au même ensemble de triplets.

Toutefois le lemme suivant montre que dès lors que le blob contient strictement plus de quatre sommets, il n'y a plus d'ambiguïté et aucun autre réseau simple de niveau 1 ne contient exactement le même ensemble de triplets.

Etant donné un réseau phylogénétique explicite enraciné binaire contenant un arc e, et dont l'ensemble des feuilles ne contient pas d, notons $N \oplus_e d$ le réseau obtenu de la façon suivante : ajouter deux nouveaux sommets d et w à $V(N)$, et remplacer l'arc e par les arcs (u,w), (w,v), et (w,d).

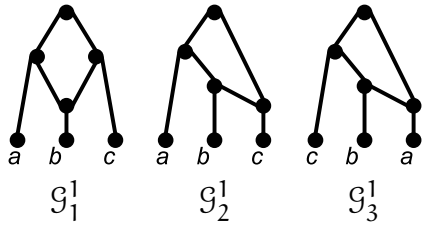


FIGURE 3.10 : Les trois réseaux simples de niveau 1 non isomorphes dont l'ensemble des triplets contenus est exactement $\mathcal{R} = \{a|bc, c|ab\}$.

Lemme 9 *Soit* $X = \{a, b, c, d\}$. *Alors, pour tous* $i \neq j \in \{1,2,3\}$, *pour tous arcs* $e_i \in \mathcal{G}_i^1$ *et* $e_j \in \mathcal{G}_j^1$

$$\mathcal{R}(\mathcal{G}_i^1 \oplus_{e_i} d) \neq \mathcal{R}(\mathcal{G}_j^1 \oplus_{e_j} d).$$

Démonstration. La définition 3.2 correspond à une analyse exhaustive de toutes les possibilités d'ajout d'une feuille d étant donné un réseau dont l'ensemble des triplets est $\mathcal{R}(\mathcal{G}_i^1)$. Donc tous les réseaux de la forme $\mathcal{R}(\mathcal{G}_i^1 \oplus_{e_i} d)$ correspondent à une de ces possibilités, et du fait que les zones sont mutuellement exclusives (puisque les ensembles de triplets sur $\{a, b, c, d\}$ auxquelles elles correspondent sont tous distincts), on ne peut avoir $\mathcal{R}(\mathcal{G}_i^1 \oplus_{e_i} d) \neq \mathcal{R}(\mathcal{G}_j^1 \oplus_{e_j} d)$ pour $i \neq j \in \{1,2,3\}$. □

3.3.2 Encodage des réseaux de niveau 1

Pour obtenir la caractérisation des réseaux de niveau 1 encodables par leur ensemble de triplets, nous allons tout d'abord prouver le lemme suivant, équivalent pour les triplets du théorème 6 pour les quadruplets.

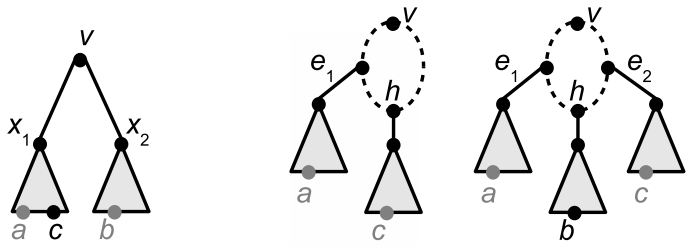
Lemme 10 *Soit* N *un réseau de niveau* 1. *L'ensemble de ses isthmes est en bijection avec les SN-ensembles de son ensemble de triplets* $\mathcal{R}(N)$.

Démonstration. Pour un isthme e de N, considérons l'ensemble A des feuilles descendantes de sa cible. Pour toutes feuilles $a_1, a_2 \in A$ et $x \notin A$, l'unique triplet sur $\{a_1, a_2, x\}$ contenu dans N est $a_1 a_2|x$. Ainsi, A est un SN-ensemble de $\mathcal{R}(N)$.

Inversement, supposons par l'absurde qu'il existe un SN-ensemble A qui ne soit pas l'ensemble des feuilles descendantes de la cible d'un isthme de N. Considérons alors l'arbre T_D obtenu à partir de N en contractant tous les arcs de N qui appartiennent à des blobs de N (ainsi les isthmes de T_D sont exactement les isthmes de N, et ses composantes biconnexes sont de simples sommets). Considérons le sommet $v = lca_{T_D}(A)$.

Si v a degré sortant 2, appelons x_1 et x_2 ses fils, comme montré en figure 3.11(i). Il existe alors une feuille a de A descendante de x_1 et une feuille b de A descendante de

x_2, ainsi qu'une feuille c de $X-A$ descendante de x_1 ou x_2 (disons x_1), puisque A n'est pas l'ensemble des feuilles descendantes de la cible d'un isthme de N. Comme $a \in A$ et $c \in X-A$ sont toutes deux descendantes de x_1 et pas b, alors $b|ac \in \mathcal{R}(N)$, ce qui est impossible puisque A est un SN-ensemble de $\mathcal{R}(N)$.



(i) (ii) (iii)

FIGURE 3.11 : Configurations impossibles d'un SN-ensemble A dans un réseau N de niveau 1 dans le cas où le plus petit ancêtre commun v de A est un blob trivial de N (i), la racine d'un blob non trivial de N dont le sommet hybride est ancêtre d'au moins un sommet de A (ii), ou la racine d'un blob non trivial de N dont les descendants du sommet hybride ne sont que des sommets de $X-A$ (iii). Les feuilles en gris appartiennent à A alors que celles en noir sont dans $X-A$. Les arcs en tirets indiquent la présence d'un chemin orienté de v vers h.

Si v a degré sortant supérieur strictement à 2, alors v correspond à un blob B de N, dont le réseau non orienté sous-jacent est un cycle, et donc B a un sommet hybride h. Deux cas se présentent alors :

- Soit il existe une feuille $c \in A$ parmi les descendants de h comme illustré en figure 3.11(ii), alors il existe un isthme e_1 incident à un sommet de B distinct de h dont la descendance contient au moins une feuille de A, appelée a (sinon le $lca_{T_D}(A)$ ne serait pas v mais un de ses descendants). S'il existe une feuille $b \in X-A$ parmi les descendants de h, alors $cb|a \in \mathcal{R}(N)$ et donc A n'est pas un SN-ensemble de $\mathcal{R}(N)$. De même s'il existe une feuille $b \in X-A$ parmi les descendants de la cible de e_1, car dans ce cas $ab|c \in \mathcal{R}(N)$. Ainsi, la seule solution pour que A ne soit pas l'ensemble des feuilles descendantes de v (et donc de la cible d'un isthme de N) est qu'il existe un autre isthme e_2 incident à un sommet de B tel qu'une feuille $b \in X-A$ est descendante de sa cible. Dans ce cas, deux triplets parmi $a|bc$, $b|ac$ et $c|ab$ sont contenus dans le réseau N, ce qui est impossible puisque A est un SN-ensemble de $\mathcal{R}(N)$.
- Soit tous les descendants de h sont des feuilles de $X-A$, appelons b l'une d'entre elles, comme dans la figure 3.11(iii). Alors il existe au moins deux autres isthmes e_1 et e_2 incidents à des sommets de B dont la descendance contient un sommet a de A pour e_1 et un sommet c de A pour e_2 (sinon le $lca_{T_D}(A)$ ne serait pas v mais un de ses descendants). Dans tous les cas, deux triplets parmi $a|bc$, $b|ac$ et $c|ab$ sont

contenus dans le réseau N, ce qui est impossible puisque A est un SN-ensemble de $\mathcal{R}(N)$.

Ainsi, nous aboutissons à une contradiction dans tous les cas, donc tout SN-ensemble de $\mathcal{R}(N)$ est l'ensemble des feuilles descendantes de la cible d'un isthme de N. □

Le lemme suivant nous permettra de déduire la caractérisation des réseaux de niveau 1 par leur ensemble d'arbres contenus et de clades souples, à partir de celle par leur ensemble de triplets.

Lemme 11 *Soit* N *un réseau de niveau* 1 *ayant au moins trois feuilles. Alors*

$$\mathcal{R}(N) = \bigcup_{T \in \mathcal{T}(N)} \mathcal{R}(T) = \bigcup_{C \in \mathcal{S}(N)} \mathcal{R}(C).$$

Démonstration. D'après la définition des clades souples, $\bigcup_{C\in\mathcal{S}(N)}\mathcal{R}(C) = \bigcup_{T\in\mathcal{T}(N)} \bigcup_{C\in\mathcal{C}(T)}\mathcal{R}(C)$. Or, tout $\mathcal{R}(C)$ est inclus dans $\mathcal{R}(T)$ d'après la proposition 2 et inversement, pour tout $a|bc \in \mathcal{R}(T)$, si l'on considère le clade $\mathcal{C}_T(lca(a,c))$, $a|bc \in \mathcal{R}(\mathcal{C}_T(lca_T(a,c)))$, donc $\bigcup_{C\in\mathcal{C}(T)}\mathcal{R}(C) = \mathcal{R}(T)$. Ainsi, on a l'égalité $\bigcup_{C\in\mathcal{S}(N)}\mathcal{R}(C) = \bigcup_{T\in\mathcal{T}(N)}\mathcal{R}(T)$.

L'égalité $\mathcal{R}(N) = \bigcup_{T\in\mathcal{T}(N)}\mathcal{R}(T)$ est donnée directement par la remarque 5. □

On peut aller plus loin sur le lien entre les triplets et les clades d'un réseau de niveau 1, avec la proposition suivante, que nous annoncions à la page 27 suite à la proposition 2.

Proposition 16 *Un clade* C *est un clade souple d'un réseau* N *de niveau 1, si et seulement si* $\mathcal{R}(C)$ *est un ensemble de triplets de* N.

Démonstration. $\Rightarrow$: voir proposition 2.

$\Leftarrow$: rappelons que d'après la remarque 10 de la page 39, un ensemble de sommets n'a qu'un plus petit ancêtre commun dans un réseau de niveau 1. Soit C un ensemble de feuilles tel que $\mathcal{R}(C)$ est un ensemble de triplets de N, et soient $x_1, x_2 \in C$ tels que $lca_N(\{x_1,x_2\}) = lca_N(C)$. Pour tout x de $X - C$, $x_1x_2|x \in \mathcal{R}(N)$ par définition de C, donc il existe un chemin de la racine de N vers x qui ne contient pas $lca_N(C)$, donc N contient un arbre T où $C = C_T(lca_N(C))$, donc $C \subseteq \mathcal{S}_N(lca_N(C))$. □

Nous pouvons maintenant donner notre théorème de caractérisation, après avoir introduit une dernière notation. Étant donné un ensemble $\mathcal{R}$ de triplets (respectivement, de clades souples $\mathcal{S}$, d'arbres $\mathcal{T}$), l'ensemble des réseaux N strictement de degré 1 tels que $\mathcal{R}(N) = \mathcal{R}$ (resp. $\mathcal{S}(N) = \mathcal{S}$, $_T(N) = T$) est noté $\mathfrak{N}_1(\mathcal{R})$ (resp. $\mathfrak{N}_1(\mathcal{S})$, $\mathfrak{N}_1(\mathcal{T})$).

Ainsi, un réseau est encodable par ses triplets (respectivement, ses clades souples, ses arbres contenus) si $|\mathfrak{N}_1(\mathcal{R})| = 1$ (resp. $|\mathfrak{N}_1(\mathcal{S})| = 1$, $|\mathfrak{N}_1(\mathcal{T})| = 1$).

Théorème 16 *Soit* N *un réseau de niveau 1 ayant au moins trois feuilles. Alors les affirmations suivantes sont équivalentes :*

(i) N *contient un blob à quatre sommets.*

(ii) N *n'est pas encodable par ses triplets, c'est-à-dire* $|\mathfrak{N}_1(\mathcal{R}(N))| > 1$.

(iii) N *n'est pas encodable par ses clades souples, c'est-à-dire* $|\mathfrak{N}_1(\mathcal{S}(N))| > 1$.

(iv) N *n'est pas encodable par ses arbres contenus, c'est-à-dire* $|\mathfrak{N}_1(\mathcal{T}(N))| > 1$.

Démonstration. (i) $\Rightarrow$ (iv) : ceci découle directement du fait que tous les réseaux simples de niveau 1 de la figure 3.10 contiennent le même ensemble d'arbres.

(iv) $\Rightarrow$ (iii) : comme $|\mathfrak{N}_1(\mathcal{T}(N))| > 1$, soit N' un réseau de niveau 1 distinct de N tel que $\mathcal{T}(N) = \mathcal{T}(N')$. Par définition des clades souples, $\mathcal{S}(N) = \bigcup_{T \in \mathcal{T}(N)} \mathcal{C}(T) = \bigcup_{T \in \mathcal{T}(N')} \mathcal{C}(T) = S(N')$, et donc $|\mathfrak{N}_1(\mathcal{S}(N))| > 1$.

(iii) $\Rightarrow$ (ii) : comme $|\mathfrak{N}_1(\mathcal{T}(N))| > 1$, soit N' un réseau de niveau 1 distinct de N tel que $\mathcal{S}(N) = \mathcal{S}(N')$. Alors $\mathcal{R}(N') = \bigcup_{C \in \mathcal{S}(N')} \mathcal{R}(C)$ d'après le lemme 11, donc $\mathcal{R}(N') = \bigcup_{C \in \mathcal{S}(N)} \mathcal{R}(C) = \mathcal{R}(N)$ et donc $N' \in \mathfrak{N}_1(\mathcal{S}(N))$.

(ii) $\Rightarrow$ (i) : Soit N' un réseau de niveau 1 distinct de N tel que $\mathcal{R}(N) = \mathcal{R}(N')$. D'après le lemme 10, $\mathcal{R}(N)$ et $\mathcal{R}(N')$ ayant les mêmes SN-ensembles, N et N' ont la même structure d'isthmes. Comme N et N' sont disctincts, il doit exister un blob B de N et un blob B' de N' tels que les deux blobs ont les mêmes feuilles descendantes mais sont distincts. Sans perte de généralité, considérons un tel blob B dont la racine est minimale pour la relation de descendance dans N. D'après le lemme 8, il existe trois feuilles a, b et c descendantes de sommets de B telles que les triplets $a|bc$ et $b|ac$ sont contenus dans N mais pas $c|ab$. Comme $\mathcal{R}(N) = \mathcal{R}(N')$ alors les feuilles a, b et c descendantes de sommets de B' sont aussi telles que les triplets $a|bc$ et $b|ac$ sont contenus dans N'. Or, d'après le lemme 9, s'il existe un sommet d' de B incident à un autre isthme que ceux menant à la racine de B, et aux feuilles a, b, et c, $\mathcal{R}(N)_{|\{x \preceq_N y | y \in B\}} \neq \mathcal{R}(N')_{|\{x \preceq_{N'} y | y \in B'\}}$, et donc $\mathcal{R}(N) \neq \mathcal{R}(N')$. Ainsi, il n'existe pas de tel sommet d', et donc B, et B', n'ont que quatre sommets. □

Ce théorème implique immédiatement le corollaire suivant, sur le nombre de réseaux de niveau 1 ayant le même ensemble de triplets, clades souples et arbres.

Corollaire 3.3 *Soit* N *un réseau de niveau 1 ayant au moins 3 feuilles. Le nombre total de réseaux* N' *de niveau 1, non isomorphes, tels que* $\mathcal{R}(N') = \mathcal{R}(N)$ *(ou de façon équivalente, tels que* $\mathcal{S}(N) = \mathcal{S}(N')$*, ou tels que* $\mathcal{T}(N) = \mathcal{T}(N')$*) est* 3^b*, où* b *est le nombre de blobs de* N *à quatre sommets.*

3.3.3 Encodage des réseaux de niveau 2 et plus

Notre caractérisation des réseaux de niveau 1 encodés par leur ensemble de triplets fait apparaître que dans certains cas (quand le réseau contient un blob à 4 sommets), même si

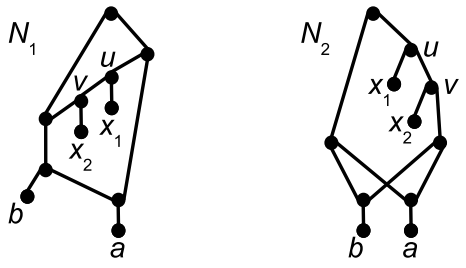
nous avons toutes les données possibles concernant la topologie des arbres, clades souples et triplets qu'ils contiennent, nous n'arrivons pas à déterminer de façon certaine la topologie du réseau à l'intérieur des blobs. On pourra toutefois nous objecter que ces topologies sont très proches dans les cas problématiques pour le niveau 1, puisque les configurations qui présentent une ambiguïté correspondent à une même version du réseau quand on passe à une version non enracinée. Se pourrait-il que ces ambiguïtés ne correspondent finalement qu'à une incertitude sur l'enracinement du réseau ?

Une autre interrogation concerne la qualité des données fournies en entrée. Peut-être que nous obtenons un blob contenant seulement quatre sommets car nous le connaissons mal : par exemple nous avons omis de séquencer une espèce également issue de ce blob, qui aurait induit un cinquième sommet dans le blob, permettant ainsi de lever l'ambiguïté ?

L'étude des niveaux supérieurs montre que pour ces deux questions, la réponse est négative, et que nous faisons face à de réelles ambiguïtés. Nous ne sommes pas parvenus à réaliser comme pour le niveau 1 une caractérisation exhaustive des réseaux encodés par leur ensemble d'arbres, clades souples et triplets, mais avons trouvé des exemples intéressants de réseaux non encodés par ces éléments pour répondre à ces questions.

Par exemple, les deux réseaux N_1 et N_2 de niveau 2 de la figure 3.12 contiennent exactement le même ensemble de triplets, et ont pourtant des configurations très différentes. En particulier leurs différences ne proviennent pas d'un enracinement différent d'un même réseau non enraciné. Ces deux réseaux simples ont exactement le même niveau, le même nombre de sommets, de sommets de spéciation, de sommets hybrides, et d'arcs : ainsi, impossible de considérer que l'un est plus parcimonieux que l'autre.

On remarque également qu'il n'y a plus comme pour le niveau 1 une borne sur la taille des blobs : en remplaçant l'arc (uv) dans N_1 et N_2 par un chemin orienté de u à v, et en ajoutant une feuille comme enfant de tous les sommets internes de ce chemin, les deux réseaux obtenus ont les mêmes ensembles de triplets.



(i) (ii)

FIGURE 3.12 : Ces deux réseaux simples de niveau 2 contiennent exactement les mêmes triplets $\{a|x_1b, b|x_1a, x_1|ab, a|x_2b, b|x_2a, x_2|ab, x_1|x_2a, a|x_1x_2, x_1|x_2b, b|x_1x_2\}$.

Toutefois, les ensembles de clades souples et d'arbres de ces deux réseaux diffèrent : par

exemple le clade souple $\{a, x_2\}$ est contenu dans N_2 mais pas dans N_1, et l'arbre constitué d'une racine parent de b et de la racine du triplet $a|x_1x_2$ est contenu dans N_1 mais pas dans N_2.

Les quatre réseaux de la figure 3.13, en revanche, ont le même ensemble de triplets et le même ensemble de clades souples. De nouveau, ils sont tout aussi parcimonieux, avec le même niveau, nombre de sommets de spéciation, de sommets hybrides, et d'arcs.

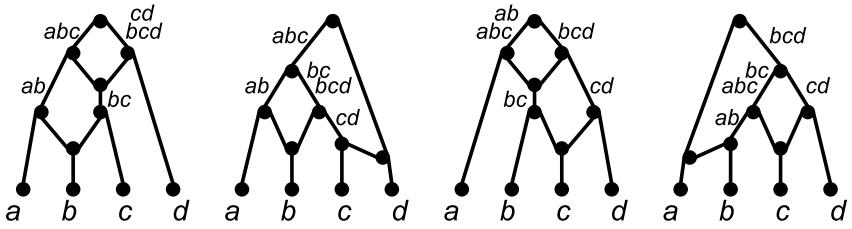


FIGURE 3.13 : Ces quatre réseaux simples de niveau 2 contiennent exactement les mêmes triplets $\{a|bc, c|ab, a|bd, d|ab, a|cd, d|ac, b|cd, d|bc\}$ et les mêmes clades souples $\{\{a, b\}, \{a, b, c\}, \{b, c\}, \{b, c, d\}, \{c, d\}, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b, c, d\}\}$.

Ainsi, il faudra considérer avec précaution la solution proposée par les méthodes de reconstruction de réseaux phylogénétiques à base de triplets ou de clades, car même en présence de données exactes et complètes, elles peuvent proposer un réseau erroné. En fait, des méthodes fournissant l'ensemble des solutions les plus parcimonieuses, éventuellement sous forme compressée en donnant la structure générale de l'arbre des blobs, ainsi que les différentes variantes pour chaque blob, seront préférées si l'on recherche un résultat fiable. Il faudra alors avoir recours à un expert ou à d'autres critères pour choisir la solution parmi ces réseaux proposés, comme nous le discuterons en conclusion.

4 Les méthodes combinatoires sur des données réelles

Dans ce chapitre nous évoquons la mise en pratique des méthodes combinatoires sur les données réelles, en évoquant la complexité de ces données, et les choix que doit faire le biologiste pour sélectionner une méthode de reconstruction de réseaux, puis les données qu'il veut sélectionner pour son analyse. Nous concluons par un exemple d'application de diverses méthodes sur des données réelles concernant l'évolution de plusieurs espèces de protéobactéries.

4.1 Sélection et prétraitement des données

4.1.1 Possibilités de types de données en entrée

Nous avons décrit dans les chapitres précédents des algorithmes combinatoires qui fonctionnent en prenant en entrée des arbres ou des éléments mathématiques qui en sont extraits. Nous avons vu comment nous pouvions nous adapter au fait que ces données pouvaient être incomplètes ou incorrectes.

Toutefois, nous nous sommes limités au cas d'arbres portant à peu près sur un même ensemble de taxons et ne contenant pas de taxons répétés. Ces données ne constituent qu'une partie des données biologiques à disposition. En effet, sans avoir à gérer directement dans nos méthodes de reconstruction le volume énorme des données de séquences d'ADN dont ces arbres sont issus, il est possible d'utiliser des données d'arbres donnant plus d'informations sur la réalité biologique.

En nous limitant aux arbres ne contenant pas de taxons répétés, nous ignorons les effets des **duplications**, phénomènes biologiques qui consistent à reproduire une partie du génome, conduisant un même individu à posséder plusieurs versions d'un même gène (appelées versions **paralogues**, les versions d'un gène issues d'un événement de spéciation étant appelées **orthologues** [Fitch, 2000]). Ces deux versions évoluent chacune de manière indépendante au gré des mutations de leur séquence. Ainsi, l'arbre reconstruit sur une famille de gènes **homologues** (c'est-à-dire qui proviennent d'ancêtres communs, que les gènes soient paralogues ou orthologues) contient fréquemment des répétitions de taxons, dans les bases de données de phylomes [Dufayard *et al.*, 2005; Huerta-Cepas *et al.*, 2008]. On l'appelle alors **arbre multi-étiqueté**. Une approche classique consiste à retraiter ces arbres pour ne garder qu'une version de chaque gène par espèce. Cette démarche est

adaptée si les deux paralogues d'une même espèce ont évolué de la même manière par la suite, elle ne l'est pas s'ils ont eu des histoires différentes dues à la recombinaison ou au transfert horizontal.

Une première approche consiste alors à extraire le maximum d'information fournie par les parties sans conflits des arbres multi-étiquetés [Scornavacca *et al.*, 2011]. Mais pour éviter toute perte d'information, il est possible de traiter directement ces arbres comme données d'entrée des méthodes de reconstruction de réseaux phylogénétiques. C'est le cas par exemple du logiciel Padre [Moulton et Huber, 2006; Huber *et al.*, 2007], ou d'approches de réconciliation d'arbres de gènes avec un arbre d'espèces qui tentent de trouver un scénario mêlant duplications, suppressions et transferts horizontaux de gènes [Górecki, 2004; Tofigh *et al.*, 2011]. On peut également ajouter des informations de datation des arbres de gènes pour améliorer la complexité des algorithmes de reconstruction de ces scénarios, comme le fait le logiciel MPR [Doyon *et al.*, 2011].

Les arbres fournis en entrée peuvent également contenir des informations de fiabilité sur chacune de leurs arêtes. Le logiciel Prunier [Abby *et al.*, 2010] les utilise dans une heuristique qui reconstruit des réseaux à partir d'arbres portant sur le même ensemble de taxons. Il considère que parmi les branches des arbres de gènes incompatibles avec l'arbre des espèces, les plus soutenues correspondent aux transferts horizontaux les plus probables.

4.1.2 Choix de la méthode de reconstruction

Face à l'abondance de méthodes mentionnées dans la section précédente et la section 2.1.1, sans oublier les méthodes géométriques, statistiques, ou plus exotiques, il est difficile pour le biologiste de savoir quelle méthode choisir pour les données dont il dispose. Un livre dédié aux réseaux phylogénétiques décrira bientôt les méthodes principales de reconstruction [Huson *et al.*, 2011], toutefois aucune base d'informations complète n'était disponible au début de cette thèse.

Nous avons donc mis en place une bibliographie interactive en anglais pour réunir et structurer les informations sur les réseaux phylogénétiques. La base de données "Who is who in Phylogenetic Networks" [1] [Gambette, 2010] contient plus de 300 publications sur ces réseaux, étiquetées par plus de 100 mots-clés, dont un nuage est montré en figure 4.1. Plus de 40 programmes dédiés à la génération, la reconstruction ou la comparaison des réseaux phylogénétiques sont également décrits.

La base logicielle du site est l'interface web bibliographique BibAdmin [2] conçue par Sergiu Chelcea. Nous l'avons modifiée pour l'adapter aux besoins de bioinformaticiens ou de biologistes à la recherche d'informations et de méthodes sur les réseaux phylogénétiques, en ajoutant notamment la possibilité de définir les mots-clés utilisés pour étiqueter les publications, ainsi que plusieurs outils pour aider à la navigation (nuages de mots cliquables,

1. http://www.atgc-montpellier.fr/phylnet
2. https://gforge.inria.fr/projects/bibadmin/

liens directs vers Google Scholar pour les auteurs, vers la publication et également la pré-publication si elle est disponible en accès libre...), et réunir des informations sur la communauté à l'origine de ces méthodes (évolution du graphe des coauteurs, informations sur les auteurs, nuage de photos pondérées selon le nombre de publications de chaque auteur dans le domaine...)

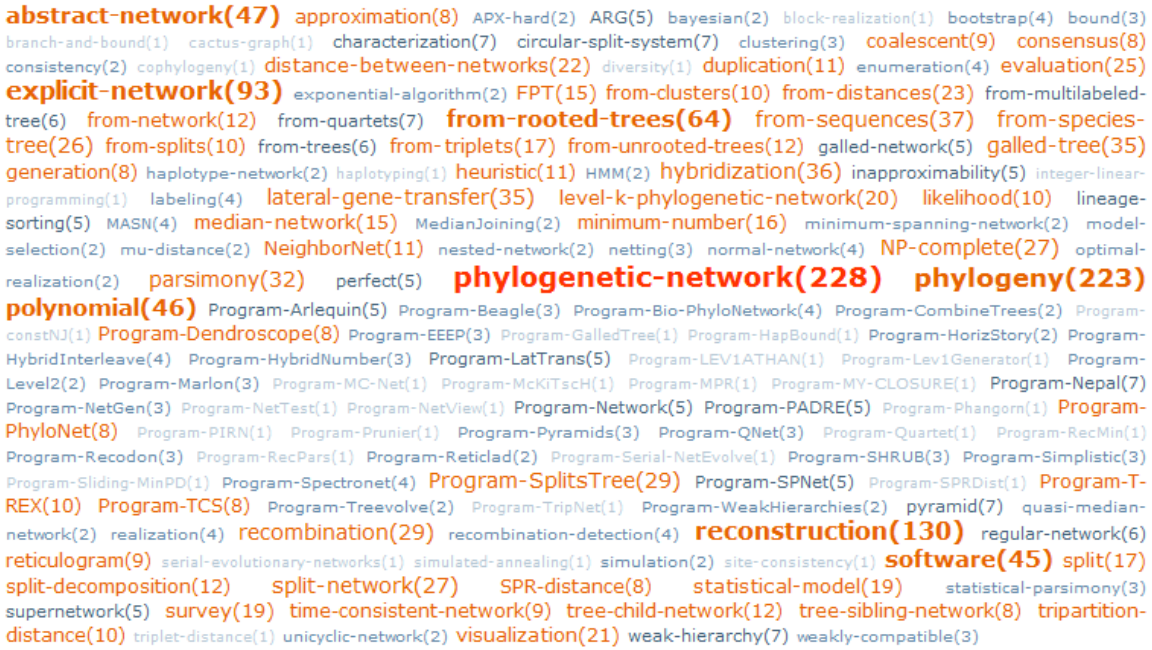


FIGURE 4.1 : Nuage des mots-clés du site "Who is who in phylogenetic networks" pour aider à choisir une méthode de reconstruction ou de traitement de réseaux phylogénétiques.

L'étiquetage des articles à l'aide de mots-clés est à la base de la navigation dans le site web. En effet, ils permettent d'indiquer par exemple le type de données en entrée (les étiquettes commençant par "from" : "from rooted trees", "from distances", "from triplets", etc.). Cliquer sur le mot-clé voulu permettra donc d'afficher toutes les publications traitant de telles méthodes. La page contiendra également un nuage de tous les mots-clés cooccurrents, ce qui permettra par exemple d'identifier des logiciels, dont les mots-clés commencent par "Program" ("Program SplitsTree", "Program Dendroscope", etc.). Pour chacun de ces programmes, une description concise est fournie avec un lien vers la page de téléchargement, que l'utilisateur du site pourra visiter directement, à moins qu'il ne veuille explorer les autres articles associés à ce programme pour en savoir plus sur son contexte d'utilisation. La page de description de ces programmes est accessible directement à l'adresse http://www.atgc-montpellier.fr/phylnet/programs.

La nature du problème résolu dans l'article est aussi décrite par des mots-clés ("reconstruction", "consensus", "comparison", "generation", "visualization", etc) tout comme sa

complexité ("NP-complete", "APX-hard", "polynomial") ou la nature des algorithmes proposés ("exponential algorithm", "FPT", "approximation", "heuristic").

Les restrictions, introduites à la section 1.4, sur le type de réseaux reconstruits ou traités, sont également indiquées de la même façon, par plus de 20 mots-clés.

Ainsi, sans atteindre la complexité de conception d'un système comme l'interface web TreeTapper [3] de Brian O'Mara pour l'aide à la découverte et l'identification de besoins d'outils phylogénétiques, mais en donnant plus d'informations que le site de Joe Felsenstein sur les logiciels existants en phylogénie [4], cette bibliographie interactive aide les utilisateurs et les concepteurs d'outils sur les réseaux phylogénétiques à mieux connaître les outils et méthodes existantes. Elle a été bien reçue par la communauté, citée dans plusieurs publications (par exemple, [To et Habib, 2009; van Iersel *et al.*, 2010a; Huber *et al.*, 2010; Huson *et al.*, 2011]) et elle reçoit en moyenne 250 visites par mois depuis son lancement en décembre 2007, depuis une centaine de pays, au premier rang desquels Etats-Unis, France, Allemagne, Royaume-Uni, Nouvelle-Zélande, Canada, Brésil et Pays-Bas.

4.1.3 Problème de choix des gènes et des espèces dans un phylome

Parmi toutes les méthodes permettant de reconstruire des réseaux phylogénétiques, les méthodes combinatoires qui constituent le coeur de notre étude nécessitent, nous l'avons vu en section 3.1.2, d'avoir à disposition des données suffisamment complètes. Ainsi, nous abordons dans cette section la façon de choisir les arbres de gènes à utiliser en entrée des méthodes proposées dans la partie I. Nous évoquerons spécifiquement les méthodes qui fonctionnent sur des données enracinées (clades ou triplets), mais les solutions proposées se généralisent directement au cas non enraciné.

a) Méthodes à partir de clades

En tant qu'objet biologique, un clade C n'a pas d'existence propre, il n'est défini que par rapport à un arbre, sur un ensemble de taxons X, qui le représente sous la forme d'un sous-arbre. Ainsi, la donnée de l'ensemble X est indispensable lorsqu'on évoque le clade C. Toute méthode de reconstruction à base de clades considère donc implicitement un ensemble de taxons qui est le même pour tous les clades.

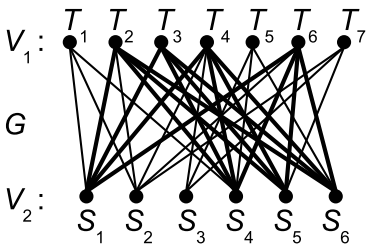
Ainsi, si les arbres disponibles dans les bases de données ne concernent pas les mêmes ensembles de taxons, une façon d'obtenir les données les plus complètes possibles - sans utiliser les techniques d'inférence de données évoquées en section 3.1.2 - est de trouver un sous-ensemble T d'arbres qui contiennent tous un même ensemble S de taxons. Il s'agit alors d'avoir des tailles assez grandes pour T et S. Ceci est donc un problème d'optimisation à deux critères, $t = |T|$ et $s = |S|$.

3. http://www.treetapper.org
4. http://evolution.genetics.washington.edu/phylip/software.html

En considérant la matrice de présence/absence des taxons dans les arbres présents dans les données montrée en figure 4.2(a), on peut proposer une expression plus formelle de ce problème. Soit M la matrice binaire à t lignes et s colonnes telle que $M_{i,j} = 1$ si l'arbre i contient le taxon j, et $M_{i,j} = 0$ sinon. Alors on cherche à obtenir une sous-matrice de M constituée uniquement de 1, de taille aussi grande que possible (en réorganisant éventuellement les lignes et colonnes).

	T_1	T_2	T_3	T_4	T_5	T_6	T_7
S_1	1	**1**	**1**	**1**	0	**1**	0
S_2	1	1	0	1	0	1	1
S_3	0	0	0	1	1	0	1
S_4	1	**1**	**1**	**1**	1	**1**	0
S_5	0	**1**	**1**	**1**	0	**1**	1
S_6	0	**1**	**1**	**1**	1	**1**	0

(a)



(b)

FIGURE 4.2 : La matrice M de présence/absence d'un ensemble de taxons $S_1 \dots S_6$ dans un ensemble d'arbres de gènes $T_1 \dots T_7$ (a) et le graphe biparti G qui la représente (b). La sous-matrice d'aire maximale de M, qui correspond à la biclique au plus grand nombre d'arêtes de G, est indiquée en gras.

On peut exprimer cette taille de plusieurs manières :
(i) soit par son périmètre, c'est-à-dire qu'on cherche à maximiser $t + s$,
(ii) soit par sa surface, auquel cas on veut maximiser $t \times s$.

Ce problème peut aussi être modélisé sous forme d'un problème de graphes, en considérant la matrice binaire comme la matrice d'adjacence d'un graphe biparti $G = (V_1 \cup V_2, E)$, illustré en figure 4.2(b), où V_1 représente l'ensemble des arbres et V_2 l'ensemble des taxons, et où E code la présence des taxons dans les arbres. Il s'agit alors de trouver une biclique comme sous-graphe induit, qui maximise :
(i) le nombre de sommets (t sommets dans V_1 et s dans V_2), il s'agit alors du problème MAXIMUM VERTEX BICLIQUE
(ii) le nombre d'arêtes, il s'agit alors du problème MAXIMUM EDGE BICLIQUE

Le problème MAXIMUM EDGE BICLIQUE est NP-complet [Peeters, 2003]. Si on ajoute la contrainte que la biclique est équilibrée, c'est-à-dire que $t = s$, le problème MAXIMUM VERTEX BICLIQUE, appelé BALANCED BICLIQUE est aussi NP-complet [Johnson, 1987].

En revanche, ce problème MAXIMUM VERTEX BICLIQUE sans la contrainte $t = s$ peut être résolu en temps polynomial grâce à un algorithme de recherche d'un **couplage maximum** (c'est-à-dire un ensemble, de taille maximale, d'arêtes qui ne partagent aucun sommet) dans un graphe biparti [Hopcroft et Karp, 1973], de la façon suivante :

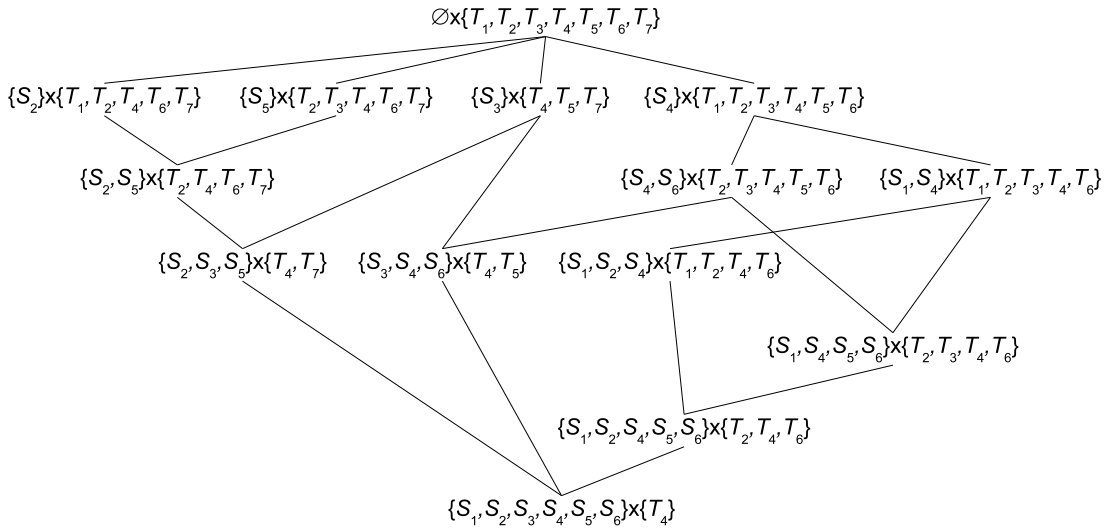


FIGURE 4.3 : Diagramme de Hasse des bicliques maximales du graphe G de la figure 4.2(b) pour l'inclusion de leurs sommets dans V_2.

- la biclique ayant le nombre maximum de sommets dans $G = (V_1 \cup V_2, E)$ est égale à un **stable maximum** (c'est-à-dire l'ensemble $V' \subset V_1 \cup V_2$ de sommets de taille maximale pour lequel $G'[V']$ est un stable) dans le graphe biparti $G' = (V_1 \cup V_2, E')$, où $E' = \{xy \mid x \in V_1, y \in V_2\} - E$,
- comme dans tout graphe, un stable maximum de G' est le complémentaire d'une **couverture de sommets minimum** de G' (c'est-à-dire l'ensemble de sommets de taille minimale qui, pour tout arête de E', contient un de ses deux sommets incidents),
- comme G' est biparti, le Théorème de König [Kőnig, 1931] permet d'y trouver une couverture de sommets minimum si l'on connaît un couplage maximum.

De plus, des approches exactes ont été conçues pour résoudre le problème MAXIMUM EDGE BICLIQUE en toute généralité. Elles sont fondées sur la construction de l'ensemble des bicliques maximales [Guénoche, 1990] (dont le diagramme de Hasse des sommets de V_2 est représenté en figure 4.3) et sont efficaces en pratique [Sanderson *et al.*, 2003; Makino et Uno, 2004].

On peut également envisager d'utiliser les diverses clôtures présentées en section 3.1.2 sur des données de bonne qualité, c'est-à-dire des "rectangles" de la matrice M ne contenant presque que des 1. Des approches combinatoires existent aussi pour résoudre ce problème de recherche de quasi-bicliques [Yan *et al.*, 2005].

b) Méthodes à partir de triplets

Pour les méthodes à base de triplets, il n'est pas nécessaire que ces données proviennent d'arbres contenant les mêmes taxons. En effet, la contrainte sur l'ensemble des triplets est qu'il soit dense. Or il est possible d'obtenir un ensemble dense de triplets à partir d'arbres ne portant pas sur les mêmes ensembles de taxons, comme montré dans l'exemple de la figure 4.4.

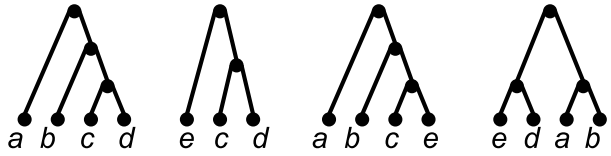


FIGURE 4.4 : Quatre arbres dont l'ensemble total de triplets est dense sur $\{a, b, c, d, e\}$.

Nous sommes alors confrontés à un nouveau problème combinatoire. Ce problème est équivalent à la généralisation du problème de la clique dans les hypergraphes 3-uniformes (c'est-à-dire une généralisation des graphes où ce ne sont pas deux sommets qui sont mis en relation par une arête, mais trois sommets mis en relation par une hyper-arête). Le fait que ce problème soit NP-complet est considéré comme folklorique [Dell et van Melkebeek, 2010] : nous écrivons ci-dessous une preuve simple de ce résultat, formulé en termes de triplets, qui assure aussi la W[1]-complétude du problème.

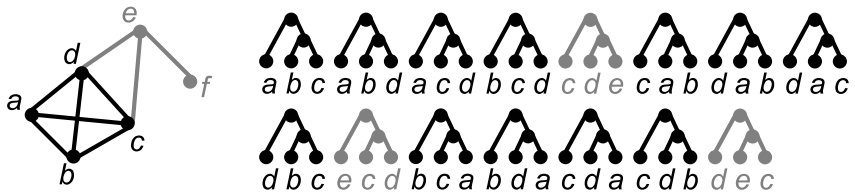
Problème 5 (MAXIMUM DENSE TRIPLET SET)
Entrée : *un ensemble $\mathcal{R}$ de triplets, et un entier* k.
Sortie : *OUI s'il existe un ensemble de feuilles $S \subseteq X$, de taille* k*, tel que $\mathcal{R}_{|S}$ est dense, NON sinon.*

Théorème 17 *Le problème* MAXIMUM DENSE TRIPLET SET *est NP-complet.*

Démonstration. On procède par réduction du problème MAXIMUM CLIQUE, qui consiste à déterminer s'il existe une clique de G de taille k. Soit un entier $k > 2$. Étant donné un graphe G, on construit l'ensemble de triplets $\mathcal{R}$ suivants : pour toute 3-clique $\{a, b, c\}$ de G, on ajoute les triplets $a|bc$, $b|ac$ et $c|ab$ à $\mathcal{R}$, comme illustré en figure 4.5.

S'il existe un ensemble S de taille k tel que $\mathcal{R}_{|S}$ est dense, alors pour tous $a \neq b \in S$, il existe une feuille $c \in S - \{a, b\}$ telle que $c|ab \in \mathcal{R}$, et donc $\{a, b, c\}$ est une 3-clique de G et en particulier a et b sont adjacents dans G. Finalement, S est donc une clique de G de taille k.

Inversement, s'il existe dans G une clique S de taille k, alors $\mathcal{R}_{|S}$ est l'ensemble de tous les triplets possibles sur S, et en particulier $\mathcal{R}_{|S}$ est dense.



(i) (ii)

FIGURE 4.5 : Réduction de MAXIMUM CLIQUE à MAXIMUM DENSE TRIPLET SET : une instance G de MAXIMUM CLIQUE (i) qui contient une clique $\{a, b, c, d\}$ à laquelle on associe un ensemble de triplets $\mathcal{R}$ (ii) tel que $\mathcal{R}_{|\{a,b,c,d\}}$, l'ensemble de triplets noirs, est dense.

De plus, vérifier qu'un ensemble de triplets sur k feuilles est dense s'effectue en temps polynomial en $O(k^3)$. Ainsi le problème est NP-complet. □

Cette réduction, qui "préserve le paramètre" k [Downey et Fellows, 1999], prouve également que le problème MAXIMUM DENSE TRIPLET SET est **W[1]-difficile** car MAXIMUM CLIQUE est W[1]-difficile, il est donc improbable qu'un algorithme en temps $O(f(k) \cdot \mathrm{poly}(n))$ puisse le résoudre.

On peut toutefois concevoir un algorithme exact incrémental pour le résoudre, ou une heuristique gloutonne qui, après une initialisation avec tous les triplets possibles, augmente progressivement l'ensemble S_{temp} de feuilles sur lesquelles les arbres en entrée contiennent un ensemble dense de triplets, en choisissant de préférence une nouvelle feuille qui maximise le nombre d'arbres contenant au moins trois feuilles de S_{temp}.

Nous allons maintenant voir comment intégrer ces algorithmes dans une interface pratique de visualisation de données d'arbres pour une sélection assistée par ordinateur de données pertinentes.

4.1.4 Interface de sélection semi-automatique d'arbres et d'espèces

Nous l'avons vu, le choix d'un ensemble d'arbres (qui correspondent chacun à une famille de gènes homologues) et d'espèces à fournir en entrée des algorithmes combinatoires de reconstruction de réseaux est un problème théoriquement difficile si on veut trouver de manière exacte des ensembles optimaux. De plus, l'utilisateur de méthodes de reconstruction phylogénétique exige un certain contrôle sur les données qu'il souhaite intégrer dans son analyse, et ne se satisfera peut-être pas des données sélectionnées automatiquement.

Ainsi, nous proposons dans cette section une interface aidant l'utilisateur à choisir un ensemble d'arbres et d'espèces qui lui conviennent et qui respectent les contraintes imposées par les méthodes de reconstruction de réseaux. La solution que nous proposons ici d'une interface de sélection semi-automatique permet également de pallier la complexité

théorique du problème : par ses interactions, l'utilisateur ajoute des contraintes qui réduisent l'espace de recherche.

Cette interface, appelée HeurisTree, est actuellement en cours d'implémentation. Elle est basée sur la visualisation en nuage arboré des données en entrée.

a) Le nuage arboré comme heuristique de sélection

Le **nuage arboré** [Gambette et Véronis, 2010] est une méthode de visualisation dont le principe est de superposer sur un même dessin des informations d'occurrences et de cooccurrences des données. Il généralise ainsi le **nuage de mots**, qui représente uniquement les informations d'occurrences des données, en faisant varier la taille des mots qui les représentent (voir à ce sujet [Viégas et Wattenberg, 2008]).

Plus concrètement, le nuage arboré consiste à placer autour d'un arbre des mots dont la taille reflète le nombre d'occurrences, et la distance dans l'arbre dépend de la cooccurrence dans les données. Dans l'application qui nous intéresse, on l'utilise pour représenter les espèces d'un ensemble d'arbres de gènes, fourni en entrée sous forme d'un tableau binaire d'arbres en lignes et d'espèces en colonnes, comme illustré dans le cadre en haut à gauche de la figure 4.6 :

- la taille du nom de l'espèce dépend du nombre d'arbres de gènes qui la contiennent, c'est-à-dire du nombre de 1 dans sa colonne,
- la distance entre deux espèces dépend du nombre d'arbres de gènes qui les contiennent toutes les deux, c'est-à-dire du nombre de lignes où les deux colonnes qui leur correspondent sont simultanément égales à 1.

Notons que la topologie de l'arbre de ce nuage arboré des espèces, construit uniquement à partir de ce tableau de présence/absence des gènes dans les espèces, est très proche de celle de la phylogénie des espèces construites à partir de l'ensemble des séquences de gènes. L'utilisation de la matrice de présence/absence des gènes dans les espèces constitue en effet une source d'informations intéressante à elle seule pour la reconstruction phylogénétique [Spencer *et al.*, 2007] et la découverte de transferts horizontaux [Halary *et al.*, 2010].

Inversement, on peut aussi représenter les arbres de gènes (désignés par le code de la famille de gènes qu'ils représentent) dans un nuage arboré qui représente leur proximité en fonction du nombre d'espèces qu'ils ont en commun. Un tel nuage est visible dans le cadre en haut à droite de la figure 4.6.

Rappelons que cette visualisation est initialement apparue sur le blog de Jean Véronis en décembre 2007 [5], pour visualiser les mots d'un texte. Dans ce contexte, elle consiste à placer les mots les plus fréquents du texte, dont la taille représente la fréquence, autour d'un arbre qui représente au mieux leur distance sémantique calculée à partir de leur cooccurrence dans le texte, c'est-à-dire le nombre de fois qu'ils apparaissent dans une même

5. http://blog.veronis.fr

fenêtre de n mots. Ainsi, les nuages arborés trouvent leur utilité au sein d'une démarche d'analyse statistique de textes et ont donc également des applications en sciences humaines. Ils permettent alors de susciter, formaliser et étayer des hypothèses de travail sur des textes, de les comparer selon leur représentation arborée, de hiérarchiser l'utilisation d'autres outils textométriques, et enfin de représenter les résultats de l'analyse [Amstutz et Gambette, 2010].

b) Description de l'interface HeurisTree

Décrivons plus en détails le concept d'HeurisTree, qui pourrait avoir l'apparence montrée en figure 4.6, et la façon dont les nuages arborés y sont utilisés.

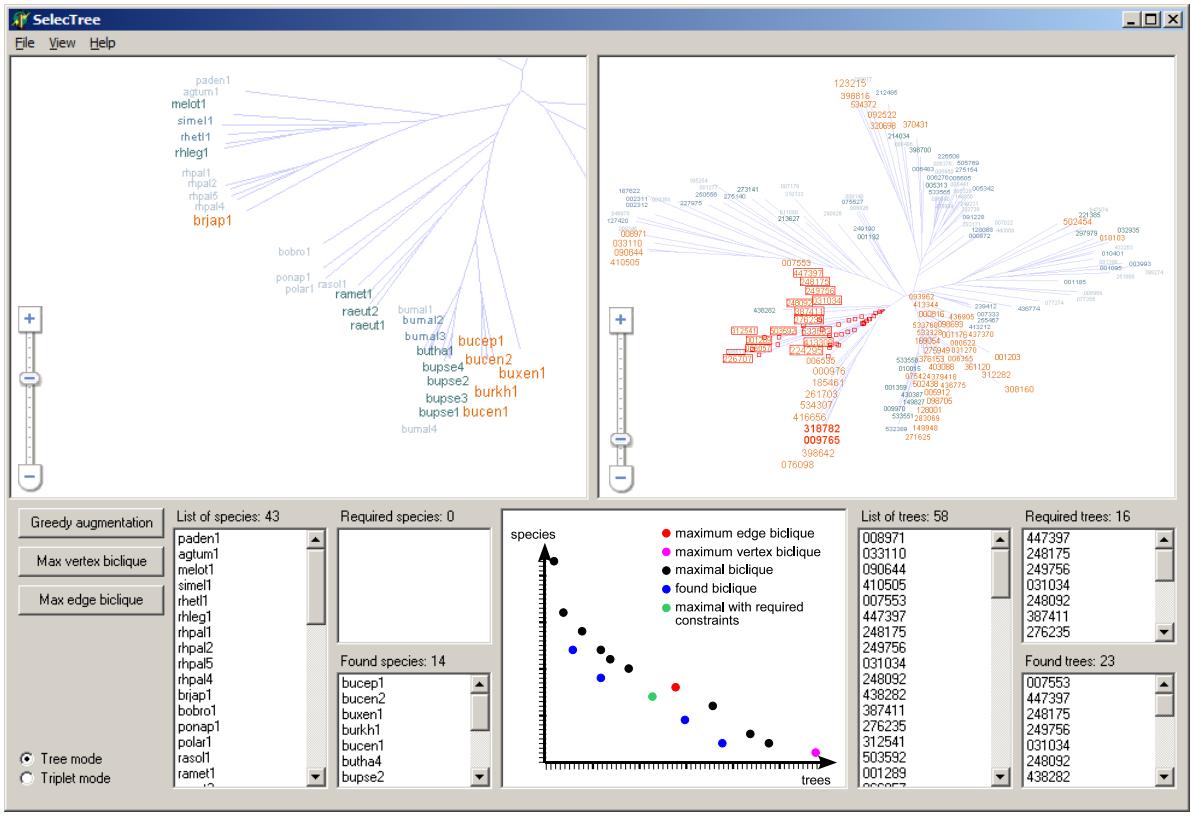


FIGURE 4.6 : Exemple d'interface possible pour Heuristree.

Après un chargement de la liste de tous les arbres de gènes du phylome au format Newick, HeurisTree calcule le nuage arboré des espèces, à gauche, et celui des arbres, à droite. La liste des espèces, et celle des arbres apparaissent sous le nuage arboré qui est associé à chacune.

On peut choisir de forcer le logiciel à faire apparaître des espèces ou des arbres dans la solution en faisant glisser les espèces voulues, depuis la liste ou le nuage arboré, vers la liste des “required species” ou “required trees”. La clôture des arbres (étant donné un ensemble T d'arbres, l'ensemble des arbres contenant toutes les espèces qui sont contenues dans tous les arbres de T) et la clôture des espèces (étant donné un ensemble S d'espèces, l'ensemble des espèces contenues dans tous les arbres qui contiennent les espèces de S) sont alors mises à jour dans le cadre en-dessous (“Found species” et “Found trees”), et un point bleu correspondant est ajouté au graphique récapitulatif.

Ce graphique représente toutes les bicliques trouvées sous forme d'un point ayant comme abscisse son nombre d'arbres et comme ordonnée son nombre d'espèces. Double-cliquer sur un point a pour effet de récupérer en mémoire les arbres et espèces auxquels il correspond pour les afficher en surbrillance dans le nuage arboré et dans les listes de “required species” et “required trees”.

Les boutons en bas à gauche permettent de lancer des algorithmes dont l'objectif est de trouver des bicliques maximales. Les points trouvés sont ajoutés au graphique récapitulatif alors que l'utilisateur garde la main et peut en parallèle tenter de trouver manuellement une meilleure solution.

Les menus peuvent être utilisés pour sauvegarder les listes de taxons, d'espèces, ou les bicliques maximales trouvées. Ils permettent également de choisir les paramètres voulus pour la construction des nuages arborés.

Un mode “triplets” est également disponible. Activable par le bouton radio en bas à gauche, il permet de vérifier si les arbres fournis contiennent un ensemble dense de triplets sur l'ensemble d'espèces éventuellement sélectionné, ou bien de trouver un tel ensemble d'espèces, de grande taille, par un algorithme glouton.

4.2 Exemples sur des données réelles

4.2.1 Outils utilisés

Le logiciel Dendroscope [Huson *et al.*, 2007] permet de représenter des arbres et réseaux phylogénétiques. Écrit en Java, il est multiplateforme, ergonomique, et permet de dessiner des arbres avec plusieurs milliers de taxons. Des outils permettent de manipuler facilement les dessins créés, comme l'outil de miroir de sous-arbres qui aide à régler l'ordre des taxons pour comparer plus facilement deux arbres ou réseaux.

De plus, il est possible d'appeler Dendroscope en ligne de commande, et d'utiliser son langage de scripts pour colorer automatiquement les noms des feuilles d'un arbre selon des classes de couleur [6].

Dans Dendroscope, trois méthodes de reconstruction de réseaux phylogénétiques enracinés à partir de clades sont actuellement implémentées : celle des réseaux de clades stricts

6. Plus de détails sur http://colorationdendroscope.gambette.com.

("Cluster network") [Huson et Rupp, 2008], des réseaux à une couche de réticulation ("Galled network", dont nous avons implémenté l'algorithme de résolution du problème MCS-r) [Huson *et al.*, 2009], et des réseaux de niveau k ("Minimum network") [van Iersel *et al.*, 2010a]. Elles demandent de charger un ensemble d'arbres enracinés en entrée, de préférence portant sur le même ensemble de taxons, car dans le cas contraire c'est la Z-clôture (avec les inconvénients que nous avons cités plus haut) qui est appliquée.

Les arbres de gènes disponibles pouvant contenir des taxons répétés à cause des duplications, on les filtre pour ne garder que les arbres qui ne contiennent aucun taxon avec doublon. On effectue alors la sélection des arbres de gènes, soit en fonction des gènes d'intérêt, soit en fonction d'espèces d'intérêt, comme présenté en section 4.1.3. Dernière étape de la préparation des données, le script restrictToCommonLeavesLinux de Celine Scornavacca [7] permet alors d'élaguer les arbres pour que tous contiennent le même ensemble de taxons.

Ces arbres peuvent alors être chargés dans Dendroscope, où il est possible de les dessiner, colorer, et surtout de reconstruire des réseaux à partir de leurs clades avec les trois méthodes citées ci-dessus. Un aperçu de l'interface d'utilisation du logiciel, pratique et intuitive, est donné en figure 4.7. Il est prévu que les prochaines versions de Dendroscope intègrent certaines des méthodes de reconstruction à partir de triplets présentées en section 2.1.1. En attendant, il est possible d'utiliser en ligne de commande les logiciels existants, Simplistic [van Iersel et Kelk, 2010] et Lev1athan [Huber *et al.*, 2010] en Java. Ils fournissent des réseaux au format eNewick [Cardona *et al.*, 2008b] et dans le langage DOT, qui peuvent donc être visualisés respectivement avec les logiciels Dendroscope et GraphViz [8].

Le logiciel SplitsTree propose également de reconstruire des réseaux abstraits non enracinés à partir d'arbres ayant le même ensemble de clades et sans doublons, avec la méthode des réseaux de consensus évoquée en section 2.1.1.

4.2.2 Utilisation sur les données HOGENOM

En attendant la finalisation du logiciel HeurisTree, nous avons utilisé un prototype d'implémentation de l'heuristique du nuage arboré, basé sur SplitsTree [Huson et Bryant, 2006] et TreeCloud [Gambette et Véronis, 2010], pour sélectionner un ensemble de 47 taxons, présentés dans la table 4.2, sur lesquels 16 arbres de gènes étaient présents dans la base de données Hogenom [Dufayard *et al.*, 2005]. Notons que comme chacun de ces arbres contient chacun des 47 taxons, l'ensemble des triplets extraits de ces arbres est dense. Parmi ces taxons, on remarquera quelques espèces de bactéries connues du grand public, comme des salmonelles (SATYI1, responsable de la thyphoïde, ou SATYP1, SAENT2...) ou le bacille de la peste (YEPES1-5). Les données correspondant à l'ensemble des résultats et figures présentés ci-dessous sont disponibles sur http://hogenom.

7. http://www.lirmm.fr/~scornava/Software.html
8. logiciel libre disponible sur http://www.graphviz.org.

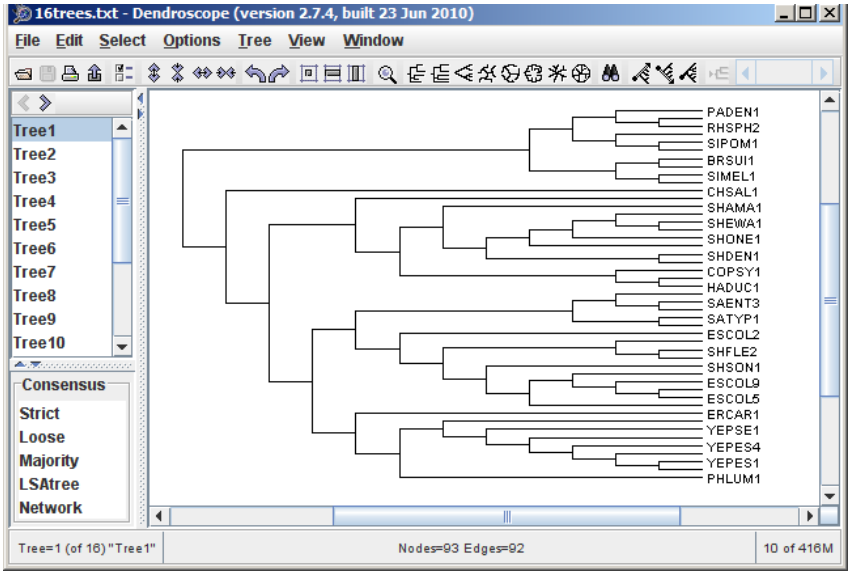


FIGURE 4.7 : Aperçu de l'interface graphique de Dendroscope.

Méthode	Figure	Temps de calcul (secondes)
Lev1athan Level-1 network	4.9(a)	10
Lev1athan Level-1 network	4.9(b)	24
Simplistic	4.9(c)	63
Simplistic	4.9(d)	32
SplitsTree Consensus network	4.10(a)	3
Dendroscope Cluster network	4.10(b)	<1
Dendroscope Galled network	4.10(c)	<1
Dendroscope Minimum network	4.10(d)	2

TABLE 4.1 : Temps de calcul des méthodes de Dendroscope, SplitsTree, Simplistic, Leav1athan, pour construire les réseaux des figures 4.9 et 4.10.

gambette.com, et les temps de calcul nécessaires pour obtenir un réseau à partir des 16 arbres sélectionnés sont indiqués dans la table 4.1.

Présentons tout d'abord les résultats obtenus avec des méthodes de triplets. Le réseau de niveau 1 de la figure 4.8 a été obtenu à partir des triplets contenus dans au moins 20% des 16 arbres extraits des données d'Hogenom. Le logiciel Lev1athan évoqué en section 2.1.1 permet de reconstruire des réseaux de niveau 1 contenant un grand nombre des triplets fournis en entrée. Les taux de support des triplets sont indiqués dans la visualisation fournie par le logiciel GraphViz : chaque taxon est étiqueté par le taux de triplets qui sont contenus dans le réseau, parmi ceux fournis en entrée qui le contiennent, et chaque arc est étiqueté par le nombre de triplets avec lesquels il est cohérent.

Abréviation	Souche	Ordre	Classe
SHDYS1	shigella dysenteriae sd197	enterobacteriales	gammaproteobacteria
SHSON1	shigella sonnei ss046	enterobacteriales	gammaproteobacteria
SHFLE1	shigella flexneri 2a str. 2457t	enterobacteriales	gammaproteobacteria
SHFLE2	shigella flexneri 2a str. 301	enterobacteriales	gammaproteobacteria
SHFLE3	shigella flexneri 5 str. 8401	enterobacteriales	gammaproteobacteria
ESCOL2	escherichia coli k12 k12	enterobacteriales	gammaproteobacteria
ESCOL5	escherichia coli o6	enterobacteriales	gammaproteobacteria
ESCOL6	escherichia coli uti89	enterobacteriales	gammaproteobacteria
ESCOL7	escherichia coli w3110	enterobacteriales	gammaproteobacteria
ESCOL9	escherichia coli cft073	enterobacteriales	gammaproteobacteria
SATYI1	salmonella typhi	enterobacteriales	gammaproteobacteria
SATYP1	salmonella typhimurium lt2	enterobacteriales	gammaproteobacteria
SAENT2	salmonella enterica subsp. enterica serovar paratyphi a str. atcc 9150	enterobacteriales	gammaproteobacteria
SAENT3	salmonella enterica subsp. enterica serovar typhi str. ty2	enterobacteriales	gammaproteobacteria
SAENT4	salmonella enterica subsp. enterica serovar typhi str. ct18	enterobacteriales	gammaproteobacteria
YEENT1	yersinia enterocolitica subsp. enterocolitica 8081	enterobacteriales	gammaproteobacteria
YEPSE1	yersinia pseudotuberculosis ip 32953	enterobacteriales	gammaproteobacteria
YEPES5	yersinia pestis nepal516	enterobacteriales	gammaproteobacteria
YEPES4	yersinia pestis kim	enterobacteriales	gammaproteobacteria
YEPES3	yersinia pestis co92	enterobacteriales	gammaproteobacteria
YEPES2	yersinia pestis biovar microtus str. 91001	enterobacteriales	gammaproteobacteria
YEPES1	yersinia pestis antiqua	enterobacteriales	gammaproteobacteria
PHLUM1	photorhabdus luminescens subsp. laumondii tto1	enterobacteriales	gammaproteobacteria
ERCAR1	erwinia carotovora subsp. atroseptica scri1043	enterobacteriales	gammaproteobacteria
ACPLE1	actinobacillus pleuropneumoniae l20	pasteurellales	gammaproteobacteria
HADUC1	haemophilus ducreyi 35000hp	pasteurellales	gammaproteobacteria
AEHYD1	aeromonas hydrophila subsp. hydrophila atcc 7966	aeromonadales	gammaproteobacteria
COPSY1	colwellia psychrerythraea 34h	alteromonadales	gammaproteobacteria
SHEWA3	shewanella sp. ana-3	alteromonadales	gammaproteobacteria
SHEWA1	shewanella sp. mr-4	alteromonadales	gammaproteobacteria
SHEWA2	shewanella sp. mr-7	alteromonadales	gammaproteobacteria
SHONE1	shewanella oneidensis mr-1	alteromonadales	gammaproteobacteria
SHEWA4	shewanella sp. w3-18-1	alteromonadales	gammaproteobacteria
SHDEN1	shewanella denitrificans os217	alteromonadales	gammaproteobacteria
SHFRI1	shewanella frigidimarina ncimb 400	alteromonadales	gammaproteobacteria
SHAMA1	shewanella amazonensis sb2b	alteromonadales	gammaproteobacteria
CHSAL1	chromohalobacter salexigens dsm 3043	oceanospirillales	gammaproteobacteria
RODEN1	roseobacter denitrificans och 114	rhodobacterales	alphaproteobacteria
SIPOM1	silicibacter pomeroyi dss-3	rhodobacterales	alphaproteobacteria
SILIC1	silicibacter sp. tm1040	rhodobacterales	alphaproteobacteria
PADEN1	paracoccus denitrificans pd1222	rhodobacterales	alphaproteobacteria
RHSPH1	rhodobacter sphaeroides 2.4.1	rhodobacterales	alphaproteobacteria
RHSPH2	rhodobacter sphaeroides atcc 17029	rhodobacterales	alphaproteobacteria
BRSUI1	brucella suis 1330	rhizobiales	alphaproteobacteria
BRMEL1	brucella melitensis 16m	rhizobiales	alphaproteobacteria
MESOR1	mesorhizobium sp. bnc1	rhizobiales	alphaproteobacteria
SIMEL1	sinorhizobium meliloti 1021	rhizobiales	alphaproteobacteria

TABLE 4.2 : Espèces de protéobactéries sélectionnées pour la reconstruction de réseaux phylogénétiques.

Si le logiciel Dendroscope n'indique pas ces étiquettes de taux de support, il permet de représenter les résultats obtenus à partir des triplets de la même manière que ceux trouvés par les méthodes de clades, pour faciliter leur comparaison. Ainsi, le réseau de la figure 4.8 est également représenté, tel que visualisé par Dendroscope, en figure 4.9(b). C'est aussi le cas du réseau construit par Lev1athan sur l'ensemble de tous les triplets des arbres fournis en entrée, en figure 4.9(a), qui montre naturellement un nombre plus élevé de sommets hybrides, mais une structure globalement similaire. Le programme Simplistic permet quant à lui de reconstruire des réseaux de niveau supérieur contenant tous les triplets en entrée : on obtient alors un réseau de niveau 7 à partir de l'ensemble des triplets contenus dans au moins 20% des arbres en entrée (voir la figure 4.9(c)). Si l'on se restreint aux triplets apparaissant dans au moins un tiers des arbres en entrée, on obtient le réseau de la figure 4.9(d) qui a niveau 4, et dont la structure est plus proche de celle des autres méthodes.

Les résultats obtenus à partir des bipartitions et clades apparaissant dans plus de 20% des arbres de gènes (c'est-à-dire au moins quatre arbres parmi les seize) sont montrés en figure 4.10. On peut remarquer que les réseaux des figures 4.10(c) et 4.10(d) ont une structure très similaire. Ceci s'explique par le fait que le réseau de niveau minimum de la figure 4.10(d) est en fait un réseau à une couche de réticulations. La quasi-totalité des différences entre ces deux réseaux correspondent aux ambiguïtés présentées en section 3.3.2 : l'incertitude sur la position du sommet hybride dans les cycles à quatre sommmets. La seule exception sur cet exemple est la position en tant qu'enfant d'un sommet hybride de la feuille SIMEL1 dans le réseau de la figure 4.10(c) et de MESOR1 dans celui de la figure 4.10(d).

Une analyse des résultats obtenus, en interprétant les arcs d'hybridation comme de possibles transferts horizontaux, ainsi qu'une comparaison plus poussée avec les résultats d'autres méthodes, sont en cours, dans le cadre du projet ANR PhylAriane impliquant des biologistes [9]. Précisons toutefois que la tâche d'évaluation des méthodes de reconstruction de réseaux phylogénétiques est particulièrement délicate.

Tout d'abord, les approches proposées pour générer des réseaux phylogénétiques artificiels afin d'effectuer des études sur données simulées sont très diverses, et leur méthodologie d'utilisation (en particulier les réglages des différents paramètres) n'est pas toujours clairement précisée [Grassly et Rambaut, 1999; Morin et Moret, 2006; Buendia et Narasimhan, 2006; Galtier, 2007; Arenas et Posada, 2008]. Quant aux jeux de données réelles contenant des transferts de matériel génétique documentés, ils sont plus rares que les données de transferts verticaux.

D'autre part, même si on avait à disposition un réseau de référence décrivant des hybridations ou transferts horizontaux avérés, la conception d'une distance entre réseaux phylogénétiques, pour mesurer la qualité d'un réseau reconstruit par rapport à ce réseau de référence, n'est pas une tâche facile. En effet, plusieurs formules de distances ont été proposées [Moret *et al.*, 2004; Nakhleh, 2004; Cardona *et al.*, 2008a, 2009b,c,d; Nakhleh,

9. http://www.lirmm.fr/phylariane

2010; Cardona *et al.*, 2010], mais elles ne respectent la propriété de **séparation** (la distance entre deux réseaux est nulle si et seulement si ils sont isomorphes) que sur des classes restreintes de réseaux. Par ailleurs, une formule de distance a été proposée pour les réseaux phylogénétiques explicites enracinés [Cardona *et al.*, 2009c], mais il est probable qu'il soit impossible de la calculer en temps polynomial, même pour des réseaux sans fratrie hybride [Cardona *et al.*, 2009a] [10].

Ainsi, les approches d'évaluation proposées en pratique jusqu'à présent utilisent des scores de proximité qui ne sont pas nécessairement des distances, basés sur le nombre de bipartitions des arbres inclus dans le réseau [Woolley *et al.*, 2008], ou l'identification correcte de la cible des transferts horizontaux dans le contexte de la réconciliation d'un arbre d'espèces avec des arbres de gènes [Abby *et al.*, 2010]. Une comparaison de ces diverses mesures, et une meilleure connaissance de leurs propriétés, est donc nécessaire avant une évaluation rigoureuse des méthodes de reconstruction de réseaux phylogénétiques.

10. par réduction du problème général d'isomorphisme de graphes à la vérification que cette distance est nulle.

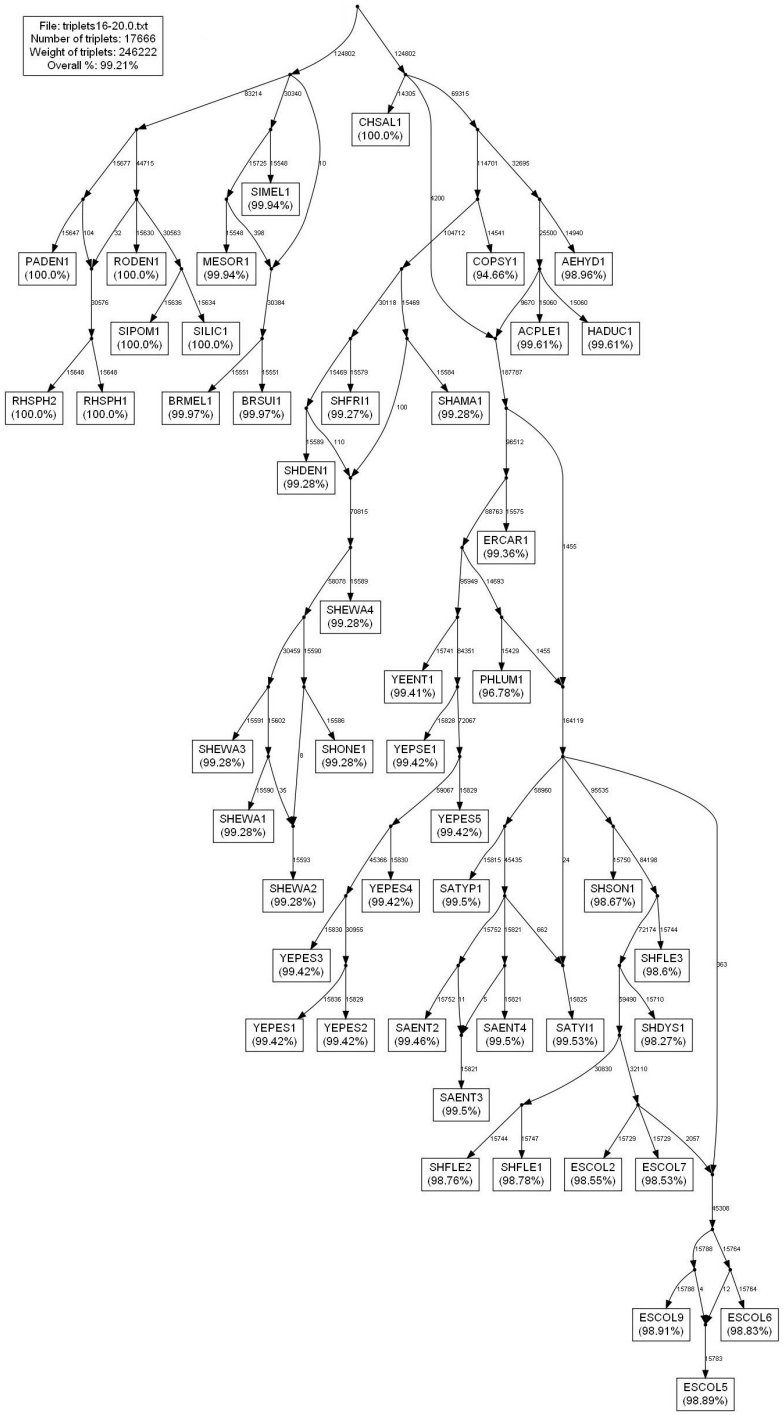


FIGURE 4.8 : Un réseau de niveau 1, reconstruit par Lev1athan à partir des triplets apparaissant dans au moins 20% des arbres sélectionnés, et visualisé avec GraphViz.

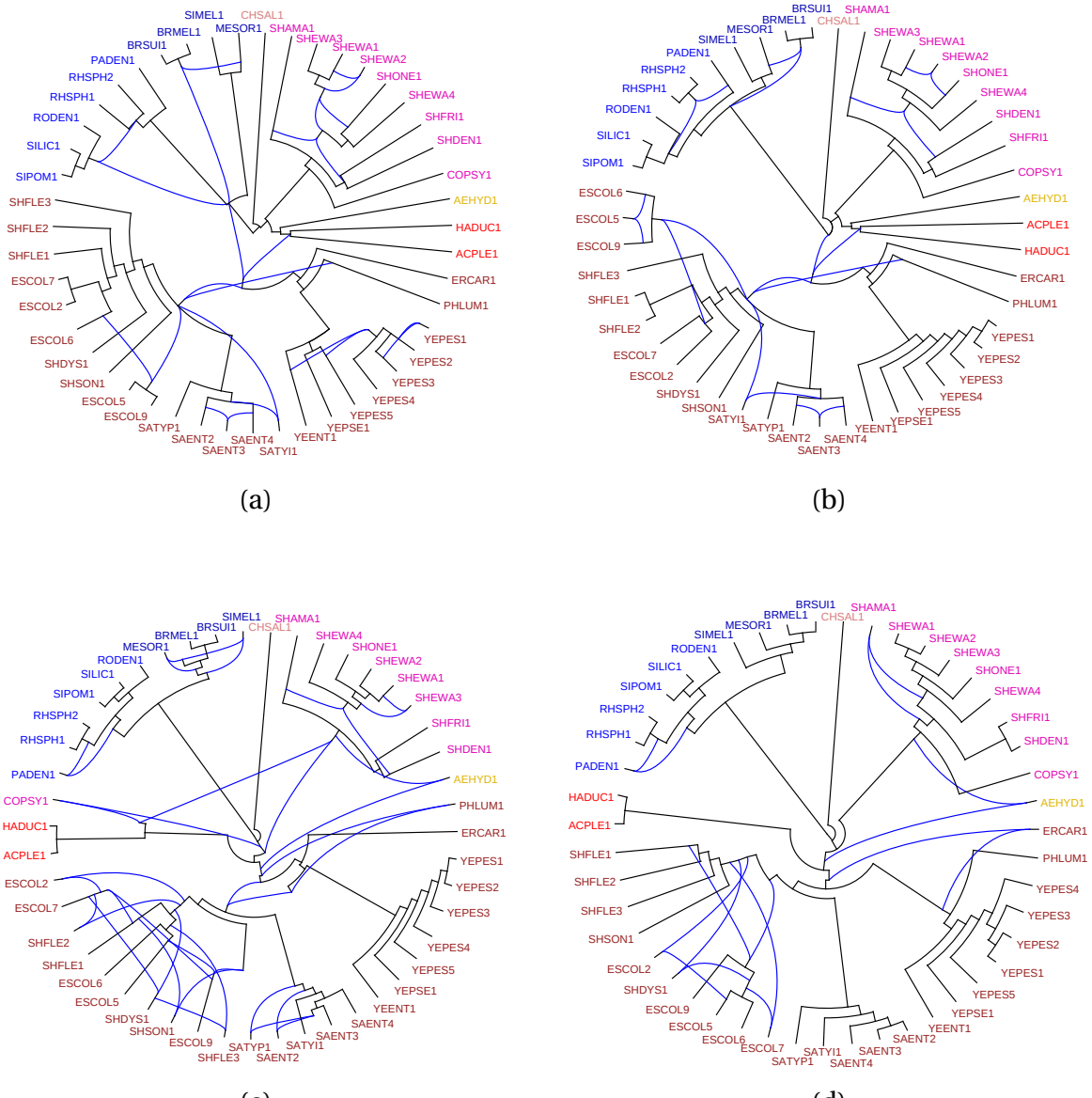


FIGURE 4.9 : Des réseaux de niveau 1 reconstruits par Lev1athan à partir de tous les triplets des 16 arbres sélectionnés (a), ou par des triplets apparaissant dans au moins 20% de ces arbres (b), visualisés avec Dendroscope pour permettre la comparaison avec les méthodes de clades de la figure 4.10. Des réseaux de niveau respectif 7 (c) et 4 (d) construits par Simplistic contenant tous les triplets apparaissant dans respectivement au moins 20% et au moins 33% des arbres.

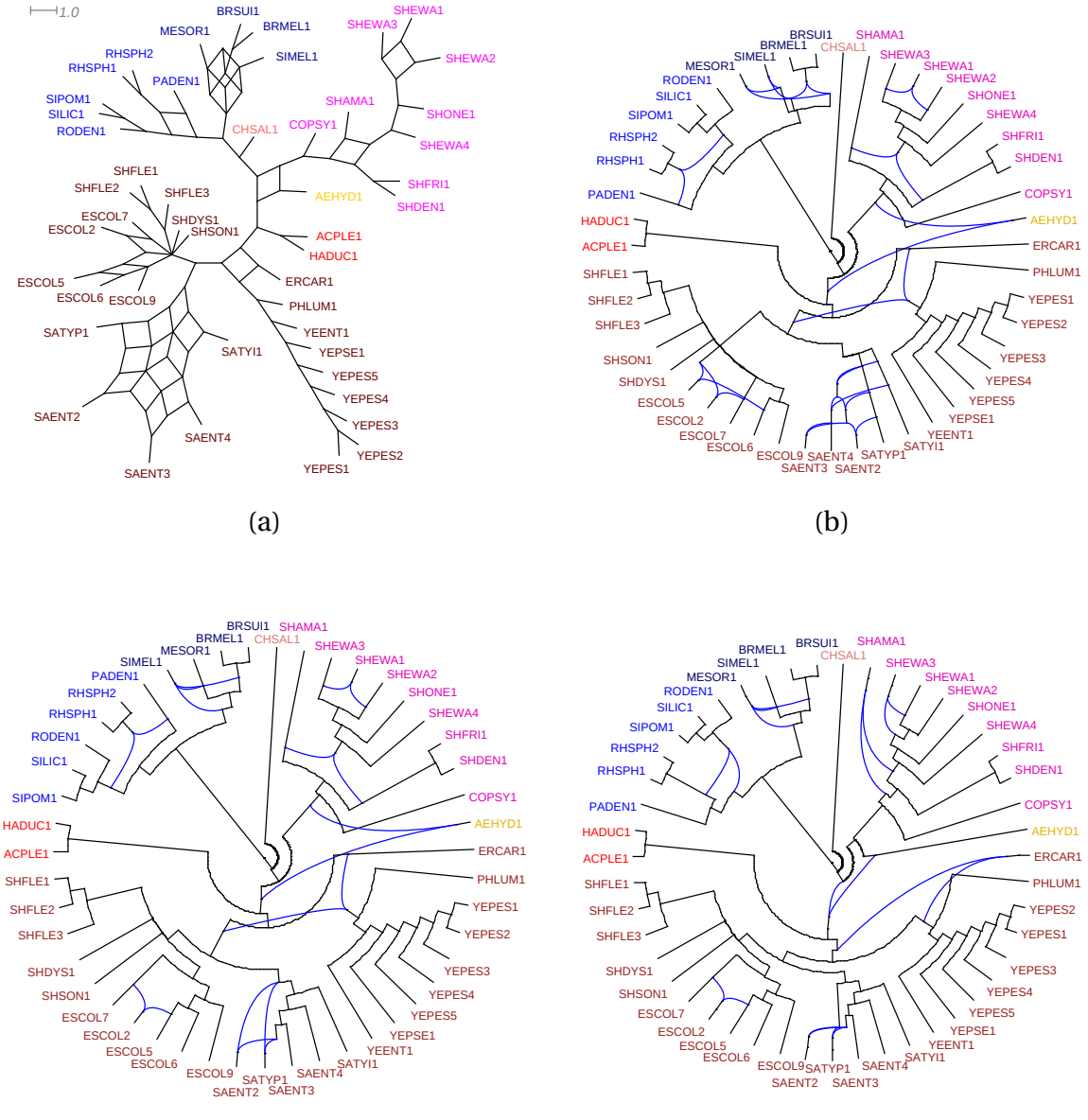


FIGURE 4.10 : Un réseau de consensus des bipartitions contenues dans 20% des 16 arbres sélectionnés (a), exemple de réseau phylogénétique abstrait non enraciné, reconstruit par SplitsTree. Des réseaux phylogénétiques explicites enracinés construits par Dendroscope à partir des clades contenus dans 20% des seize arbres : réseau de clades stricts (b), réseau à une couche de réticulation (c), réseau de niveau minimum (d).

Conclusion et perspectives

Problèmes ouverts

Avant de conclure sur les perspectives générales issues de ce travail sur la reconstruction de réseaux phylogénétiques, nous présentons quelques problèmes ouverts.

Les problèmes sur les quadruplets et les réseaux phylogénétiques explicites non orientés abordés en section 2.2 sont intéressants du point de vue de la théorie des graphes car ils montrent plus de symétrie que les triplets. Ils semblent donc plus directement reliés aux problèmes classiques de théorie des graphes, et pourraient être un moyen de mieux comprendre la combinatoire des triplets et des réseaux phylogénétiques explicites enracinés.

La complexité en temps du calcul de l'ensemble de tous les quadruplets contenus dans un réseau non enraciné de niveau k peut être améliorée, il devrait être possible d'obtenir la complexité optimale en $O(n^4)$.

Le problème ouvert le plus déconcertant sur les réseaux phylogénétiques non enracinés et les quadruplets est l'existence d'un algorithme polynomial pour déterminer s'il est possible de reconstruire un réseau simple non enraciné de niveau 1 qui contient un ensemble dense de quadruplets. Ce problème est similaire au problème NON-BETWEENNESS avec un ensemble dense de contraintes, dont la complexité est également inconnue.

Trouver un ensemble dense de quadruplets contenu dans un unique réseau non enraciné de niveau k, pour tout k, conduirait à une preuve de NP-complétude du problème LEVEL-k QUARTET CONSISTENCY, pour $k > 1$. De plus, l'approche pour le partitionnement d'un ensemble dense de triplets en différents blobs du réseau de niveau k à reconstruire [To et Habib, 2009] ne se traduit pas directement en termes de quadruplets. Ainsi, il faut adapter cette stratégie ou la modifier plus en profondeur, dans l'objectif de trouver un algorithme polynomial pour résoudre le problème LEVEL-k QUARTET CONSISTENCY pour un k fixé et un ensemble dense de quadruplets. La même remarque est valable pour les réseaux simples non enracinés de niveau k, où les algorithmes de reconstruction pour les triplets dans le contexte enraciné ne peuvent être adaptés.

Enfin, les résultats sur la structure des réseaux enracinés de niveau k de la section 1.4.3 pourraient conduire à des formules de dénombrement des réseaux de niveau k. Ces résultats de structure sont également valables pour les réseaux non enracinés, qui peuvent être décomposés en arbres non enracinés de générateurs non enracinés de niveau k. Ces générateurs peuvent être définis comme les multigraphes biconnexes 3-réguliers à $2k-2$ sommets. Les générateurs constituent une piste naturelle pour trouver un algorithme de complexité paramétrée en k pour la reconstruction de réseaux enracinés de niveau k depuis un ensemble dense de triplets [van Iersel et Kelk, 2009], et la même question se pose pour la reconstruction non enracinée de niveau k à partir d'un ensemble dense de quadruplets.

En ce qui concerne la méthode en deux étapes de reconstruction de réseaux à une couche de réticulations à partir de clades présentée en section 2.3, on peut envisager une amélioration et une évaluation fine de la complexité de l'algorithme de séparation et évaluation utilisé pour résoudre le problème MINIMUM ATTACHMENT.

De nouveaux résultats sur la caractérisation de certaines classes de réseaux à partir de leur ensemble de triplets, quadruplets ou clades constituent également un objectif intéressant pour mieux comprendre les structures sous-jacentes à ces objets. Ainsi, l'existence d'obstructions de taille finie ou de propriétés simples pour caractériser les ensembles de tous les triplets ou clades d'un réseau de niveau 1, ou de tous les quadruplets d'un réseau non enraciné de niveau 1, reste une question ouverte. Y répondre de manière positive conduirait à des algorithmes d'édition de triplets pour obtenir un réseau non enraciné de niveau 1, du type de celui présenté pour les arbres en section 3.1.1.

Enfin, un nouveau problème combinatoire, MAXIMUM DENSE TRIPLET SET, ayant été introduit en section 4.1.3, il serait intéressant de trouver des approches directes efficaces pour le résoudre.

Perspectives sur les méthodes combinatoires en phylogénie réticulée

Comme nous l'avons vu, cette thèse dédiée à une approche combinatoire de la reconstruction des réseaux phylogénétiques a permis de préciser les définitions et propriétés des objets mathématiques introduits dans la littérature, en synthétisant les relations connues et en présentant de nouvelles relations entre eux. Un panel de méthodes exactes et d'heuristiques d'optimisation souvent fondées sur des techniques d'algorithmique et de décomposition de graphes, auquel nous ajoutons deux propositions d'algorithmes de reconstruction à partir de clades ou à partir de quadruplets, fournissent généralement des résultats rapides à l'aide de logiciels faciles à manipuler, après un éventuel pré-traitement.

Toutefois nous avons observé et démontré au chapitre 3 certaines de leurs limites, qui nécessitent de préciser leur contexte d'utilisation : en raison du non-encodage de certains réseaux par leur ensemble de triplets ou de clades, il est déconseillé de considérer le réseau fourni par une méthode combinatoire comme un résultat absolument fiable reflétant de façon certaine des événements biologiques passés. Développer une visualisation des parties peu fiables du réseau, en proposant par exemple une coloration des arêtes peu soutenues par les données en entrée, ou bien en facilitant la visualisation, pour chaque arête du réseau, de l'ensemble des données en entrée qui expliquent sa présence permettrait de mettre en relief cette incertitude et les éventuelles ambiguïtés du réseau.

De plus, pour obtenir des réseaux phylogénétiques plus fiables, il semble nécessaire de passer par une démarche de validation statistique, en retournant aux données de séquences. Ainsi, les méthodes combinatoires citées ou proposées dans cette thèse nous semblent être plutôt des outils d'exploration des données, susceptibles d'indiquer une structure globale, et de fournir un ou des bons candidats pour la structure détaillée du réseau. La vraisemblance de la structure réticulée devrait ensuite être évaluée, avant de tenter diverses modifications du réseau afin d'améliorer ce score de vraisemblance.

On peut alors envisager qu'au-delà d'une utilisation comme outil exploratoire, nécessitant de comparer finement les résultats obtenus par plusieurs algorithmes, les méthodes combinatoires soient utilisées dans une démarche globale intégrant une évaluation statistique, comme un moyen de fournir un candidat de qualité. Elles peuvent également servir de guide pour trouver des modifications pertinentes de la structure (en incitant à respecter la structure arborée globale qui apparaît dans les données, par exemple), ou des données à considérer en entrée (en négligeant celles qu'on suppose correspondre à du bruit). Cette proposition de méthodologie est résumée dans la figure 4.11.

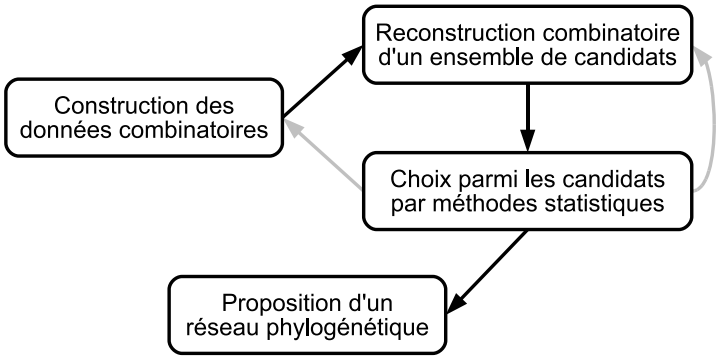


FIGURE 4.11 : Méthodologie de reconstruction de réseaux phylogénétiques fiables.

Le socle statistique d'une telle méthode, qui consisterait à calculer la vraisemblance des séquences de gènes observées en fonction de la topologie du réseau phylogénétique reconstruit reste toutefois à préciser. Quelques premières approches et modèles statistiques ont été proposés au cours de la dernière décennie [Strimmer et Moulton, 2000; Jin *et al.*, 2006; Snir et Tuller, 2009; Kubatko, 2009], mais demandent une évalution approfondie et une comparaison des modèles à l'aide de données réelles. Cette approche hybride mêlant combinatoire et statistiques est un des axes de recherche du projet PhylAriane, collaboration entre le LIRMM et l'ISEM de Montpellier, et le LBBE de Lyon.

Précisons également que pour identifier les transferts horizontaux et les recombinaisons, d'autres méthodes ignorent les données de phylogénie en se concentrant sur les séquences des gènes et d'éventuels biais de constitution en nucléotides [11] [Becq *et al.*, 2010]. Les résultats de ces méthodes semblent aussi présenter un intérêt en tant qu'indices supplémentaires pour valider des hypothèses de transferts horizontaux entre espèces.

Dans ce contexte, il serait également intéressant d'étudier plus en détails les "autoroutes de transferts" [Beiko *et al.*, 2005; Bansal *et al.*, 2011], qui correspondent au transfert de plusieurs gènes entre deux espèces. La question de la fonction des gènes ainsi partagés, et leurs interactions avec les gènes de l'organisme dans lequel ils ont été transférés constitue, elle aussi, une piste d'étude à suivre pour mieux comprendre les mécanismes et les apports de l'évolution "horizontale", par rapport à l'évolution "verticale" des génomes.

11. Une compilation de programmes dédiés à la détection de recombinaisons est disponible sur http://www.bioinf.manchester.ac.uk/recombination/programs.shtml

Annexes

Bibliographie

Sophie Abby, Eric Tannier, Manolo Gouy et Vincent Daubin : Detecting lateral gene transfers by statistical reconciliation of phylogenetic forests. *BMC Bioinformatics*, 11(324), 2010. http://dx.doi.org/10.1186/1471-2105-11-324. Cité pages 130 et 144.

Louigi Addario-Berry, Mike Hallett et Jens Lagergren : Towards identifying lateral gene transfer events. *In Proceedings of the eigth Pacific Symposium on Biocomputing (PSB'03)*, 2003. http://www.ncbi.nlm.nih.gov/pubmed/12603035. Cité page 62.

Alfred V. Aho, Yehoshua Sagiv, Thomas G. Szymanski et Jeffrey D. Ullman : Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. *SIAM Journal on Computing*, 10(3):405–421, 1981. http://dx.doi.org/10.1137/0210030. Cité pages 65 et 66.

Delphine Amstutz et Philippe Gambette : Utilisation de la visualisation en nuage arboré pour l'analyse littéraire. *In Proceedings of the tenth International Conference on statistical analysis of textual data (JADT'10)*, pages 227–238, 2010. http://www.ledonline.it/ledonline/jadt-2010.html. Cité page 138.

Miguel Arenas et David Posada : Recodon : Coalescent simulation of coding DNA sequences with recombination, migration and demography. *BMC Bioinformatics*, 8(458), 2008. http://dx.doi.org/10.1186/1471-2105-8-458. Cité pages 120 et 143.

Miguel Arenas, Gabriel Valiente et David Posada : Characterization of reticulate networks based on the coalescent with recombination. *Molecular Biology and Evolution*, 25:2517–2520, 2008. http://dx.doi.org/10.1093/molbev/msn219. Cité page 120.

Jean-Christophe Aude, Yolande Diaz-Lazcoz, Jean-Jacques Codani et Jean-Loup Risler : Applications of the pyramidal clustering method to biological objects. *Computers and Chemistry*, 23 (3-4):303–315, 1999. http://dx.doi.org/10.1016/S0097-8485(99)00006-6. Cité page 22.

Hans-Jürgen Bandelt : Four-point characterization of the dissimilarity functions obtained from indexed closed weak hierarchies, 1992. Mathematisches Seminar, Universität Hamburg. Cité pages 34, 35, 56 et 58.

Hans-Jürgen Bandelt et Andreas Dress : Reconstructing the shape of a tree from observed dissimilarity data. *Advances in Applied Mathematics*, 7:309–343, 1986. http://dx.doi.org/10.1016/0196-8858(86)90038-2. Cité page 106.

Hans-Jürgen Bandelt et Andreas W. M. Dress : Weak hierarchies associated with similarity measures : an additive clustering technique. *Bulletin of Mathematical Biology*, 51:113–166, 1989. http://dx.doi.org/10.1007/BF02458841. Cité pages 33 et 34.

Hans-Jürgen Bandelt et Andreas W. M. Dress : A canonical decomposition theory for metrics on a finite set. *Advances in Mathematics*, 92(1):47–105, 1992a. http://dx.doi.org/10.1016/0001-8708(92)90061-O. Cité pages 58 et 87.

Hans-Jürgen Bandelt et Andreas W. M. Dress : Split decomposition : A new and useful approach to phylogenetic analysis of distance data. *Molecular Phylogenetics and Evolution*, 1(3):242–252, 1992b. http://dx.doi.org/10.1016/1055-7903(92)90021-8. Cité page 35.

Hans-Jürgen Bandelt et Andreas W. M. Dress : An order theoretic framework for overlapping clustering. *Discrete Mathematics*, 136(1-3):21–37, 1994. http://dx.doi.org/10.1016/0012-365X(94)00105-R. Cité page 65.

Hans-Jürgen Bandelt, Peter Forster, Bryan C. Sykes et Martin B. Richards : Mitochondrial portraits of human population using median networks. *Genetics*, 141:743–753, 1995. http://www.genetics.org/cgi/content/abstract/141/2/743. Cité pages 22 et 35.

Mukul S. Bansal, J. Peter Gogarten et Ron Shamir : Detecting highways of horizontal gene transfer. *In Proceedings of the Eighth RECOMB Comparative Genomics Satellite Workshop (RECOMB-CG'10)*, volume 6398 de *Lecture Notes in Computer Science*, pages 109–120. Springer Verlag, 2011. http://dx.doi.org/10.1007/978-3-642-16181-0_10. Cité page 154.

Mihaela Baroni, Charles Semple et Mike Steel : A framework for representing reticulate evolution. *Annals of Combinatorics*, 8:398–401, 2004. http://dx.doi.org/10.1007/s00026-004-0228-0. Cité page 53.

Mihaela Baroni et Mike Steel : Accumulation phylogenies. *Annals of Combinatorics*, 10(1):19–30, 2006. http://dx.doi.org/10.1007/978-3-642-10631-6_121. Cité page 27.

Jean-Pierre Barthélemy, François Brucker et Christophe Osswald : Combinatorial optimization and hierarchical classifications. *4OR : A Quaterly Journal of Operations Research*, 2(3):179–219, 2004. http://dx.doi.org/10.1007/s10288-004-0051-9. Cité page 58.

Jean-Pierre Barthélemy et Alain Guénoche : *Les arbres et les représentations des proximités*. Masson, 1988. Cité pages 17 et 20.

André Batbedat : Les isomorphismes HTE et HTS, après la bijection de Benzécri-Johnson. *Metron*, 46:47–59, 1988. Cité page 34.

André Batbedat : Les dissimilarités Médas ou Arbas. *Statistique et analyse des données*, 14:1–18, 1989. Cité page 34.

Kenneth E. Batcher : Sorting networks and their applications. *In Proceedings of the AFIPS Spring Joint Computer Conference*, volume 32, pages 307–314, 1968. http://dx.doi.org/10.1145/1468075.1468121. Cité page 66.

Jennifer Becq, Cécile Churlaud et Patrick Deschavanne : A benchmark of parametric methods for horizontal transfers detection. *PLoS ONE*, 5(4):e9989, 2010. http://dx.doi.org/10.1371/journal.pone.0009989. Cité page 154.

Robert G. Beiko et Nicholas Hamilton : Phylogenetic identification of lateral genetic transfer events. *BMC Evolutionary Biology*, 6(15), 2006. http://dx.doi.org/10.1186/1471-2148-6-15. Cité page 63.

Robert G. Beiko, Timothy J. Harlow et Mark A. Ragan : Highways of gene sharing in prokaryotes. *Proceedings of the National Academy of Sciences*, 102(40):14332–14337, 2005. http://dx.doi.org/10.1073/pnas.0504068102. Cité page 154.

Vincent Berry et David Bryant : Faster reliable phylogenetic analysis. *In RECOMB99*, pages 59–68, 1999. http://dx.doi.org/10.1145/299432.299457. Cité page 65.

Vincent Berry et Olivier Gascuel : Inferring evolutionary trees with strong combinatorial evidence. *Theoretical Computer Science*, 240(2):271–298, 2000. http://dx.doi.org/10.1016/S0304-3975(99)00235-2. Cité page 74.

Vincent Berry, Tao Jiang, Paul E. Kearny, Ming Li et Todd Wareham : Quartet cleaning : Improved algorithms and simulations. *In Proceedings of the seventh Annual European Symposium on Algorithms (ESA'99)*, volume 1643 de *Lecture Notes in Computer Science*, pages 313–324. Springer Verlag, 1999. http://dx.doi.org/10.1007/3-540-48481-7_28. Cité page 106.

Vincent Berry et François Nicolas : Improved parameterized complexity of the maximum agreement subtree and maximum compatible tree problems. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 3(3):289–302, 2006. http://dx.doi.org/10.1109/TCBB.2006.39. Cité pages 68 et 91.

Patrice Bertrand et Melvin F. Janowitz : Pyramids and weak hierarchies in the ordinal model for clustering. *Discrete Applied Mathematics*, 122(1-3):55–81, 2002. http://dx.doi.org/10.1016/S0166-218X(01)00354-7. Cité page 35.

Olaf R.P. Bininda-Emonds, Robin M.D. Beck et Andy Purvis : Getting to the roots of matrix representation. *Systematic Biology*, 54(4):668–672, 2005. http://dx.doi.org/10.1080/10635150590947113. Cité page 69.

Alix Boc, Hervé Philippe et Vladimir Makarenkov : Inferring and validating horizontal gene transfer events using bipartition dissimilarity. *Systematic Biology*, 59(2):195–211, 2010. http://dx.doi.org/10.1093/sysbio/syp103. Cité page 63.

Sebastian Böcker, Quang Bao Anh Bui et Anke Truss : An improved fixed-parameter algorithm for minimum-flip consensus trees. *In Proceedings of the Third International Workshop on Parameterized and Exact Computation (IWPEC'08)*, volume 5018 de *Lecture Notes in Computer Science*, pages 43–54, 2008. http://dx.doi.org/10.1007/978-3-540-79723-4_6. Cité page 93.

Magnus Bordewich, Simone Linz, Katherine St. John et Charles Semple : A reduction algorithm for computing the hybridization number of two trees. *Evolutionary Bioinformatics*, 3:86–98, 2007. http://www.ncbi.nlm.nih.gov/pubmed/19461978. Cité page 62.

Magnus Bordewich et Charles Semple : Computing the hybridization number of two phylogenetic trees is fixed-parameter tractable. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 4:458–466, 2007a. http://dx.doi.org/10.1109/tcbb.2007.1019. Cité page 62.

Magnus Bordewich et Charles Semple : Computing the minimum number of hybridization events for a consistent evolutionary history. *Discrete Applied Mathematics*, 155:914–918, 2007b. http://dx.doi.org/10.1016/j.dam.2006.08.008. Cité page 62.

Ulrik Brandes et Sabine Cornelsen : Phylogenetic graph models beyond trees. *Discrete Applied Mathematics*, 157(10):2361–2369, 2009. http://dx.doi.org/10.1016/j.dam.2008.06.031. Cité page 28.

François Brucker : *Modèles de classification en classes empiétantes*. Thèse de doctorat, Ecole des Hautes Etudes en Sciences Sociales, France, 2001. http://francois.brucker.perso.centrale-marseille.fr/publications/these_francois_brucker.pdf. Cité page 34.

David Bryant : *Building Trees, Hunting for Trees, and Comparing Trees : theory and method in phylogenetic analysis*. Thèse de doctorat, Dept. Mathematics, University of Canterbury, New Zealand, 1997. http://citeseer.ist.psu.edu/685475.html. Cité pages 105 et 111.

David Bryant et Vincent Berry : A structured family of clustering and tree construction methods. *Advances in Applied Mathematics*, 27(4):705–732, 2001. http://dx.doi.org/10.1006/aama.2001.0758. Cité page 33.

Patricia Buendia et Giri Narasimhan : Serial NetEvolve : A flexible utility for generating serially-sampled sequences along a tree or recombinant network. *Bioinformatics*, 18(22):2313–2314, 2006. http://dx.doi.org/10.1093/bioinformatics/btl387. Cité page 143.

Peter Buneman : The recovery of trees from measures of dissimilarity. *In* F.R. Hodson, D.G. Kendall et P. Tautu, éditeurs : *Mathematics in Archeological and Historical Sciences*, pages 387–395. Edimburgh University Press, 1971. http://homepages.inf.ed.ac.uk/opb/homepagefiles/phylogeny-scans/manuscripts.pdf. Cité page 20.

Jaroslaw Byrka, Pawel Gawrychowski, Katharina T. Huber et Steven Kelk : Worst-case optimal approximation algorithms for maximizing triplet consistency within phylogenetic networks. *Journal of Discrete Algorithms*, 8(1):65–75, 2010. http://dx.doi.org/10.1016/j.jda.2009.01.004. Cité pages 66 et 69.

Gabriel Cardona, Mercè Llabrés, Francesc Rosselló et Gabriel Valiente : The comparison of tree-sibling time consistent phylogenetic networks is graph-isomorphism complete, 2009a. http://arxiv.org/abs/0902.4640. Cité page 144.

Gabriel Cardona, Mercè Llabrés, Francesc Rosselló et Gabriel Valiente : Metrics for phylogenetic networks II : Nodal and triplets metrics. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 6(3):454–469, 2009b. http://dx.doi.org/10.1109/TCBB.2008.127. Cité page 143.

Gabriel Cardona, Mercè Llabrés, Francesc Rosselló et Gabriel Valiente : On Nakhleh's metric for reduced phylogenetic networks. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 6(4):629–638, 2009c. http://dx.doi.org/10.1109/TCBB.2009.33. Cité pages 143 et 144.

Gabriel Cardona, Mercè Llabrés et Francesc Rosselló : A metric for galled networks. *In Jornadas de Bioinformática, Workshop on Bioinformatics for Personalized Medicine*, 2010. http://arxiv.org/abs/1009.0652. Cité page 144.

Gabriel Cardona, Mercè Llabrés, Francesc Rosselló et Gabriel Valiente : A distance metric for a class of tree-sibling phylogenetic networks. *Bioinformatics*, 24(13):1481–1488, 2008a. http://dx.doi.org/10.1093/bioinformatics/btn231. Cité page 143.

Gabriel Cardona, Mercè Llabrés, Francesc Rosselló et Gabriel Valiente : Metrics for phylogenetic networks I : Generalizations of the Robinson-Foulds metric. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 6(1):46–61, 2009d. http://dx.doi.org/10.1109/TCBB.2008.70. Cité page 143.

Gabriel Cardona, Francesc Rosselló et Gabriel Valiente : Extended newick : It is time for a standard representation. *BMC Bioinformatics*, 9(460), 2008b. http://dx.doi.org/10.1186/1471-2105-9-532. Cité page 140.

Gabriel Cardona, Francesc Rosselló et Gabriel Valiente : Tripartitions do not always discriminate phylogenetic networks. *Mathematical Biosciences*, 211(2), 2008c. http://dx.doi.org/10.1016/j.mbs.2007.11.003. Cité page 53.

Insa Cassens, Patrick Mardulyn et Michel C. Milinkovitch : Evaluating intraspecific network construction methods using simulated sequence data : Do existing algorithms outperform the global maximum parsimony approach ? *Systematic Biology*, 54(3):363–372, 2005. http://www.jstor.org/stable/20061240. Cité page 23.

Ho-Leung Chan, Jesper Jansson, Tak-Wah Lam et Siu-Ming Yiu : Reconstructing an ultrametric galled phylogenetic network from a distance matrix. *Journal of Bioinformatics and Computational Biology*, 4(4):807–832, 2006. http://dx.doi.org/10.1007/11549345_20. Cité page 54.

Maw-Shang Chang, Chuang-Chieh Lin et Peter Rossmanith : New fixed-parameter algorithms for the minimum quartet inconsistency problem. *Theory of Computing Systems*, 2(47):342–367, 2010. http://dx.doi.org/10.1007/s00224-009-9165-y. Cité page 106.

Pierre Charbit, Michel Habib, Vincent Limouzy, Fabien de Montgolfier, Mathieu Raffinot et Michaël Rao : A note on computing set overlap classes. *Information Processing Letters*, 108(4):186–191, 2008. http://dx.doi.org/10.1016/j.ipl.2008.05.005. Cité page 89.

Duhong Chen, Oliver Eulenstein, David Fernández-Baca et Michael Sanderson : Minimum-flip supertrees : Complexity and algorithms. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 3(2):165–173, 2006. http://dx.doi.org/10.1109/TCBB.2006.26. Cité page 93.

Benny Chor : From quartets to phylogenetic trees. *In Proceedings of the* 25^{th} *Conference on Current Trends in Theory and Practice of Informatics : Theory and Practice of Informatics (SOFSEM'98)*, volume 1521 de *Lecture Notes in Computer Science*, pages 36–53. Springer Verlag, 1998. http://dx.doi.org/10.1007/3-540-49477-4_3. Cité page 65.

Charles Choy, Jesper Jansson, Kunihiko Sadakane et Wing-Kin Sung : Computing the maximum agreement of phylogenetic networks. *Theoretical Computer Science*, 335(1):93–107, 2005. http://dx.doi.org/10.1016/j.tcs.2004.12.012. Cité pages 32 et 37.

Josh Collins, Simone Linz et Charles Semple : Quantifying hybridization in realistic time. *Journal of Computational Biology*, 2011. À paraître, http://dx.doi.org/10.1089/cmb.2009.0166. Cité page 62.

Hans Colonius et Hans-Henning Schulze : Tree structures for proximity data. *British Journal of Mathematical and Statistical Psychology*, 34:167–180, 1981. http://www.staff.uni-oldenburg.de/hans.colonius/download/Tree_structures_for_proximity_data_complete.pdf. Cité page 20.

Pierre Darlu et Pascal Tassy : *La Reconstruction phylogénétique. Concepts et Méthodes*. Masson, 1993. http://sfs.snv.jussieu.fr/pdf/Darlu_Tassy_online.pdf. Cité page 7.

Michael DeGiorgio et James H. Degnan : Fast and consistent estimation of species trees using supermatrix rooted triples. *Molecular Biology and Evolution*, 27(3):552–569, 2010. http://dx.doi.org/10.1093/molbev/msp250. Cité page 65.

James H. Degnan et Noah A. Rosenberg : Discordance of species trees with their most likely gene trees. *PLoS Genetics*, 2(5):e68, 2006. http://dx.doi.org/10.1371/journal.pgen.0020068. Cité page 64.

Holger Dell et Dieter van Melkebeek : Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *In Proceedings of the* 42^{nd} *ACM symposium on Theory of computing (STOC'10)*, 2010. http://dx.doi.org/10.1145/1806689.1806725. Cité page 135.

Edwin Diday : Orders and overlapping clusters in pyramids. *In* J. De Leeuw et al., éditeurs : *Multidimensional Data Analysis*, volume 136, pages 201–234. DSWO Press, Leiden, 1986. http://hal.inria.fr/inria-00075822. Cité page 35.

Rod Downey et Michael R. Fellows : *Parameterized complexity*. Monographs in Computer Science. Springer Verlag, 1999. Cité page 136.

Jean-Philippe Doyon, Celine Scornavacca, Konstantin Yu Gorbunov, Gergely J. Szöllösi, Vincent Ranwez et Vincent Berry : An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications, and transfers. *In Proceedings of the Eighth RECOMB Comparative Genomics Satellite Workshop (RECOMB-CG'10)*, volume 6398 de *Lecture Notes in Computer Science*, pages 93–108. Springer Verlag, 2011. http://dx.doi.org/10.1007/978-3-642-16181-0_9, logiciel disponible sur http://www.atgc-montpellier.fr/MPR/. Cité page 130.

Andreas W. M. Dress : Towards a theory of holistic clustering. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 37:271–289, 1997. Cité page 107.

Andreas W. M. Dress, Katharina T. Huber et Vincent Moulton : Some uses of the Farris transform in mathematics and phylogenetics - a review. *Annals of Combinatorics*, 11(1):1–37, 2007. http://dx.doi.org/10.1007/s00026-007-0302-5. Cité page 34.

Andreas W. M. Dress et Daniel H. Huson : Constructing splits graphs. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 1(3):109–115, 2004. http://dx.doi.org/10.1109/TCBB.2004.27. Cité pages 30, 35 et 58.

Andreas W. M. Dress, Vincent Moulton, Mike Steel et Taoyang Wu : Species, clusters and the "Tree of life" : A graph-theoretic perspective. *Journal of Theoretical Biology*, 265(4):535–542, 2010. http://dx.doi.org/10.1016/j.jtbi.2010.05.031. Cité page 6.

Jean-François Dufayard, Laurent Duret, Simon Penel, Manolo Gouy, François Rechenmann et Guy Perrière : Tree pattern matching in phylogenetic trees : Automatic search for orthologs or paralogs in homologous gene sequence databases. *Bioinformatics*, 21:2596–2603, 2005. http://dx.doi.org/10.1093/bioinformatics/bti325. Cité pages 103, 129 et 140.

Laurent Excoffier et Peter E. Smouse : Using allele frequencies and geographic subdivision to reconstruct gene trees within a species : Molecular variance parsimony. *Genetics*, 136:343–359, 1994. http://www.genetics.org/cgi/content/abstract/136/1/343. Cité page 22.

Joseph Felsenstein : *Inferring Phylogenies*. Sinauer Associates, Inc., 2004. http://www.amazon.com/gp/reader/0878931775/. Cité page 64.

Guillaume Fertin, Anthony Labarre, Irena Rusu, Eric Tannier et Stéphane Vialette : *Combinatorics of Genome Rearrangements*. Computational Molecular Biology. MIT Press, 2009. Cité page 7.

Walter M. Fitch : Homology : a personal view on some of the problems. *Trends in Genetics*, 16 (5):227–231, 2000. http://dx.doi.org/10.1016/S0168-9525(00)02005-9. Cité page 129.

Michel Foucault : *Les mots et les choses - Une archéologie des sciences humaines*, pages 137–176. Tel Gallimard, 1966. http://foucault.50webs.com/books/1966_MC_Chap5.htm. Cité page 4.

Nicolas Galtier : A model of horizontal gene transfer and the bacterial phylogeny problem. *Systematic Biology*, 56:633–642, 2007. http://dx.doi.org/10.1080/10635150701546231. Cité pages 120 et 143.

Philippe Gambette : Who is who in phylogenetic networks : articles, authors and programs, 2010. http://www.atgc-montpellier.fr/phylnet. Cité pages 9, 11, 54 et 130.

Philippe Gambette, Vincent Berry et Christophe Paul : An obstruction approach to reconstruct phylogenies and level-k networks from triplets, 2008. Manuscrit, http://www.lirmm.fr/~gambette/2008GambetteBerryPaul.pdf. Cité page 10.

Philippe Gambette, Vincent Berry et Christophe Paul : The structure of level-k phylogenetic networks. *In Proceedings of the* 20^{th} *Annual Symposium on Combinatorial Pattern Matching (CPM'09)*, volume 5577 de *Lecture Notes in Computer Science*, pages 289–300. Springer Verlag, 2009. http://dx.doi.org/10.1007/978-3-642-02441-2_26. Cité pages 10 et 11.

Philippe Gambette, Vincent Berry et Christophe Paul : Quartets and unrooted phylogenetic networks, 2010. Manuscrit, http://www.lirmm.fr/~gambette/2010GambetteBerryPaul.pdf. Cité pages 10, 12 et 24.

Philippe Gambette et Katharina T. Huber : A note on encodings of phylogenetic networks of bounded level, 2010. Soumis, http://arxiv.org/abs/0906.4324. Cité pages 10, 11, 12 et 54.

Philippe Gambette et Daniel H. Huson : Improved layout of phylogenetic networks. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 5(3), 2008. http://http://dx.doi.org/10.1109/tcbb.2007.1046. Cité page 65.

Philippe Gambette et Jean Véronis : Visualising a text with a tree cloud. *In International Federation of Classification Societies 2009 Conference (IFCS'09)*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 561–570, 2010. http://dx.doi.org/10.1007/978-3-642-10745-0_61. Cité pages 11, 137 et 140.

Paweł Górecki : Reconciliation problems for duplication, loss and horizontal gene transfer. *In Proceedings of the eighth Annual International Conference on Research in Computational Molecular Biology (RECOMB'04)*, pages 316–325, 2004. http://dx.doi.org/10.1145/974614.974656. Cité page 130.

Jens Gramm et Rolf Niedermeier : A fixed-parameter algorithm for minimum quartet inconsistency. *Journal of Computer and System Sciences*, 67(4):723–741, 2003. http://dx.doi.org/10.1016/S0022-0000(03)00077-1. Cité pages 106 et 111.

Verne Grant : *Plant Speciation*. Columbia University Press, 1971. Cité page 5.

Nicholas C. Grassly et Andrew Rambaut : Treevolve, a program to simulate the evolution of DNA sequences under different population dynamic scenarios, 1999. http://www.cecalc.ula.ve/BIOINFO/servicios/herr2/TREEVOLVE/manual.html. Cité page 143.

Stefan Grünewald, Kristoffer Forslund, Andreas W. M. Dress et Vincent Moulton : QNet : An agglomerative method for the construction of phylogenetic networks from weighted quartets. *Molecular Biology and Evolution*, 24(2):532–538, 2007. http://dx.doi.org/10.1093/molbev/msl180. Cité pages 65 et 72.

Stefan Grünewald, Katharina T. Huber, Vincent Moulton, Charles Semple et Andreas Spillner : Characterizing weak compatibility in terms of weighted quartets. *Advances in Applied Mathematics*, 42(3):329–341, 2009. http://dx.doi.org/10.1016/j.aam.2008.07.002. Cité page 83.

Stefan Grünewald, Katharina T. Huber et Qiong Wu : Two new closure rules for constructing phylogenetic super-networks. *Bulletin of Mathematical Biology*, 70(7):1906–1924, 2008. http://dx.doi.org/10.1007/s11538-008-9331-4. Cité page 115.

Alain Guénoche : Graphical representation of a boolean array. *Computers and the Humanities*, 20:277–281, 1986. http://www.jstor.org/stable/30204351. Cité page 35.

Alain Guénoche : Construction du treillis de Galois d'une relation binaire. *Mathématiques et Sciences Humaines*, 109:41–53, 1990. http://www.numdam.org/item?id=MSH_1990__109__41_0. Cité page 134.

Sylvain Guillemot et Vincent Berry : Fixed-parameter tractability of the maximum agreement supertree problem. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 7(2):342–353, 2010. http://dx.doi.org/10.1109/TCBB.2008.93. Cité pages 66, 68 et 107.

Sylvain Guillemot et Matthias Mnich : Kernel and fast algorithm for dense triplet inconsistency. *In Proceedings of the seventh Annual Conference on Theory and Applications of Models of Computation (TAMC'10)*, volume 6108 de *Lecture Notes in Computer Science*, pages 247–257. Springer Verlag, 2010. http://dx.doi.org/10.1007/978-3-642-13562-0_23. Cité pages 111 et 112.

Dan Gusfield et Vikas Bansal : A fundamental decomposition theory for phylogenetic networks and incompatible characters. *In Proceedings of the ninth Annual International Conference on Research in Computational Molecular Biology (RECOMB'05)*, volume 3500 de *Lecture Notes in Computer Science*, pages 217–232. Springer Verlag, 2005. http://dx.doi.org/10.1007/11415770_17. Cité page 89.

Dan Gusfield, Vikas Bansal, Vineet Bafna et Yun S. Song : A decomposition theory for phylogenetic networks and incompatible characters. *Journal of Computational Biology*, 14(10):1247–1272, 2007. http://dx.doi.org/10.1089/cmb.2006.0137. Cité page 87.

Dan Gusfield, Satish Eddhu et Charles Langley : The fine structure of galls in phylogenetic networks. *INFORMS Journal on Computing*, 16(4):459–469, 2004. http://dx.doi.org/10.1287/ijoc.1040.0099. Cité page 39.

Walter Guttmann et Markus Maucher : Variations on an ordering theme with constraints. *In Proceedings of the* 4^{th} *IFIP International Conference on Theoretical Computer Science (TCS'06)*, volume 209 de *IFIP*, pages 77–90, 2006. http://dx.doi.org/10.1007/978-0-387-34735-6_10. Cité page 82.

Sébastien Halary, Jessica W. Leigh, Bachar Cheaib, Philippe Lopez et Eric Bapteste : Network analyses structure genetic diversity in independent genetic worlds. *Proceedings of the National Academy of Sciences*, 107(1):127–132, 2010. http://dx.doi.org/10.1073/pnas.0908978107. Cité page 137.

Mike Hallett et Jens Lagergren : Efficient algorithms for lateral gene transfers problems. *In Proceedings of the fifth Annual International Conference on Research in Computational Molecular Biology (RECOMB'01)*, pages 141–148, 2001. http://dx.doi.org/10.1145/369133.369188. Cité page 62.

Dov Harel et Robert E. Tarjan : Fast algorithms for finding nearest common ancestors. *SIAM Journal on Computing*, 13(2):338–355, 1984. http://dx.doi.org/10.1137/0213024. Cité page 75.

Willi Hennig : *Phylogenetic Systematics*, pages 29–32. University of Illinois Press, 1966. Translated by D. Dwight Davis and Rainer Zangerl. Cité page 5.

Monika R. Henzinger, Valerie King et Tandy Warnow : Constructing a tree from homeomorphic subtrees, with applications to computational evolutionary biology. *Algorithmica*, 24:1–13, 1999. http://dx.doi.org/10.1007/PL00009268. Cité pages 65 et 66.

Barbara R. Holland, Glenn Conner, Katharina T. Huber et Vincent Moulton : Imputing supertrees and supernetworks from quartets. *Systematic Biology*, 56(1):57–67, 2007. http://dx.doi.org/10.1080/10635150601167013. Cité page 115.

Barbara R. Holland et Vincent Moulton : Consensus networks : A method for visualising incompatibilities in collections of trees. *In Proceedings of the third Workshop on Algorithms in Bioinformatics (WABI'03)*, volume 2812 de *Lecture Notes in Computer Science*, pages 165–176. Springer Verlag, 2003. http://dx.doi.org/10.1007/b13243. Cité pages 30 et 63.

John E. Hopcroft et Richard M. Karp : An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973. http://dx.doi.org/10.1137/0202019. Cité page 133.

Carl L. Hubbs : Hybridization between fish species in nature. *Systematic Zoology*, 4(1):1–20, 1955. http://www.jstor.org/stable/2411933. Cité page 5.

Katharina T. Huber, Bengt Oxelman, Martin Lott et Vincent Moulton : Reconstructing the evolutionary history of polyploids from multilabeled trees. *Molecular Biology and Evolution*, 23(9):1784–1791, 2007. http://dx.doi.org/10.1093/molbev/msl045. Cité page 130.

Katharina T. Huber, Leo van Iersel, Steven Kelk et Radoslaw Suchecki : A practical algorithm for reconstructing level-1 phylogenetic networks. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 2010. À paraître, http://dx.doi.org/10.1109/TCBB.2010.17. Cité pages 106, 132 et 140.

Richard R. Hudson : Gene genealogies and the coalescent process. *Oxford Surveys in Evolutionary Biology*, 7:1–44, 1991. Cité page 65.

Jaime Huerta-Cepas, Anibal Bueno, Joaquín Dopazo et Toni Gabaldón : PhylomeDB : a database for genome-wide collections of gene phylogenies. *Nucleic Acids Research*, 36(supp. 1):D491–D496, 2008. http://dx.doi.org/10.1093/nar/gkm899. Cité page 129.

Jaime Huerta-Cepas, Hernán Dopazo, Joaquín Dopazo et Toni Gabaldón : The human phylome. *Genome Biology*, 8:R109, 2007. http://dx.doi.org/10.1186/gb-2007-8-6-r109. Cité page 9.

Daniel H. Huson et David Bryant : Application of phylogenetic networks in evolutionary studies. *Molecular Biology and Evolution*, 23(2):254–267, 2006. http://dx.doi.org/10.1093/molbev/msj030, logiciel disponible sur http://www.splitstree.org. Cité pages 20, 22 et 140.

Daniel H. Huson, Tobias Dezulian, Markus Franz, Christian Rausch, Daniel C. Richter et Regula Rupp : Dendroscope - an interactive tree drawer. *BMC Bioinformatics*, 8(460), 2007. http://dx.doi.org/10.1186/1471-2105-8-460, logiciel disponible sur http://www-ab.informatik.uni-tuebingen.de/software/dendroscope/. Cité pages 23 et 139.

Daniel H. Huson, Tobias Dezulian, Tobias Klöpper et Mike Steel : Phylogenetic super-networks from partial trees. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 1(4):151–158, 2004. http://dx.doi.org/10.1109/TCBB.2004.44. Cité pages 30 et 115.

Daniel H. Huson et Tobias Klöpper : Computing recombination networks from binary sequences. *In Proceedings of the fifth European Conference on Computational Biology (ECCB'05)*, volume 21(suppl. 2) de *Bioinformatics*, pages ii159–ii165, 2005. http://dx.doi.org/10.1093/bioinformatics/bti1126. Cité page 53.

Daniel H. Huson et Tobias Klöpper : Beyond galled trees - decomposition and computation of galled networks. *In Proceedings of the eleventh Annual International Conference on Research in Computational Molecular Biology (RECOMB'07)*, volume 4453 de *Lecture Notes in Computer Science*, pages 211–225. Springer Verlag, 2007. http://dx.doi.org/10.1007/978-3-540-71681-5_15. Cité pages 26 et 36.

Daniel H. Huson, Tobias Klöpper, Peter J. Lockhart et Mike Steel : Reconstruction of reticulate networks from gene trees. *In Proceedings of the ninth Annual International Conference on Research in Computational Molecular Biology (RECOMB'05)*, volume 3500 de *Lecture Notes in Computer Science*, pages 233–249. Springer Verlag, 2005. http://dx.doi.org/10.1007/11415770_18. Cité page 89.

Daniel H. Huson et Regula Rupp : Summarizing multiple gene trees using gene networks. *In Proceedings of the eighth Workshop on Algorithms in Bioinformatics (WABI'08)*, volume 5251 de *Lecture Notes in Computer Science*, pages 296–305. Springer Verlag, 2008. http://dx.doi.org/10.1007/978-3-540-87361-7_25. Cité pages 34, 66, 87 et 140.

Daniel H. Huson, Regula Rupp, Vincent Berry, Philippe Gambette et Christophe Paul : Computing galled networks from real data. *In Proceedings of the 17th Annual Conference on Intelligent Systems for Molecular Biology & 8th European Conference on Computational Biology (ISMB/ECCB'09)*, volume 25(12) de *Bioinformatics*, pages i85–i93, 2009. http://dx.doi.org/10.1093/bioinformatics/btp217. Cité pages 10, 11, 85 et 140.

Daniel H. Huson, Regula Rupp et Celine Scornavacca : *Phylogenetic Networks : Concepts, Algorithms and Applications*. Cambridge University Press, 2011. http://www.phylogenetic-networks.org. Cité pages 54, 86, 130 et 132.

Trinh N. D. Huynh, Jesper Jansson, Nguyen Bao Nguyen et Wing-Kin Sung : Constructing a smallest refining galled phylogenetic network. *In Proceedings of the ninth Annual International Conference on Research in Computational Molecular Biology (RECOMB'05)*, volume 3500 de *Lecture Notes in Computer Science*, pages 265–280. Springer Verlag, 2005. http://dx.doi.org/10.1007/11415770_20. Cité page 62.

Jesper Jansson : On the complexity of inferring rooted evolutionary trees. *In Proceedings of the Brazilian Symposium on Graphs, Algorithms, and Combinatorics (GRACO 2001)*, volume 7 de *Electronic Notes in Discrete Mathematics*, pages 50–53, 2001. http://dx.doi.org/10.1016/S1571-0653(04)00222-7. Cité page 105.

Jesper Jansson, Joseph H.-K. Ng, Kunihiko Sadakane et Wing-Kin Sung : Rooted maximum agreement supertrees. *Algorithmica*, 43(4):293–307, 2005. http://dx.doi.org/10.1007/s00453-004-1147-5. Cité pages 65 et 66.

Jesper Jansson, Nguyen Bao Nguyen et Wing-Kin Sung : Algorithms for combining rooted triplets into a galled phylogenetic network. *SIAM Journal on Computing*, 35(5):1098–1121, 2006. http://dx.doi.org/10.1137/S0097539704446529. Cité pages 66, 68 et 69.

Jesper Jansson et Wing-Kin Sung : Inferring a level-1 phylogenetic network from a dense set of rooted triplets. *Theoretical Computer Science*, 363(1):60–68, 2006. http://dx.doi.org/10.1016/j.tcs.2006.06.022. Cité pages 24, 37, 65, 66, 67 et 73.

Guohua Jin, Luay Nakhleh, Sagi Snir et Tamir Tuller : Maximum likelihood of phylogenetic networks. *Bioinformatics*, 22(21):2604–2611, 2006. http://dx.doi.org/10.1093/bioinformatics/btl452. Cité page 154.

David S. Johnson : The NP-completeness column : An ongoing guide. *Journal of Algorithms*, 8 (5):438–448, 1987. http://dx.doi.org/10.1016/0196-6774(87)90021-6. Cité page 133.

Björn H. Junker et Falk Schreiber : *Analysis of Biological Networks*. Wiley Series in Bioinformatics. Wiley-Blackwell, 2008. Cité page 7.

Volker Kaibel et Alexander Schwartz : On the complexity of polytope isomorphism problems. *Graphs and Combinatorics*, 19(2):215–230, 2003. http://dx.doi.org/10.1007/s00373-002-0503-y. Cité page 118.

Iyad A. Kanj, Luay Nakhleh, Cuong Than et Ge Xia : Seeing the trees and their branches in the network is hard. *Theoretical Computer Science*, 401:153–164, 2008. http://dx.doi.org/10.1016/j.tcs.2008.04.019. Cité pages 25, 61 et 86.

Sampath K. Kannan, Eugene L. Lawler et Tandy J. Warnow : Determining the evolutionary tree using experiments. *Journal of Algorithms*, 21(1):26–50, 1996. http://dx.doi.org/10.1006/jagm.1996.0035. Cité page 65.

Steven Kelk, 2008 : http://homepages.cwi.nl/~kelk/lev3gen/. Cité pages 42 et 117.

Denes Kőnig : Gráphok és mátrixok. *Mathematikai és Fizikai Lapok*, 38:116–119, 1931. Cité page 134.

John F.C. Kingman : Origins of the coalescent : 1974-1982. *Genetics*, 156:1461–1463, 2000. http://www.genetics.org/cgi/content/full/156/4/1461. Cité page 64.

Christian Komusiewicz et Johannes Uhlmann : A cubic-vertex kernel for flip consensus tree. *In IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'08)*, Dagstuhl, Germany, 2008. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany. http://dx.doi.org/10.4230/LIPIcs.FSTTCS.2008.1760. Cité page 93.

Laura S. Kubatko : Identifying hybridization events in the presence of coalescence via model selection. *Systematic Biology*, 58(5):478–488, 2009. http://dx.doi.org/10.1093/sysbio/syp055. Cité page 154.

Eugene L. Lawler et David E. Wood : Branch-and-bound methods : A survey. *Operations Research*, 14(4):699–719, 1966. http://www.jstor.org/stable/168733. Cité page 98.

Pierre Legendre et Vladimir Makarenkov : Improving the additive tree representation of a given dissimilarity matrix using reticulations. *In Data Analysis, Classification, and Related Methods, Proceedings of the seventh Conference on the International Federation of Classification Societies (IFCS'00)*, pages 35–40. Springer Verlag, 2000. http://www.info2.uqam.ca/~makarenv/makarenv/Article_IFCS.pdf. Cité page 23.

Eugene M. Luks : Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25(1):42–65, 1982. http://dx.doi.org/10.1016/0022-0000(82)90009-5. Cité page 118.

Bin Ma, Lusheng Wang et Ming Li : Fixed topology alignment with recombination. *Discrete Applied Mathematics*, 104:281–300, 2000. http://dx.doi.org/10.1016/S0166-218X(00)00196-7. Cité page 39.

Dave MacLeod, Robert L. Charlebois, W. Ford Doolittle et Eric Bapteste : Deduction of probable events of lateral gene transfer through comparison of phylogenetic trees by recursive consolidation and rearrangement. *BMC Evolutionary Biology*, 5(27), 2005. http://dx.doi.org/10.1186/1471-2148-5-27. Cité pages 23 et 62.

Vladimir Makarenkov, Dmytro Kevorkov et Pierre Legendre : Phylogenetic network construction approaches. *In Applied Mycology and Biotechnology*, pages 61–97, 2006. http://dx.doi.org/10.1016/S1874-5334(06)80006-7. Cité page 53.

Kazuhisa Makino et Takeaki Uno : New algorithms for enumerating all maximal cliques. *In Proceedings of the Ninth Scandinavian Workshop on Algorithm Theory (SWAT'04)*, volume 3111 de *Lecture Notes in Computer Science*, pages 260–272. Springer Verlag, 2004. http://dx.doi.org/10.1007/978-3-540-27810-8_23. Cité page 134.

Christopher A. Meacham : Theoretical and computational considerations of the compatibility of qualitative taxonomic characters. *In* J. Felsenstein, éditeur : *Numerical Taxonomy*, volume G1 de *NATO ASI Series*, pages 304–314. Springer Verlag, 1983. Cité page 115.

Gary L. Miller : Graph isomorphism, general remarks. *Journal of Computer and System Sciences*, 18(2):128–142, 1979. http://dx.doi.org/10.1016/0022-0000(79)90043-6. Cité pages 118 et 119.

Bernard M. E. Moret, Luay Nakhleh, Tandy Warnow, C. Randal Linder, Anna Tholse, Anneke Padolina, Jerry Sun et Ruth Timme : Phylogenetic networks : Modeling, reconstructibility, and accuracy. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 1(1):13–23, 2004. http://dx.doi.org/10.1109/TCBB.2004.10. Cité page 143.

Monique M. Morin et Bernard M. E. Moret : NETGEN : generating phylogenetic networks with diploid hybrids. *Bioinformatics*, 22(15):1921–1923, 2006. http://dx.doi.org/10.1093/bioinformatics/btl191. Cité pages 120 et 143.

David A. Morrison : Networks in phylogenetic analysis : new tools for population biology. *International Journal for Parasitology*, 35:567–582, 2005. http://dx.doi.org/10.1016/j.ijpara.2005.02.007. Cité page 53.

David A. Morrison : Using data-display networks for exploratory data analysis in phylogenetic studies. *Molecular Biology and Evolution*, 27(5):1044–1057, 2010. http://dx.doi.org/10.1093/molbev/msp309. Cité page 53.

Vincent Moulton et Katharina T. Huber : Phylogenetic networks. *In* O. Gascuel, éditeur : *Mathematics of evolution and phylogeny*, pages 178–200. Oxford University Press, 2005. Cité page 58.

Vincent Moulton et Katharina T. Huber : Phylogenetic networks from multi-labeled trees. *Journal of Mathematical Biology*, 52(5):613–632, 2006. http://dx.doi.org/10.1007/s00285-005-0365-z. Cité page 130.

Luay Nakhleh : *Phylogenetic Networks*. Thèse de doctorat, University of Texas at Austin, U.S.A., 2004. http://portal.acm.org/citation.cfm?id=1048424. Cité pages 53 et 143.

Luay Nakhleh : A metric on the space of reduced phylogenetic networks. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 7(2), 2010. http://dx.doi.org/10.1109/TCBB.2009.2. Cité page 143.

Luay Nakhleh, Derek Ruths et Li-San Wang : RIATA-HGT : A fast and accurate heuristic for reconstructing horizontal gene transfer. *In Proceedings of the* 11^{th} *Annual International Computing and Combinatorics Conference (COCOON'06)*, volume 3595 de *Lecture Notes in Computer Science*, pages 84–93. Springer Verlag, 2005a. http://dx.doi.org/10.1007/11533719_11. Cité page 63.

Luay Nakhleh, Tandy Warnow, C. Randal Linder et Katherine St. John : Reconstructing reticulate evolution in species - theory and practice. *Journal of Computational Biology*, 12(6):796–811, 2005b. http://dx.doi.org/10.1089/cmb.2005.12.796. Cité page 62.

Jaroslav Opatrny : Total ordering problem. *SIAM Journal on Computing*, 8(1):111–114, 1979. http://dx.doi.org/10.1137/0208008. Cité page 70.

Judea Pearl et Michael Tarsi : Structuring causal trees. *Journal of Complexity*, 2(1):60–77, 1986. http://dx.doi.org/10.1016/0885-064X(86)90023-3. Cité page 65.

René Peeters : The maximum edge biclique problem is NP-complete. *Discrete Applied Mathematics*, 131(3):651–654, 2003. http://dx.doi.org/10.1016/S0166-218X(03)00333-0. Cité page 133.

David Posada et Keith A. Crandall : Intraspecific gene genealogies : trees grafting into networks. *TRENDS in Ecology and Evolution*, 16(1):37–45, 2001. http://dx.doi.org/10.1016/S0169-5347(00)02026-7. Cité page 53.

Mark A. Ragan : Trees and networks before and after Darwin. *Biology Direct*, 4(43), 2009. http://dx.doi.org/10.1186/1745-6150-4-43. Cité pages 4 et 7.

Vincent Ranwez, Vincent Berry, Alexis Criscuolo, Pierre-Henri Fabre, Sylvain Guillemot, Celine Scornavacca et Emmanuel J. P. Douzery : PhySIC : a veto supertree method with desirable properties. *Systematic Biology*, 56(5):798–817, 2007. http://dx.doi.org/10.1080/10635150701639754. Cité page 115.

Francesc Rosselló et Gabriel Valiente : All that glisters is not galled. *Mathematical Biosciences*, 221 (1):54–59, 2009. http://dx.doi.org/10.1016/j.mbs.2009.06.007. Cité pages 32 et 59.

Michael J. Sanderson, Amy C. Driskell, Richard H. Ree, Oliver Eulenstein et Sasha Langley : Obtaining maximal concatenated phylogenetic data sets from large sequence databases. *Molecular Biology and Evolution*, 20(7):1036–1042, 2003. http://dx.doi.org/10.1093/molbev/msg115. Cité page 134.

Alexander Schrijver : *Combinatorial Optimization - Polyhedra and Efficiency*. Springer Verlag, 2003. Cité page 20.

Celine Scornavacca, Vincent Berry et Vincent Ranwez : Building species trees from larger parts of phylogenomic databases. *Information and Computation*, 209(3):590–605, 2011. http://dx.doi.org/10.1016/j.ic.2010.11.022. Cité page 130.

Charles Semple et Mike Steel : *Phylogenetics*. Oxford University Press, 2003. Cité page 33.

Charles Semple et Mike Steel : Unicyclic networks : compatibility and enumeration. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 3:84–91, 2006. http://dx.doi.org/10.1109/TCBB.2006.14. Cité pages 39, 42, 50 et 51.

João C. Setubal et João Meidanis : *Introduction to Computational Molecular Biology*. PWS Publishing Company, 1997. Cité page 7.

Neil J.A. Sloane : The on-line encyclopedia of integer sequences, 2010. Published electronically at http://www.research.att.com/~njas/sequences/. Cité page 119.

Sagi Snir et Tamir Tuller : The NET-HMM approach : Phylogenetic network inference by combining maximum likelihood and hidden markov models. *Journal of Bioinformatics and Computational Biology*, 7(4):625–644, 2009. http://dx.doi.org/10.1142/S021972000900428X. Cité page 154.

Matthew Spencer, David Bryant et Edward Susko : Conditioned genome reconstruction : How to avoid choosing the conditioning genome. *Systematic Biology*, 56(1):25–43, 2007. http://dx.doi.org/10.1080/10635150601156313. Cité page 137.

Mike Steel : The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of Classification*, 9:91–116, 1992. http://dx.doi.org/10.1007/BF02618470. Cité pages 65 et 70.

Mike Steel et Angèle M. Hamel : Finding a maximum compatible tree is NP-hard for sequences and trees. *Applied Mathematics Letters*, 9(2):55–60, 1996. http://dx.doi.org/10.1016/0893-9659(96)00012-2. Cité page 90.

Korbinian Strimmer et Vincent Moulton : Likelihood analysis of phylogenetic networks using directed graphical models. *Molecular Biology and Evolution*, 17(6):875–881, 2000. http://www.ncbi.nlm.nih.gov/pubmed/10833193. Cité page 154.

Cuong Than, Derek Ruths et Luay Nakhleh : PhyloNet : A software package for analyzing and reconstructing reticulate evolutionary relationships. *BMC Bioinformatics*, 9(322), 2008. http://dx.doi.org/10.1186/1471-2105-9-322. Cité page 63.

Torsten Tholey : Improved algorithms for the 2-vertex disjoint paths problem. *In Proceedings of the 35*$^{\text{th}}$ *International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'09)*, volume 5404 de *Lecture Notes in Computer Science*, pages 546–557. Springer Verlag, 2009. http://dx.doi.org/10.1007/978-3-540-95891-8_49. Cité page 69.

Thu-Hien To et Michel Habib : Level-k phylogenetic networks are constructable from a dense triplet set in polynomial time. *In Proceedings of the* 20$^{\text{th}}$ *Annual Symposium on Combinatorial Pattern Matching (CPM'09)*, volume 5577 de *Lecture Notes in Computer Science*, pages 275–288. Springer Verlag, 2009. http://dx.doi.org/10.1007/978-3-642-02441-2_25. Cité pages 66, 67, 68, 73, 132 et 151.

Ali Tofigh, Mike Hallett et Jens Lagergren : Simultaneous identification of duplications and lateral gene transfers. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 8(2):517–535, 2011. http://dx.doi.org/10.1109/TCBB.2010.14. Cité page 130.

Leo van Iersel, Judith Keijsper, Steven Kelk, Leen Stougie, Ferry Hagen et Teun Boekhout : Constructing level-2 phylogenetic networks from triplets. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 6(4):667–681, 2009a. http://dx.doi.org/10.1109/TCBB.2009.22. Cité pages 39, 42, 66, 67, 68 et 122.

Leo van Iersel et Steven Kelk : A short note on the tractability of constructing phylogenetic networks from clusters, 2009. http://arxiv.org/abs/0912.4502. Cité page 151.

Leo van Iersel et Steven Kelk : Constructing the simplest possible phylogenetic network from triplets. *Algorithmica*, 2010. À paraître, http://dx.doi.org/10.1007/s00453-009-9333-0. Cité pages 68 et 140.

Leo van Iersel et Steven Kelk : When two trees go to war. *Journal of Theoretical Biology*, 269(1):245–255, 2011. http://dx.doi.org/10.1016/j.jtbi.2010.10.032. Cité pages 27, 68 et 85.

Leo van Iersel, Steven Kelk et Matthias Mnich : Uniqueness, intractability and exact algorithms : reflections on level-k phylogenetic networks. *Journal of Bioinformatics and Computational Biology*, 7(4):597–623, 2009b. http://dx.doi.org/10.1142/S0219720009004308. Cité pages 68, 105 et 122.

Leo van Iersel, Steven Kelk, Regula Rupp et Daniel H. Huson : Phylogenetic networks do not need to be complex : Using fewer reticulations to represent conflicting clusters. *In Proceedings of the* 18^{th} *International Conference on Intelligent Systems in Molecular Biology (ISMB'10)*, volume 26(12) de *Bioinformatics*, pages i124–i131, 2010a. http://dx.doi.org/10.1093/bioinformatics/btq202. Cité pages 23, 85, 132 et 140.

Leo van Iersel, Charles Semple et Mike Steel : Locating a tree in a phylogenetic network. *Information Processing Letters*, 110(23), 2010b. http://dx.doi.org/10.1016/j.ipl.2010.07.027. Cité pages 59, 61 et 86.

Fernanda B. Viégas et Martin Wattenberg : Tag clouds and the case for vernacular visualization. *ACM Interactions*, 15(4):49–52, 2008. http://dx.doi.org/10.1145/1374489.1374501. Cité page 137.

Rainer Wetzel : *Zur Visualisierung abstrakter Ähnlichkeitsbeziehungen*. Thèse de doctorat, Universität Bielefeld, Germany, 1995. Cité page 58.

Stephen J. Willson : Unique determination of some homoplasies at hybridization events. *Bulletin of Mathematical Biology*, 69(5):1709–1725, 2007. http://dx.doi.org/10.1007/s11538-006-9187-4. Cité page 53.

Stephen J. Willson : Properties of normal phylogenetic networks. *Bulletin of Mathematical Biology*, 72(2):340–358, 2010a. http://dx.doi.org/10.1007/s11538-009-9449-z. Cité pages 27 et 59.

Stephen J. Willson : Regular networks can be uniquely constructed from their trees. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 2010b. À paraître, http://dx.doi.org/10.1109/TCBB.2010.69. Cité page 122.

Steven M. Woolley, David Posada et Keith A. Crandall : A comparison of phylogenetic network methods using computer simulation. *PLoS ONE*, 3(4):e1913, 2008. http://dx.doi.org/10.1371/journal.pone.0001913. Cité pages 30 et 144.

Bang Ye Wu : Constructing the maximum consensus tree from rooted triples. *Journal of Combinatorial Optimization*, 29:29–39, 2004. http://dx.doi.org/10.1023/B:JOCO.0000021936.04215.68. Cité page 105.

Gang Wu, Jia-Huai You et Guohui Lin : A polynomial time algorithm for the minimum quartet inconsistency problem with $O(n)$ quartet errors. *Information Processing Letters*, 100:167–171, 2006. http://dx.doi.org/10.1016/j.ipl.2006.05.013. Cité page 106.

Yufeng Wu : Close lower and upper bounds for the minimum reticulate network of multiple phylogenetic trees. *In Proceedings of the* 18^{th} *International Conference on Intelligent Systems in Molecular Biology (ISMB'10)*, volume 26(12) de *Bioinformatics*, pages i140–i148, 2010. http://dx.doi.org/10.1093/bioinformatics/btq198. Cité page 62.

Yufeng Wu et Jiayin Wang : Fast computation of the exact hybridization number of two phylogenetic trees. *In Proceedings of the sixth International Symposium on Bioinformatics Research and Applications (ISBRA'10)*, Lecture Notes in Computer Science. Springer Verlag, 2010. http://dx.doi.org/10.1007/978-3-642-13078-6_23. Cité page 62.

Changhui Yan, J. Gordon Burleigh et Oliver Eulenstein : Identifying optimal incomplete phylogenetic data sets from sequence databases. *Molecular Phylogenetics and Evolution*, 35(3):528–535, 2005. http://dx.doi.org/10.1016/j.ympev.2005.02.008. Cité page 134.

Viktor N. Zemlyachenko, Nikolai M. Korneenko et Regina I. Tyshkevich : Graph isomorphism problem. *Journal of Mathematical Sciences*, 29(4):1426–1481, 1985. http://dx.doi.org/10.1007/BF02104746. Cité page 118.

Glossaire français-anglais

Cette thèse fait référence à certains concepts initialement introduits en anglais, dont la traduction en français n'est pas encore établie.

Nous indiquons donc ci-dessous pour les termes choisis en français le concept correspondant en anglais dans la littérature, ainsi que d'autres traductions non évidentes.

français	**anglais**
arbre multi-étiqueté	MUL-tree / multi-labeled tree
arc d'hybridation	hybrid edge / arc
arc de spéciation	tree edge / arc
bipartition	split
blob	bridgeless component
clade	cluster
clade souple	softwired cluster
clade strict	hardwired cluster
N contient les triplets, clades…	N is consistent with the triplets, clusters…
hiérarchie faible	weak hierarchy
ensembles circulaires de bipartitions	circular split systems
exogroupe	outgroup
graphe orienté	directed graph
isthme	cut edge / arc
nombre d'hybridation	hybridization number
nuage arboré	tree cloud
réseau à une couche de réticulation	galled network
réseau de niveau 1	galled tree, level-1 network
réseau de bipartitions	split network
réseau de clades stricts	cluster network
réseau abstrait	abstract / implicit / data display network
réseau explicite	explicit / evolutionary network
réseau sans descendance hybride	tree-child network
réseau sans fratrie hybride	tree-sibling network
SN-ensemble	SN-set
sommet hybride	hybrid / reticulation / recombination vertex
sommet de spéciation	split vertex, speciation node
suppression de gène	gene deletion

Index

Table des figures

Liste des tableaux

Publications en marge du sujet de thèse

Outre les travaux décrits dans ce manuscrit, j'ai participé pendant mes trois années de thèse à d'autres travaux de recherche, qui ont conduit à des publications.

Algorithmique des graphes

Lors de mon stage de master, j'avais étudié un paramètre de graphes, le nombre intervallaire, qui permet de définir un certain type de graphes d'intersection, les graphes 2-intervallaires, avec Michel Habib et Stéphane Vialette. À l'occasion des Journées Graphes et Algorithmes de 2007, nous avons commencé avec Christophe Crespelle une collaboration sur un nouveau paramètre de graphes, la *linéarité*, qui exprime le nombre d'ordres sur les sommets nécessaires afin que les voisinages des sommets d'un graphe apparaissent comme une union d'intervalles dans ces ordres. Ce paramètre permet donc un codage efficace des voisinages. Or nous avons montré qu'il n'était pas borné par une constante pour des classes simples de graphes d'intersection : les graphes d'intervalles et les graphes de permutation. Nous avons donc proposé un codage efficace des voisinages dans ces deux classes, dans l'article suivant :

- Christophe Crespelle et Philippe Gambette (2009), Efficient Neighbourhood Encoding for Interval Graphs and Permutation Graphs and $O(n)$ Breadth-First Search, *Proceedings of the 20th International Workshop on Combinatorial Algorithms (IWOCA'09)*, *LNCS* 5874, p. 146-157.

Dans cet article, nous avons également proposé un algorithme exploitant de manière fine les voisinages. Il s'agit d'un *parcours en largeur des graphes de permutation*, prenant en entrée un modèle de permutation du graphe et un ordre de priorité sur les sommets, qui s'exécute en temps $O(n)$, où n est le nombre de sommets du graphe. Nous avons généralisé cet algorithme aux graphes trapézoïdaux, illustrés en figure 1, dans l'article :

- Christophe Crespelle et Philippe Gambette (2010), Unrestricted and Complete Breadth-First Search of Trapezoid Graphs in $O(n)$ Time, *Information Processing Letters* 110, p. 497-502.

Traitement automatique des langues naturelles

Jean Véronis avait proposé fin 2007 une visualisation des mots d'un texte inspirée des nuages de mots et des arbres phylogénétiques, que nous avons appelé *nuage arboré*, évo-

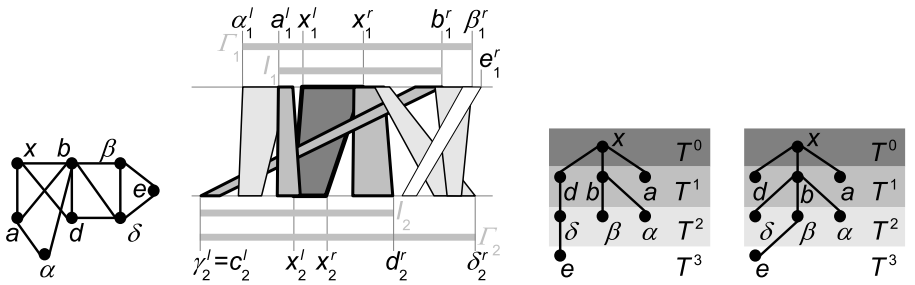


FIGURE 1 : Un graphe trapézoidal (i), son modèle trapézoïdal et les diverses données stockées en mémoire à un instant de l'exécution de l'algorithme (ii), deux arbres de parcours en largeur, l'un respectant l'ordre de priorité $(x\beta\delta dbeca\alpha)$ (iii) et l'autre respectant l'ordre $(x\beta\delta bdeca\alpha)$ (iv).

quée dans le chapitre 4 de ce manuscrit pour ses applications en bioinformatique. Nous avons testé la robustesse des arbres reconstruits en fonction de la formule statistique choisie pour exprimer une distance sémantique entre deux mots d'un texte basée sur leur cooccurrence dans le texte, dans l'article :

- Philippe Gambette et Jean Véronis (2009), Visualising a Text with a Tree Cloud, *Proceedings of the International Federation of Classification Societies 2009 Conference (IFCS'09), Studies in Classification, Data Analysis, and Knowledge Organization* 40, p. 561-570.

Cette collaboration a débouché sur le logiciel libre TreeCloud de construction de nuages arborés, référencé par le Projet Plume [1]. Avec Jean-Charles Bontemps, nous avons conçu une interface web disponible sur le site http://www.treecloud.org, dont une copie d'écran est montrée en figure 2. Nous avons également écrit un article avec Delphine Amstutz pour détailler les possibles utilisations de cette visualisation en analyse littéraire, par une analyse contrastée de deux pièces de Corneille notamment, *Cinna* et *Othon* :

- Delphine Amstutz et Philippe Gambette (2010), Utilisation de la visualisation en nuage arboré pour l'analyse littéraire, *Proceedings of the 10th International Conference on statistical analysis of textual data (JADT'10), Statistical Analysis of Textual Data*, p. 227-238.

Avec Hyeran Lee, nous avons travaillé, suite à la rencontre interdisciplinaire de doctorants OSIDMESH 2009 [2], sur une méthode permettant une évaluation automatique de la *densité des idées*. Il s'agit d'un indicateur linguistique dont la dégradation est liée avec l'apparition de la maladie d'Alzheimer, comme nous avons pu le vérifier avec des résultats présentés dans un poster à CEDIL 2010. Cette méthode, à base de règles linguistiques

1. http://www.projet-plume.org/relier/treecloud
2. http://www.lirmm.fr/~semindoc/Osidmesh.html

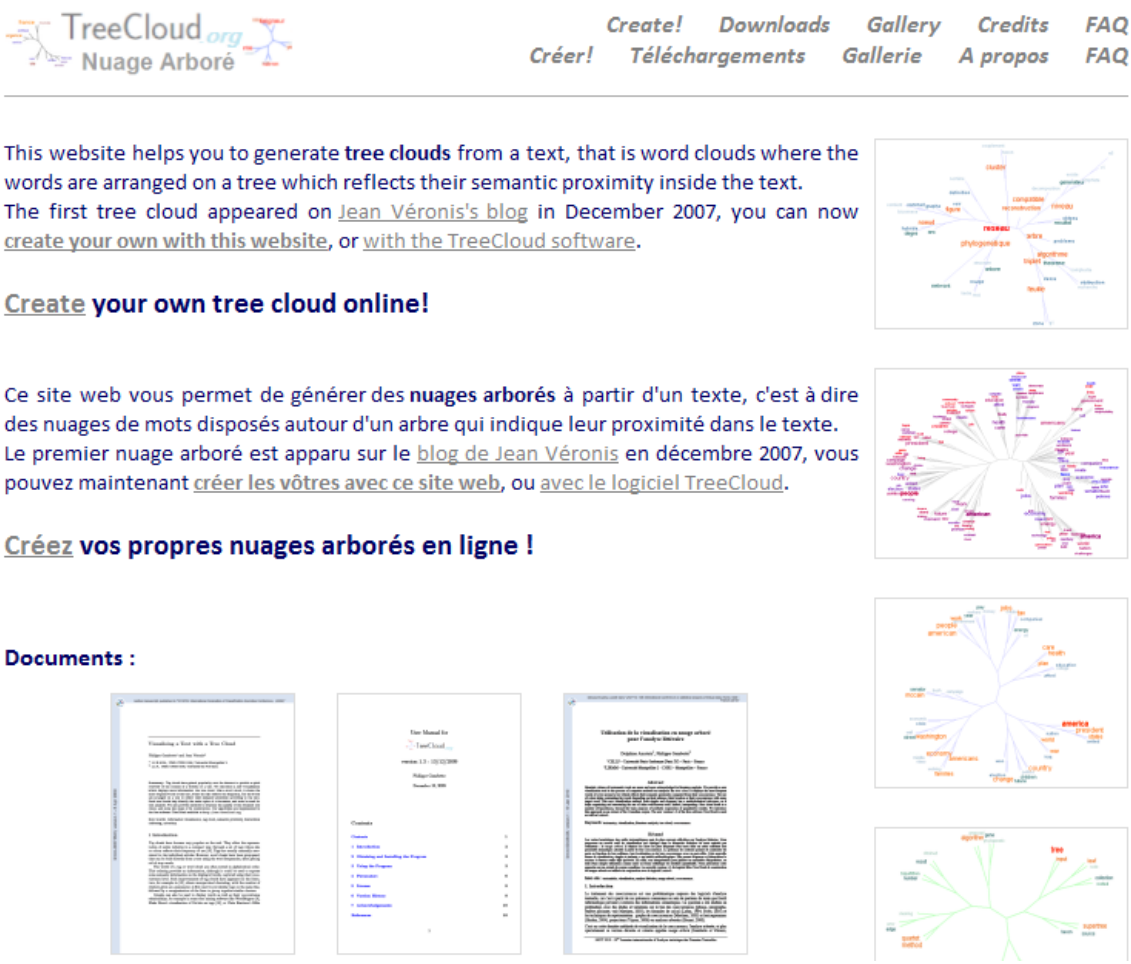


FIGURE 2 : Capture d'écran du site web TreeCloud.org de construction de nuages arborés.

appliquées sur un texte étiqueté grammaticalement, a été présentée lors du colloque RECITAL 2010. Elle est implémentée dans le logiciel libre Densidées [3], dont une copie d'écran est montrée en figure 3.

- Hyeran Lee, Philippe Gambette, Elsa Maillé et Constance Thuillier (2010), Densidées : calcul automatique de la densité des idées dans un corpus oral, *Douzièmes Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL'10)*.
- Hyeran Lee, Philippe Gambette, et Melissa Barkat-Defradas (2010), Utilisation de l'analyse textuelle automatique dans la recherche sur la maladie d'Alzheimer, poster présenté au *deuxième Colloque international des jeunes chercheurs en Didactique des Langues et en Linguistique (CEDIL'10)*.

3. http://code.google.com/p/densidees/

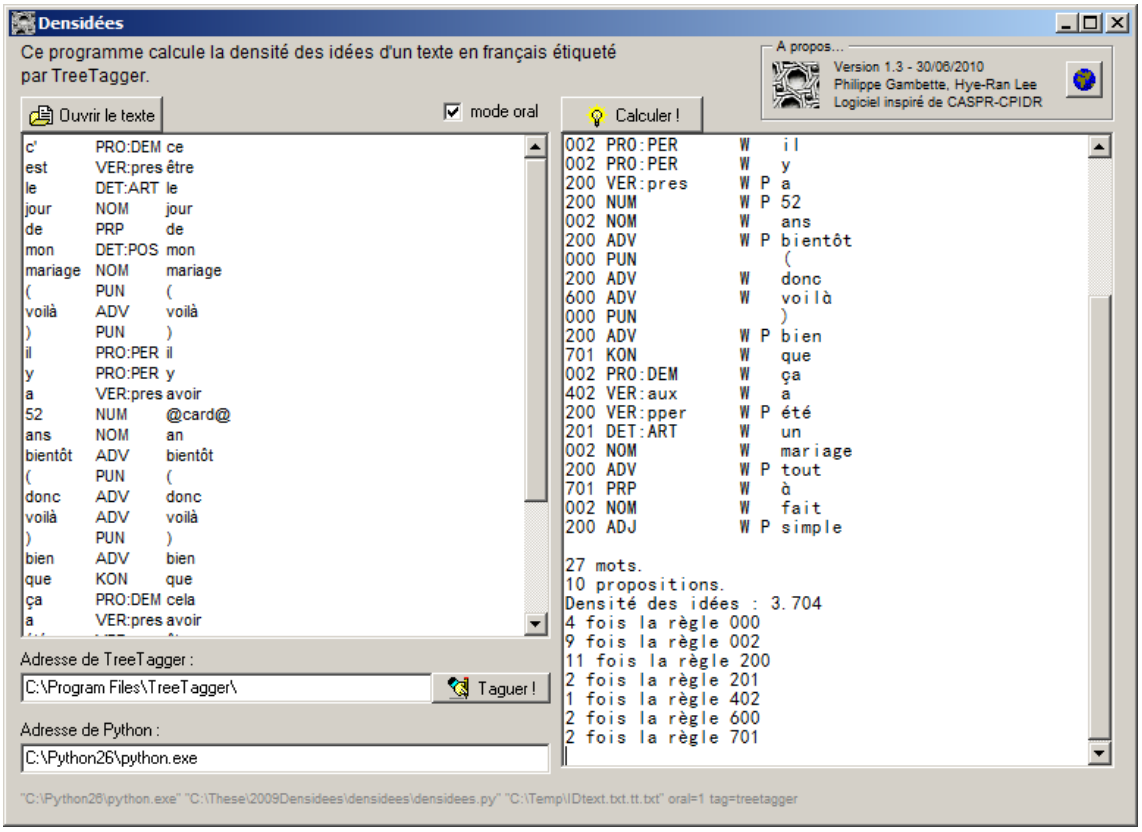


FIGURE 3 : Capture d'écran du logiciel libre Densidées de calcul de la densité des idées d'un texte en français.

Abstract

Phylogenetic networks generalize the tree concept to model Evolution, by allowing edges between branches inside the tree to reflect genetic material exchanges between coexisting species. Lots of combinatorial approaches have been designed to reconstruct networks from data extracted from a set of contradictory gene trees. These approaches can be divided into several categories depending on the kind of input, i.e. triplets, quartets, clusters and splits, and on the kind of structure restrictions they impose on reconstructed networks.
We particularly analyze the structure of one class of such restricted networks, namely level-k phylogenetic networks, and adapt this level parameter to the unrooted context. We also give new combinatorial methods to reconstruct phylogenetic networks from clusters - implemented in Dendroscope - or quartets. We study the limits of combinatorial methods (complexity explosion, noise and silence in the data, ambiguity in the reconstucted network), and the way to tackle them, in particular with an appropriate data preprocessing. Finally we illustrate the results of these reconstruction methods on a dataset, and we conclude on how to use them in a global methodology which integrates statistical aspects.

Keywords: *phylogeny, phylogenetic networks, combinatorics, graph algorithms*

Résumé

Les réseaux phylogénétiques généralisent le modèle de l'arbre pour décrire l'évolution, en permettant à des arêtes entre les branches de l'arbre d'exprimer des échanges de matériel génétique entre espèces coexistantes. De nombreuses approches combinatoires - fondées sur la manipulation d'ensembles finis d'objets mathématiques - ont été conçues pour reconstruire ces réseaux à partir de données extraites de plusieurs arbres de gènes contradictoires. Elles se divisent en plusieurs catégories selon le type de données en entrée (triplets, quadruplets, clades ou bipartitions) et les restrictions de structure sur les réseaux reconstruits.
Nous analysons en particulier la structure d'une classe de réseaux restreints, les réseaux de niveau k, et adaptons ce paramètre de niveau au contexte non enraciné. Nous donnons aussi de nouvelles méthodes combinatoires pour reconstruire des réseaux phylogénétiques, à partir de clades - méthode implémentée dans le logiciel Dendroscope - ou de quadruplets. Nous étudions les limites de ces méthodes combinatoires (explosion de complexité, bruit et silence dans les données, ambiguïté des réseaux reconstruits) et la façon de les prendre en compte, en particulier par un pré-traitement des données. Finalement, nous illustrons les résultats de ces méthodes de reconstruction sur des données réelles avant de conclure sur leur utilisation dans une méthodologie globale qui intègre des aspects statistiques.

Mots clefs : *phylogénie, réseaux phylogénétiques, combinatoire, algorithmique des graphes*

LIRMM — 161, rue Ada — 34095 Montpellier cedex 5 — France