Combinaison de modèles phylogénétiques et longitudinaux pour l'analyse des séquences biologiques : reconstruction de HMM profils ancestraux

Jean-Baka Domelevo Entfellner

ACADÉMIE DE MONTPELLIER
UNIVERSITÉ MONTPELLIER 2
Sciences et Techniques du Languedoc

THÈSE

présentée au Laboratoire d'Informatique,
de Robotique et de Microélectronique de Montpellier
en vue de l'obtention du diplôme de doctorat

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

Combinaison de modèles phylogénétiques et longitudinaux pour l'analyse des séquences biologiques : reconstruction de HMM profils ancestraux

par

Jean-Baka DOMELEVO ENTFELLNER

Soutenue le 15 décembre 2011, devant le jury composé de :

Directeur de thèse
M. Olivier GASCUEL, Directeur de Recherche CNRS ..... LIRMM, Université Montpellier 2

Rapporteurs
M. Mathieu BLANCHETTE, Assistant Professor ..... McGill University, Montréal, Canada
Mme Sophie SCHBATH, Directrice de Recherche INRA ..... MIG, INRA Jouy-en-Josas

Examinateurs
M. Manolo GOUY, Directeur de Recherche CNRS ..... LBBE, Université Lyon 1
M. Vincent RANWEZ, Maître de Conférences ..... ISEM, Université Montpellier 2

Je dédie cette thèse à ma mère,
partie trop tôt pour voir son fils docteur.

Table des matières

Remerciements

Les remerciements, on garde toujours ça pour la fin. Cerise sur le gâteau d'un travail de plusieurs années qui doit beaucoup à son auteur mais aussi pas mal aux autres, à ceux qui l'ont enduré (l'auteur) pendant que « ça » se faisait.

Les remerciements, c'est pas facile. On ne sait pas qui citer en premier, qui en dernier. Attention, je vous préviens : il y en aura forcément qui seront au milieu, coincés entre untel et tel autre. C'est comme ça, n'en déplaise aux mordus de l'hypertexte, la parole écrite est irrémédiablement linéaire.

Les remerciements, c'est la seule partie de ce petit pavé qui échappe à la relecture du directeur de thèse. Interdit. C'est ma chose à moi. C'est d'ailleurs peut-être bien lui que je vais remercier en premier lieu, ce cher Olivier Gascuel qui acceptait, un beau jour du premier trimestre 2007, de signer avec moi pour trois (?) ans. Le pauvre ne pouvait pas savoir dans quoi il s'embarquait. Je le remercie donc, et c'est bien le moins, pour cette confiance qu'il m'a accordée. Je ne suis pas toujours facile à vivre, notamment lorsqu'il faut accepter ma propension à faire tout un tas de choses autres que le travail pour lequel une institution me rémunère et à ne produire que sous la pression créée par l'imminence d'une date butoir.

En dehors d'Olivier, Laurent Bréhélin, Alain Jean-Marie et Vincent Ranwez sont sans nul doute les collègues qui m'ont le plus apporté par les discussions que j'ai pu avoir avec eux quant à des sujets touchant à mon travail au cours de ce doctorat. L'occasion m'est donnée ici de reconnaître cette dette que j'ai envers eux et de les remercier chaleureusement. Laurent et Vincent m'ont toujours apporté des remarques et des commentaires pertinents lors des Comités de Suivi de Thèse auxquels ils ont participé, notamment

Vincent en ce qui concerne la possibilité de moduler l'architecture des modèles en fonction de la position phylogénétique. Alain a dispensé pour l'école doctorale un cours sur les processus de Markov qui m'a beaucoup servi et qui a trouvé un prolongement dans les nombreuses discussions que nous avons eues par la suite, notamment à propos de la modélisation des longueurs d'insertion dans les phylo-HMM.

Je veux remercier ici Sophie Schbath et Mathieu Blanchette, qui ont accepté de rapporter sur ma thèse dans des délais très courts et qui m'ont fait part de remarques pertinentes sur le contenu de mon manuscrit. Je remercie également Manolo Gouy d'avoir accepté avec bienveillance de faire partie de mon jury de thèse.

Je remercie celles et ceux qui m'ont donné l'opportunité d'enseigner à l'Université Montpellier 2 et, partant, qui m'ont permis de me confirmer à moi-même l'intérêt que j'éprouvais pour la chose pédagogique. Ces remerciements vont à toute l'équipe du département d'informatique, au sein de laquelle je peux singulariser quelques personnes : Philippe Janssen a été officiellement mon tuteur, mais avant tout mon capitaine d'équipe... de volley-ball. Merci à lui pour les encouragements sur le terrain. Christophe Dony et Annie Château m'ont fait confiance pour assurer un enseignement en programmation applicative. Merci à eux, tout particulièrement à Annie pour sa gentillesse et les commandes de vin de Touraine et d'Amboise ! Je veux remercier aussi les repsonsables de l'enseignement transversal FLIN102, dans lequel je suis intervenu : Thérèse Libourel, Sèverine Bérard et Anne-Muriel Arigon. Mes étudiants, auxquels j'ai tant bien que mal tenté de faire passer un message, peuvent aussi être remerciés, pour s'être levés tôt et – parfois – avoir fait preuve d'esprit logique et critique. C'est tout le bien qu'on leur souhaite.

Une mention toute particulière va à Gilles Caraux, avec lequel j'ai pu enseigner la statistique en M2. Gilles m'aura apporté bien plus que son expérience d'une carrière d'enseignant en statistique et en analyse de données : celle d'un cuisinier hors pair doublé d'un hôte chaleureux. Il a offert à toute ma petite famille le gîte et le couvert lorsque nous en avons eu soudainement besoin, ce que je n'oublie pas.

Au chapitre des coups de main techniques, je délivre une mention spéciale à Floréal Morandat, toujours disponible pour aider un collègue en détresse à déboguer son code C++. Par ailleurs, ce tapuscrit de thèse est mis en forme avec un style LaTeX développé principalement par Floréal. Mon bon vieux Flop, je te dois une fière chandelle ! Je remercie également Céline Scornavacca et Julien Dutheil pour leur aide sur l'excellente bibliothèque logicielle Bio++.

Mes collègues doctorants de l'équipe MAB ont bien sûr leur place dans ces remerciements. Je tiens à citer mes collègues de bureau Olivier Mirabeau, Nicolas Terrapon et Nicolas Philippe, pour les heures passées ensemble à travailler, rire à gorge déployée en

dérangeant tout le monde alentour, jouer aux Échecs ou débattre de tel ou tel sujet. Je remercie aussi Sam, Raluca, Pierre, Yoan, Mathieu J. et bien sûr Amel. Les deux premiers ont déjà soutenu ; pour le reste, comme l'a si bien dit Léa Fehner, « qu'un seul tienne et les autres suivront. » Courage, moussaillons !

Je tiens à remercier tous mes autres collègues de l'équipe MAB, dans laquelle il règne une ambiance toujours agréable et amicale : Mathieu L., Fabio, JP, Laurent, Vincent L., Éric, Alban (les nombreux repas partagés le midi avec ces quatre derniers auront été l'occasion de discussions souvent drôles et animées, parfois passionnantes et parfois, c'est humain, inintéressantes), Vincent B. et François (merci de m'avoir fait découvrir les vins du Mas de la Séranne).

La fin d'une thèse, c'est aussi la fin d'une période. En ce qui me concerne, ce passage par Montpellier a découpé une tranche de vie bien remplie. Lorsque je me retourne en arrière sur ces quatre années, je vois un club d'Échecs passer de quelque soixante licenciés à plus de deux cents (avec une réunification au passage), des camarades syndicalistes et des mouvements de grève contre la loi LRU, un voyage en Afrique, des voitures et des garages, etc. : durant ces quelques années, j'ai énormément appris en bioinformatique mais aussi en relations humaines ou en mécanique automobile. Dans ce dernier domaine, je remercie Jean-Philippe, Bertrand, Ahmed, Bubu, David et tous les casseurs du coin. Une petite pensée aussi pour mes compagnons de transsaharienne, Lulu et Roulian.

Mes camarades du SNTRS-CGT et plus généralement les militants syndicaux de la CGT, des Solidaires et du SNESUP de l'UM2 ont été plus que des collègues : à leur côté, j'ai fait ma première expérience directe de l'action revendicative sur mon lieu de travail. Je les salue fraternellement, ainsi bien sûr que tous les copains de Lutte Ouvrière à Montpellier.

Je remercie chaleureusement les collègues des services du LIRMM, pour leur disponibilité et leur gentillesse. Je pense tout particulièrement à Nicole Olivet, Elisabeth Grèverie, Elisabeth Petiot, Laurie Lavernhe et Nicolas Serrurier, qui ont assuré ou assurent encore l'accueil au laboratoire. Je veux remercier également les employés du groupe Alter Services, auxquels on pense trop peu souvent alors même qu'ils nous permettent au quotidien de travailler dans de bonnes conditions. Bon, il y a tellement de personnes à remercier au labo que je ne peux citer tout le monde : on comprendra.

Je n'oublie pas non plus les salarié(e)s du CROUS de Montpellier, de Languedoc Restauration et de la Sodexo qui m'ont régulièrement fait la cuisine durant les quatre années qu'a duré cette thèse. Faut-il rappeler ici que les nourritures spirituelles ne sont pas, loin s'en faut, les plus indispensables à la vie concrète d'un homme ?

C'est une chose évidente, mille fois dite par d'autres, mais ni l'évidence, ni la répétition

ne m'empêchent d'être ce mille-et-unième : je remercie mes parents, sans lesquels je n'aurais pu en arriver là.

Bien évidemment, je termine ces remerciements en citant les membres de ma petite famille à moi : ma douce Bérengère est et a été une compagne bien endurante pendant les périodes de travail intense. Elle m'aura soutenu dans les moments de déception ou de doute, accompagné dans les moments de joie. Je la remercie du fond du cœur et souhaite avoir encore bien des occasions de le faire ! Notre petit Mopti m'a lui aussi accompagné quelques fois au LIRMM, fort à propos lorsqu'il s'agissait d'éviter les situations de « travail isolé ». À la maison aussi, il s'est roulé plus d'une fois en une chaude boule de poils sur mes pieds pendant que je travaillais, les mains réchauffées par mon ordinateur portable, les pieds par mon chien !

À toutes et à tous, ceux que j'ai cités comme ceux que j'ai oubliés, je dis un grand merci !

Introduction

En exposant au milieu du dix-neuvième siècle la théorie de l'évolution, Charles Darwin donnait au monde un outil puissant d'analyse et de compréhension des organismes vivants, perçus pour la première fois dans leur grande diversité actuelle *et* dans ce qu'ils héritent en commun du passé. Ce faisant, il rendait possible les comparaisons interindividuelles ou interspécifiques, aujourd'hui fondamentales dans le domaine des sciences biologiques. On a découvert depuis lors que séquence, structure et fonction sont liées : des protéines présentant des séquences d'acides aminés que l'on peut mettre en correspondance l'une avec l'autre partageront souvent des fonctions semblables (par exemple l'hémoglobine et la myoglobine sont des protéines assurant toutes deux le transport de l'oxygène).

Lorsqu'on souhaite étudier des séquences partageant la même fonction, on les regroupe naturellement pour ensuite former un modèle statistique descriptif de l'ensemble. Traditionnellement, il existe deux sortes de tels modèles :

- ou bien l'on insiste sur les caractéristiques communes dans ce qu'elles ont de directement mesurable sur les séquences contemporaines,
- ou bien l'on modélise avant tout l'*histoire évolutive* qui a mené aux séquences observées.

Dans la première classe de modèles on peut citer les HMM profils (*Hidden Markov Models*, ou modèles de Markov cachés), qui donnent une interprétation statistique linéaire (ou longitudinale) des séquences apparentées en décrivant la succession de caractères qui les constitue. La deuxième classe de modèles comprend entre autres les arbres phylogénétiques, qui décrivent l'évolution des séquences à partir de processus substitutionnels permettant de mesurer la vraisemblance des différents chemins évolutifs ayant pu mener

aux séquences contemporaines.

La combinaison de ces deux types de modèles reste à ce jour très peu courante. Parmi les précurseurs en la matière, on peut citer divers travaux dus à plusieurs auteurs :

- Mitchison et Durbin [Mitchison et Durbin, 1995] ont été les premiers à présenter une tentative de construction de modèles alliant phylogénies et modèles longitudinaux. Leur approche s'appuyait logiquement sur les HMM profils qu'ils venaient de faire connaître à la communauté en les important d'une autre discipline, la reconnaissance de la parole. Malheureusement, leur objectif (l'amélioration des alignements de séquence) ne fut pas clairement atteint.
- Un peu plus tard, Qian et Goldstein [Qian et Goldstein, 2003] ont repris l'idée de ces auteurs pour proposer une approche plus systématique mais toujours incomplète. À la différence de Mitchison et Durbin, leur objectif n'était pas de raffiner des alignements de séquences mais d'aller rechercher des homologues distants, ce qu'ils ont pu réaliser avec un certain succès.
- Enfin, Siepel et Haussler [Siepel et Haussler, 2005] ont présenté, il y a quelques années, la dérivation d'un modèle de type HMM dont les états génèrent ou évaluent non plus un caractère, mais directement un alignement de tels caractères, étant donné un arbre phylogénétique sous-jacent. Ces modèles ont démontré une efficacité certaine dans le cadre de la détermination de positions conservées au sein d'une séquence, ou d'autres applications similaires. Mais les auteurs n'ont présenté que des exemples de modèles avec un nombre relativement restreint d'états, souvent avec des paramètres *ad hoc*.

J'ai démarré ma thèse sous la direction d'Olivier Gascuel en octobre 2007. Après avoir suivi une formation de premier et second cycle en mathématiques et informatique, j'ai souhaité me tourner vers un domaine me permettant de mettre mes connaissances théoriques et pratiques au service des sciences du vivant. C'est donc assez naturellement que je me suis rapproché de l'équipe Méthodes et Algorithmes pour la Bioinformatique et de son directeur. L'argument de cette thèse était initialement donné par la volonté de combiner efficacement les deux classes de modèles introduites plus haut. Le travail d'Adam Siepel et de David Haussler [Siepel et Haussler, 2005] nous était apparu comme théoriquement séduisant, et nous nous proposions de travailler à utiliser ce genre de modèles pour prédire la fonction de gènes inconnus par des recherches de similarité dans les bases de données. En effet, l'équipe MAB pilotait alors le projet ANR « PlasmoExplore », centré sur le développement d'outils méthodologiques permettant de mieux comprendre l'organisme vecteur de la malaria, *Plasmodium falciparum*, dont une grande part du génome reste à l'heure actuelle fonctionnellement inexpliquée.

Ainsi, nous avons publié en 2008 un travail destiné à modéliser, à partir d'un alignement de séquences homologues et de l'arbre phylogénétique supportant celles-ci, la

séquence homologue se trouvant sur une tierce feuille de l'arbre [Domelevo Entfellner et Gascuel, 2008a]. Lors d'une communication la même année [Domelevo Entfellner et Gascuel, 2008b], nous avons découvert les travaux de Mitchison et Durbin [Mitchison et Durbin, 1995] et de Qian et Goldstein [Qian et Goldstein, 2003], proposant de créer non pas un mais plusieurs phylo-HMM à partir d'un seul couple constitué d'un alignement et d'une phylogénie. Après avoir fait un détour plutôt infructueux par les modèles de substitution site-dépendants, nous avons décidé d'approfondir les idées de Qian et Goldstein pour créer des HMM qui utilisent pleinement l'information phylogénétique afférente à un ensemble de séquences. À l'occasion d'un comité de suivi de thèse, Laurent Bréhélin et Vincent Ranwez ont suggéré que la structure même des HMM que nous appelions « phylogénisés » pouvait être choisie différente pour chacun des nœuds de l'arbre, en fonction des séquences évolutivement les plus proches. D'autres idées nouvelles sont venues par la suite, notamment celles tournant autour de la modélisation de l'évolution des longueurs d'insert, qui doivent beaucoup à des discussions avec Alain Jean-Marie. Nous avons également proposé une idée simple concernant les émissions sur les états d'insertion d'un HMM, qui doivent être apprises phylogénétiquement à partir de feuilles qui affichent non pas un caractère, mais une collection plus ou moins grande de caractères insérés.

Considérée dans sa globalité, l'approche que nous décrivons dans cette thèse consiste à introduire une part d'information phylogénétique dans le processus d'apprentissage de modèles stochastiques que sont les HMM profils. À partir d'un alignement de séquences et d'une phylogénie supportant ces dernières, nous déterminons chacun des paramètres du HMM profil comme étant le fruit d'une évolution le long des branches de l'arbre. La diversité des modèles ainsi reconstruits en chacun des nœuds de l'arbre phylogénétique nous permet de détecter des homologues distants en plus grand nombre que lorsque nous utilisons un seul HMM non « phylogénisé ». Les paramètres des phylo-HMM profils que nous construisons sont tous déterminés à partir d'une approche basée sur la phylogénie : choix des colonnes conservées, émissions de caractères sur les états Match et les états d'Insertion, transitions entre états du HMM. Après avoir démontré la pertinence d'une modélisation phylogénétique pour ces paramètres, nous décrivons en détail les processus d'inférence pour chacun de ceux-ci, en présentant notamment des processus d'évolution jamais utilisés à notre connaissance en biologie moléculaire pour ce qui concerne la modélisation de l'évolution des caractères représentant des transitions dans le HMM ou encore les longueurs d'insertion entre deux colonnes conservées.

L'apport le plus important de notre travail en termes de résultats concrets réside dans le choix d'une architecture de HMM orienté par la phylogénie (figure 1, étiquette 1). Ce choix permet de considérer des architectures alternatives mieux à même de modéliser un sous-groupe particulier des séquences d'apprentissage correspondant à un voisinage phylogénétique précis.

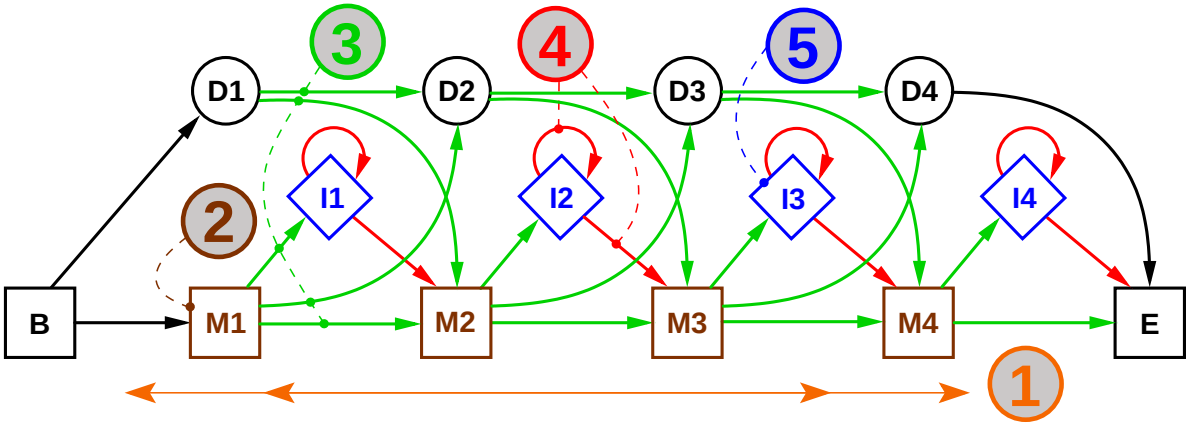


Figure 1. *Un HMM profil correspondant à l'architecture Plan7 utilisé par le logiciel* HMMER. *On décrit dans cette thèse une méthode de dérivation basée sur la phylogénie pour les cinq catégories de paramètres désignées ici : en (**1**) le nombre d'états Match et leur mise en correspondance avec des sites de l'alignement d'apprentissage, en (**2**) les probabilités d'émission sur les états Match, en (**3**) les transitions quittant les états Match et de Délétion, en (**4**) celles ayant pour origine un état d'Insertion, et enfin en (**5**) les émissions depuis les états d'Insertion.*

Alors que très peu d'auteurs ont jusqu'ici accepté de concevoir les transitions empruntées par les séquences dans les HMM profils (figure 1, étiquette 3) commme des caractères faisant l'objet d'une évolution le long des branches d'un arbre, l'étude de la phylogénisation des transitions quittant les états Match et de Délétion nous a conduits à proposer un modèle de substitution GTR (*General Time-Reversible*) pour ces transitions.

L'étude de la phylogénisation des transitions quittant les états d'Insertion (figure 1, étiquette 4) nous a amenés à nous pencher sur la question du calcul de vraisemblance pour les arbres phylogénétiques établis sur des caractères quantitatifs. L'objectif étant de calculer un paramètre statistique propre à la distribution des longueurs d'insertion engendrée par chacun des états d'Insertion du HMM, nous avons développé deux approches originales de résolution, la seconde faisant intervenir une modélisation de l'évolution de ces longueurs par un processus markovien de type Ornstein-Uhlenbeck.

Nous proposons également une méthode simple de prise en compte des caractères insérés entre deux colonnes conservées permettant un traitement phylogénétique des émissions sur les états d'Insertion (figure 1, étiquette 5).

Le choix du jeu de données sur lequel appliquer notre modèle pour obtenir des résultats significatifs s'est révélé délicat. La difficulté consistait à choisir un jeu de données

biologiques présentant des séquences évolutivement distantes (moins de 50% d'identité de séquence pour tout couple pris aléatoirement au sein d'une même famille) sur lequel on puisse disposer d'un alignement *et* d'une phylogénie suffisamment fiables pour que cette dernière reflète fidèlement l'évolution des caractères tels que l'alignement les présente. La tâche est plus compliquée qu'il n'y paraît, alors qu'on peut trouver des ensembles de séquences divergentes difficiles à aligner de façon pleinement satisfaisante (SCOP/ASTRAL) ou bien des séquences correctement alignées les unes aux autres mais parfois évolutivement trop proches (TreeFam, PhylomeDB) et souvent réunies dans des familles pléthoriques. Il nous fallait également avoir pour chacune des familles de test un ensemble bien défini de vrais positifs. Nous avons finalement retenu deux bancs de test différents. Le premier est le banc SABmark [Van Walle *et al.*, 2005], pour lequel nous nous sommes appuyés sur des méthodes d'alignement visant la cohérence pour obtenir un alignement multiple à partir d'alignements deux-à-deux de référence. Le second est le banc Treefam [Li *et al.*, 2006], qui fournit à la fois des alignements protéiques et les arbres correspondants, relus et corrigés par une équipe d'annotateurs. Nous pouvons ainsi montrer comment se comporte notre méthodologie sur deux jeux de données bien différents.

Dans un premier temps, nous présentons le contexte dans lequel s'insèrent nos travaux : après une brève introduction aux séquences biologiques et au problème de l'alignement multiple, nous montrons quels sont les modèles en vigueur pour représenter une famille de séquences homologues. Nous terminons notre état de l'art en indiquant quelles ont été les tentatives d'inclusion de l'information phylogénétique dans de tels modèles.

Nous consacrons le deuxième temps de cette thèse à l'exposition détaillée de notre démarche et donc des méthodes de dérivation des paramètres menant à nos modèles. Enfin, dans un troisième temps, nous évaluons l'efficacité de ces modèles sur les bancs de test choisis en déclinant nos tests selon les différents sous-problèmes auxquels notre méthodologie tente d'apporter une réponse, c'est-à-dire selon les différents items de la figure 1.

Première partie

État de l'art

CHAPITRE

1 Introduction aux séquences biologiques

Depuis la découverte de la structure de l'ADN en 1953 par Rosalind Franklin (auteure méconnue de plusieurs radiographies de l'ADN aux rayons X), Maurice Wilkins, James Watson et Francis Crick, le champ des sciences biologiques a connu une évolution spectaculaire. Aujourd'hui, une grande proportion des publications en biologie inclut une part d'étude des séquences. Compte tenu de la complexité et de la quantité de données à traiter, cette étude s'appuie aujourd'hui massivement sur des outils statistiques et algorithmiques, qu'il s'agisse par exemple de modélisation des réseaux d'interactions biologiques ou encore de recherche dans les bases de données de séquences. La puissance de calcul des ordinateurs entre donc en jeu, et c'est alors le champ de la bioinformatique qui se développe fortement depuis le début des années 1990. Notre travail s'inscrivant pleinement dans ce cadre, nous présentons dans cette introduction les objets biologiques sur lesquels opèrent les modèles que nous décrirons plus loin.

Sommaire

1.1 Dogme central de la biologie moléculaire

Alors que dès le dix-neuvième siècle les travaux de Lamarck (1744–1829), Darwin (1809–1882) et Mendel (1822–1884) laissaient entrevoir la possibilité d'une évolution des

espèces due à des facteurs extérieurs aux organismes vivants mais dont ceux-ci garderaient une trace imprimée d'une certaine façon *en eux*, c'est le vingtième siècle qui a apporté la découverte de ce fameux acide désoxyribonucléique (ADN) en 1953. Dès lors, les biologistes moléculaires et les évolutionnistes n'ont eu de cesse de se pencher sur la façon dont cet ADN (qui porte ce qu'on appelle le *génome*) s'exprime dans les cellules des organismes vivants. Très tôt, la communauté scientifique a décrit et adopté un schéma général pour l'expression de ce génome, en indiquant le chemin suivant : ADN → ARN → protéine (cf. figure 1.2).

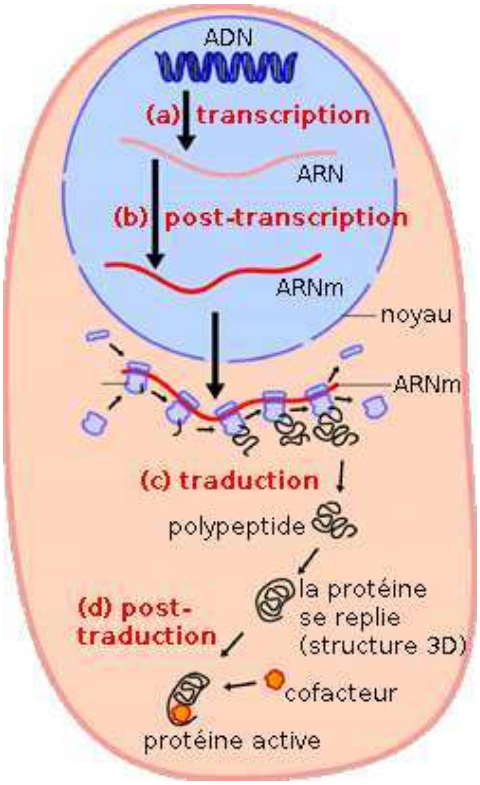


Figure 1.2. *Schéma canonique de biosynthèse des protéines (ici chez les eucaryotes), correspondant au « dogme central » de la biologie moléculaire. C'est à l'intérieur du noyau qu'a d'abord lieu l'activité de transcription (a) : l'ADN double brin est traduit en ARN pré-messager. En (b), l'activité de post-transcription consiste essentiellement à retirer du brin d'ARN pré-messager des zones appelées « introns », qui n'entreront pas dans le processus de traduction. Le produit de ce processus appelé « épissage » est libéré par le noyau : c'est l'ARN messager (ARNm). Lors de l'étape de traduction (c), l'ARN messager est lu par les ribosomes pour produire un polypeptide, c'est-à-dire une chaîne d'acides aminés. Après repliement et éventuellement activation par une molécule cofacteur (post-traduction, (d)), la protéine est active dans la cellule. Enfin, des modifications post-traductionnelles peuvent encore intervenir pour modifier la fonction de la protéine (méthylation, acétylation, etc).*

Ce schéma général, communément appelé « dogme central de la biologie moléculaire » [Crick, 1958, 1970], a été cependant fortement ébranlé depuis les années 1970. On sait maintenant qu'une grande partie du génome n'est pas traduite en protéines. Chez l'homme par exemple, si l'on estime [Birney *et al.*, 2007] qu'une fraction allant de 74% à 93% du génome est exprimée (ADN → ARN), seuls 1,1% à 1,5% passent l'étape de traduction ARN → protéine [Venter *et al.*, 2001; Lander *et al.*, 2001]. Ainsi, il ressort de l'état des connaissances actuelles qu'une grande partie des génomes sert essentiellement à la *régulation génique*, un ensemble de processus indispensables au fonctionnement des organismes : alors que l'ensemble de nos cellules dispose exactement du même patrimoine génétique, c'est la régulation qui fait que les cellules des fibres musculaires ne synthétisent

pas les mêmes protéines que celles du foie, pour ne citer que deux exemples de tissus. C'est encore la régulation qui modifie l'expression des gènes en fonction de paramètres dynamiques issus du milieu dans lequel se trouve la cellule (stress, signalisation extérieure, etc). Enfin, nous sommes loin d'avoir percé tous les mystères de l'information génétique, et de nombreuses portions des génomes restent à l'heure actuelle sans explication concrète. Par exemple, le caractère répétitif de toute une classe de portions de séquences (« éléments transposables », ou « transposons ») issues de réplications (copier-coller) ou de translocations (couper-coller) qui forment chez certains organismes la majeure partie du génome, soulève de nombreuses interrogations sans réponse à ce jour.

Néanmoins, malgré toutes les brèches ouvertes dans le dogme central de la biologie moléculaire, ce dernier donne un schéma opérationnel qui n'a jamais été fondamentalement remis en cause mais bien plutôt de nombreuses fois validé par l'expérience. Ainsi, pour beaucoup de chercheurs en biologie moléculaire, l'objet d'étude s'est déplacé du génome (ensemble des séquences nucléotidiques, ou « ADN ») vers les données d'expression (séquences ARN) ou encore vers le protéome (ensemble des protéines formant le patrimoine d'une espèce donnée). L'étude du protéome suffit par exemple à inférer des liens évolutifs entre espèces, à fabriquer des familles de séquences homologues, etc. Le protéome est dans cette thèse l'objet d'étude de référence, mais les modèles décrits ou développés opèrent avant tout sur des *séquences biologiques*, suites de lettres prises dans un alphabet de taille finie. Dans le cadre de ces modèles et sur le plan théorique, le fait que l'on parle de séquences d'ADN ou bien de protéines induit un changement d'alphabet : le génome est construit sur un alphabet à 4 lettres (les nucléotides), tandis que le protéome, issu du génome, est décrit par un alphabet de 20 lettres (chacune correspondant à un acide aminé). Dans ce qui suit, nous donnons brièvement quelques caractéristiques fondamentales des séquences génomiques et protéiques.

1.2 Séquences nucléotidiques, séquences protéiques : des suites de lettres qui font sens

1.2.1 Séquences nucléotidiques

L'acide désoxyribonucléique (ADN), avec sa structure hélicoïdale en double brin mise au jour en 1953, se compose de paires de *nucléotides*. Un nucléotide est une molécule complexe, typique dans sa composition de ce qu'on rencontre en chimie organique : il s'agit d'un assemblage d'atomes de carbone, d'oxygène, d'hydrogène, d'azote et de phosphore (plus précisément une baze azotée, un sucre et un groupe phosphate). Les nucléotides en usage dans les ADN du vivant sont au nombre de quatre : l'adénine (A), la cytosine (C), la guanine (G) et la thymine (T). La formule chimique brute pour la guanine telle qu'on la trouve dans l'ADN est par exemple $C_{10}H_{14}N_5O_7P$. Les deux membres d'une paire de

nucléotides dans l'ADN double brin sont situés chacun sur un brin, et les appariements possibles sont au nombre de deux (si l'on n'ordonne pas les brins) : A-C et G-T.

L'acide ribonucléique (ARN) est aussi une suite de nucléotides, mais l'uracile (U) y remplace la thymine. À cette différence près, la molécule d'ARN est donc une copie « en négatif » d'une portion de l'un des brins d'une molécule d'ADN double brin : l'étape de *transcription* présentée en figure 1.2 est donc une étape de quasi recopiage.

1.2.2 Séquences protéiques

Les *protéines* sont véritablement les « machines à tout faire » du vivant. Lorsqu'elles ne participent pas à la *structure* des tissus, ces molécules complexes accomplissent un nombre incalculable de tâches qui forment ce que l'on appelle les circuits métaboliques du vivant, c'est-à-dire l'ensemble des actions de transformation de l'énergie et du matériel moléculaire au sein de la cellule. Les protéines sont tantôt catalyseurs, tantôt transmetteurs, tantôt effecteurs, etc. On peut citer en guise d'exemple un certain nombre de ces tâches, avec quelques-unes des protéines qui en sont responsables :

1. catalyse de réactions (enzymes),
2. liaison avec un ligand (récepteurs),
3. signalisation intra- (protéines kinases) et intercellulaire (hormones peptidiques, récepteurs d'hormones),
4. structure des cellules (kératine, collagènes),
5. motricité des tissus (kinésine, myosine).

Toutes les protéines peuvent être considérées du point de vue de ce que l'on appelle leur structure primaire, c'est-à-dire comme une suite d'*acides aminés*. Les acides aminés sont des molécules (exemple : la méthionine, de formule brute $C_5H_{11}NO_2S$) que les êtres vivants sont capables de synthétiser et/ou d'assimiler. Les acides aminés sont au nombre de 20 (on choisit ici d'ignorer la pyrrolysine et la sélénocystéine, spécifiques à certaines protéines ou à certains groupes du vivant), et ces 20 molécules sont les briques permettant de fabriquer *toutes* les protéines que l'on rencontre dans le règne du vivant. La table 1.1 donne leur nom français, leur trigramme international et leur lettre, tout aussi internationalement standardisée.

Lors de la phase de *traduction* représentée en figure 1.2, chaque triplet de nucléotides consécutifs lu sur le brin d'ARN messager est traduit en un acide aminé selon un code génétique quasiment universel, à quelques exceptions près. On reproduit ce codage en table 1.2.

Alanine	Ala	A
Cystéine	Cys	C
Acide aspartique (aspartate)	Asp	D
Acide glutamique (glutamate)	Glu	E
Phénylalanine	Phe	F
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Lysine	Lys	K
Leucine	Leu	L
Méthionine	Met	M
Asparagine	Asn	N
Proline	Pro	P
Glutamine	Gln	Q
Arginine	Arg	R
Sérine	Ser	S
Thréonine	Thr	T
Valine	Val	V
Tryptophane	Trp	W
Tyrosine	Tyr	Y

Table 1.1. *Les 20 acides aminés du vivant : noms, trigrammes et lettres standard*

La chaîne de nucléotides de l'ARN messager, coupée en des sites bien particuliers (codons start et codons stop), engendre ainsi ces chaînes d'acides aminés que sont les protéines. La taille varie beaucoup d'une protéine à l'autre, typiquement entre 20 et 20.000 acides aminés (avec presque 30.000 acides aminés de long, la titine, protéine élastique du muscle strié, est la plus grande protéine connue). La structure primaire d'une protéine étant sa description linéaire comme suite de lettres d'un alphabet à vingt caractères, sa structure secondaire décrit l'assemblage de structures élémentaires (hélices alpha, feuillets bêta et boucles) constituant la protéine (cf. figure 1.3 et plus loin, section 1.3). Enfin, la structure ternaire décrit la conformation tridimensionnelle adoptée par la protéine lorsqu'elle se trouve dans le cytoplasme ou dans les composants de la cellule, c'est-à-dire *in vivo*. Les méthodes de cristallographie ou de résonance magnétique nucléaire (RMN) qui permettent d'établir la structure tridimensionnelle d'une protéine étant coûteuses en temps et en argent, la communauté dispose de relativement peu de ces stuctures : 74.601 structures recensées au 19 juillet 2011 dans la base de données « Protein Data Bank » (http://www.pdb.org) contre 530.264 séquences dans la base de données UniProtKB/Swiss-Prot (séquences vérifiées par l'expertise scientifique) ou encore

Première lettre	Deuxième lettre U		Deuxième lettre C		Deuxième lettre A		Deuxième lettre G		Troisième lettre
U	UUU	phénylalanine	UCU	sérine	UAU	tyrosine	UGU	cystéine	U
	UUC		UCC		UAC		UGC		C
	UUA	leucine	UCA		UAA	codons stop	UGA	codon stop	A
	UUG		UCG		UAG		UGG	tryptophane	G
C	CUU	leucine	CCU	proline	CAU	histidine	CGU	arginine	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	glutamine	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	isoleucine	ACU	thréonine	AAU	asparagine	AGU	sérine	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	lysine	AGA	arginine	A
	AUG	méthionine	ACG		AAG		AGG		G
G	GUU	valine	GCU	alanine	GAU	acide aspartique	GGU	glycine	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	acide glutamique	GGA		A
	GUG		GCG		GAG		GGG		G

Table 1.2. *Le code génétique universel : traduction de l'ARN en acides aminés*

16.014.674 séquences dans la base UniProtKB/TrEMBL (séquences vérifiées ou prédites par des méthodes automatiques d'annotation) [1]. Bien que de nombreuses propriétés d'une protéine découlent de sa structure tridimensionnelle, sa représentation unidimensionnelle sous la forme d'une suite d'acides aminés fournit déjà un aliment suffisant pour nombre d'études comparatives. C'est cette réprésentation qu'utilisent les modèles que nous décrivons ici. Cette approche permet d'inférer des histoires évolutives ou encore de repérer des similarités même ténues entre protéines apparentées, à la condition cependant de disposer de modèles suffisamment fins. Le but de cette thèse est d'exposer de tels modèles, s'appuyant à la fois sur la représentation linéaire des protéines comme enchaînements d'acides aminés, et sur l'histoire évolutive sous-jacente à toute famille de protéines homologues.

1.2.3 Propriétés physico-chimiques des acides aminés

Les vingt acides aminés introduits plus haut, s'ils permettent de construire toute la diversité des protéines du vivant, ne sont évidemment pas interchangeables aveuglément

1. Ces deux chiffres sont relatifs à la version de juillet 2011 de la base de données UniProtKB (http://www.uniprot.org)

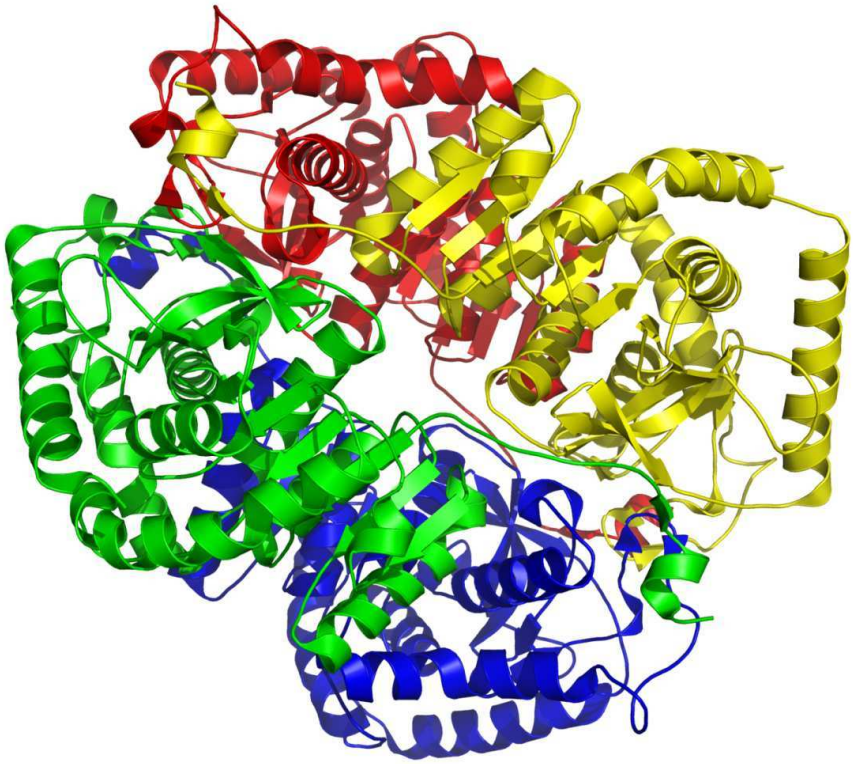


Figure 1.3. *Structures secondaires d'une molécule de lactate déshydrogénase (ici issue de tissus musculaires chez l'homme). Les quatre couleurs correspondent aux quatre sous-unités de la molécule. Les hélices alpha sont représentées sous la forme de ressorts tandis que les feuillets bêta sont figurés par des flèches plates. Le restant, ici en « fil de fer », constitue ce que l'on appelle les boucles, c'est-à-dire des zones avec peu de contraintes de structure.*

les uns aux autres : ces molécules possèdent des propriétés physico-chimiques qu'elles partagent ou non entre elles, formant ainsi des sous-groupes chevauchants. Parmi les différentes propriétés en question, on peut citer :

- l'hydrophobicité. Les acides aminés *hydrophobes* occupent plutôt l'intérieur de la structure d'une protéine. Comme leur nom l'indique, ils ont tendance à fuir le contact avec tout milieu aqueux. S'ils sont en surface, ils constituent des points d'adhérence,
- la polarité. Les acides aminés *polaires* posèdent une orientation naturelle qui les rend capables de former des barrières à la frontière entre deux milieux, et leur confère une bonne solvabilité dans les milieux polaires (par exemple l'eau). Ils tendent donc à

occuper la surface des protéines. Certains d'entre eux sont chargés électriquement, c'est-à-dire que leur chaîne latérale présente un défaut ou un excès d'électron(s),
– l'aromaticité. Les acides aminés *aromatiques* sont caractérisés par la présence d'un ou deux cycle(s) benzène (c'est-à-dire à 6 atomes de carbone), possiblement hétérocycliques comme l'est l'histidine, où deux atomes d'azote apparaissent dans le cycle.

On peut voir en figure 1.4 l'ensemble des acides aminés regroupés en fonction de leurs différentes propriétés [Taylor, 1986]. Ces propriétés et la nature des interactions entre protéines font que, au cours de l'évolution, un événement de substitution entre deux acides aminés donnés se trouve plus ou moins probable. Cette constatation est essentielle pour ce qui nous concerne, car elle permet d'envisager les séquences biologiques non pas comme des séquences aléatoires, mais comme des séquences consituées à la fois en fonction d'une histoire *évolutive* et de pressions de sélection *structurelles* dues aux différents rôles joués par les protéines synthétisées dans les organismes vivants. Cette représentation duale des séquences biologiques est au cœur du travail que nous exposons dans ce manuscrit.

1.3 Structures, domaines et bases de données protéiques

Une chaîne d'acides aminés n'est pas un ensemble linéaire dépourvu de structure. Les interactions atomiques entre les différents résidus (on désigne par « résidu » les groupements d'atomes qui différencient un acide aminé d'un autre acide aminé, et par extension ce terme indique l'acide aminé lui-même, pris en tant que caractère issu d'une protéine) qui composent cette chaîne tendent à lui faire adopter des *structures secondaires* bien particulières. La chaîne d'acides aminés s'organise parfois localement en spiralant (on appelle « hélice alpha » une telle structure secondaire), parfois en formant une nappe plutôt plane (on parle de « feuillet bêta »). En certains endroits de la chaîne, la structure secondaire n'est pas clairement définie et n'est pas assimilable à l'une des deux catégoriés précitées. On parle alors de « boucle ». Ces boucles sont en général mises à profit lorsque la protéine prend sa conformation tridimensionnelle, pour opérer un « coude », c'est-à-dire un changement de direction de la chaîne permettant d'obtenir une protéine relativement ramassée dans l'espace. Dans la représentation tridimensionnelle de la protéine de lactate déshydrogénase que nous donnons en figure 1.3, on lit bien les hélices (sortes de tire-bouchons), les feuillets (représentés par des flèches) et les boucles (en fil de fer).

On vient d'évoquer ci-dessus le fait que les protéines prennent une conformation tridimensionnelle bien particulière *in vivo*. Cette conformation est en lien direct avec la fonction de la protéine, et fait ressortir un certain nombre de zones « actives » qui permettent effectivement de réaliser la fonction en question. Il s'agira par exemple d'un

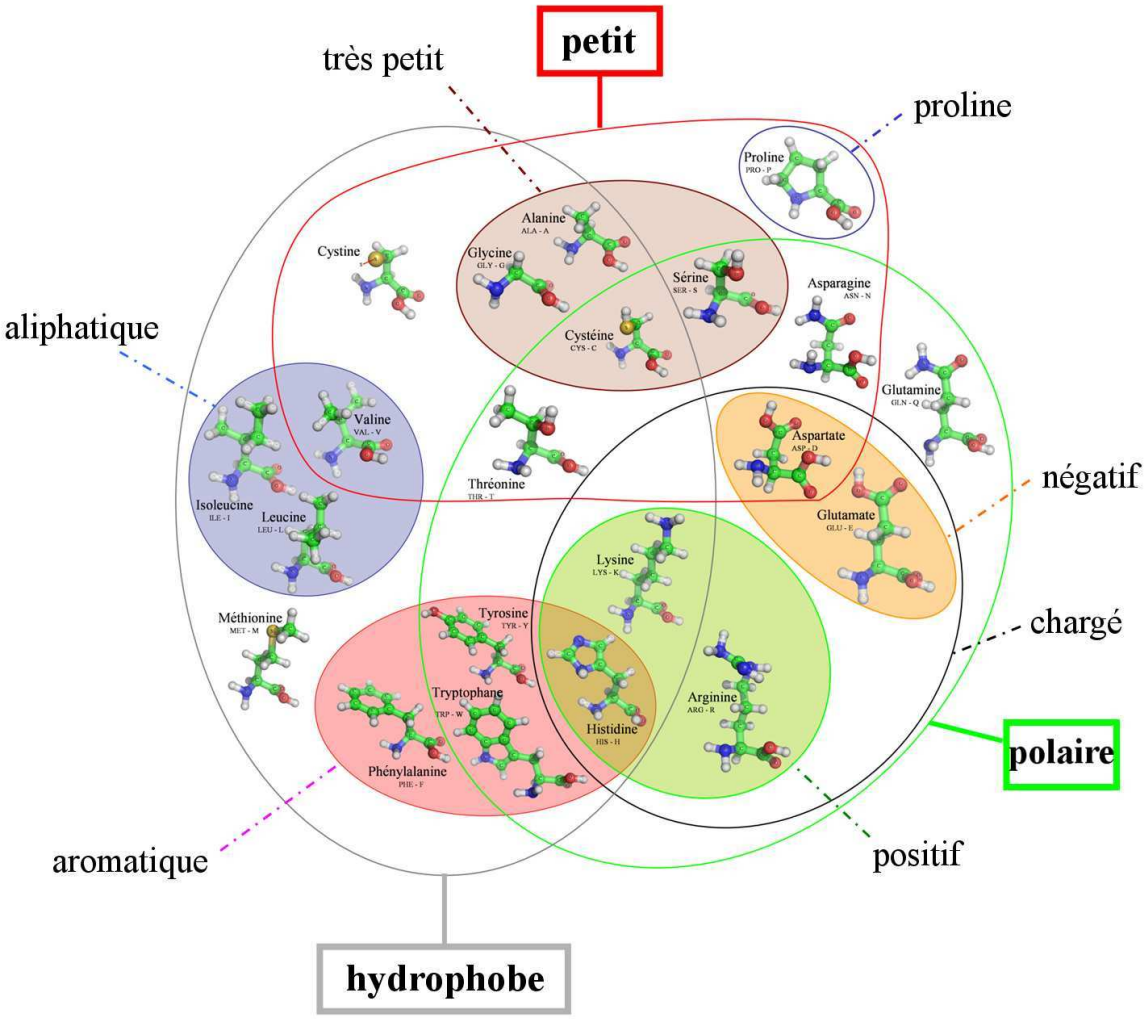


Figure 1.4. *Molécules d'acides aminés représentées dans un modèle « boules et bâtons » et caractérisées par leurs différentes propriétés (classification issue des travaux de Dayhoff [Dayhoff* et al.*, 1978] et de Taylor [Taylor, 1986]). Les atomes carbone sont représentés en vert, l'azote en bleu, l'oxygène en rouge, le soufre en jaune et l'hydrogène en gris.*

assemblage de quelques acides aminés formant un récepteur sur lequel viendra s'arrimer une hormone. Un tel assemblage, qui se trouve actif dans certaines situations pour réaliser une fonction biologique, est appelé *site actif*. Il est à noter que ces assemblages courts sont souvent très bien conservés, c'est-à-dire se retrouvent *à l'identique* chez toutes les protéines ayant cette même fonction, car sur une si petite combinaison d'acides aminés une mutation entraîne souvent la diminution de l'efficacité de la fonction, voire sa disparition.

Les sites actifs, aussi appelés « motifs protéiques », sont donc les éléments essentiels des unités fonctionnelles de la protéine. Cependant ces unités fonctionnelles sont plus étendues que les sites actifs eux-mêmes, parce qu'un site actif a besoin d'un *contexte* pour être fonctionnellement efficace (par exemple, il « faut faire en sorte » qu'un site agissant comme récepteur se trouve nécessairement placé en surface de la structure tridimensionnelle de la protéine). L'unité fonctionnelle et structurale prise dans son ensemble est appelée « domaine protéique ».

Les différentes fonctions propres à une protéine étant réalisées par ces domaines protéiques, ces derniers constituent un support efficace pour repérer des similarités entre protéines. En effet, les études comparatives montrent que les domaines protéiques ont tendance à être bien conservés au cours de l'évolution des séquences protéiques. C'est ainsi que, pour faciliter les études comparatives et pour permettre la reconnaissance de protéines homologues, les décennies 1990 et 2000 ont vu fleurir les bases de données recensant les domaines protéiques, en donnant des signatures statistiques et regroupant les protéines sur la base de la présence partagée de tels domaines. On peut citer par exemple les bases SCOP [Murzin *et al.*, 1995] (http://scop.mrc-lmb.cam.ac.uk/scop/), Pfam [Finn *et al.*, 2010] (http://pfam.sanger.ac.uk/), Prosite [Sigrist *et al.*, 2010] (http://prosite.expasy.org/), ProDom [Servant *et al.*, 2002] (http://prodom.prabi.fr/), CATH [Pearl *et al.*, 2003] (http://www.cathdb.info/) ou encore Panther [Thomas *et al.*, 2003b] (http://www.pantherdb.org/).

Les chaînes d'acides aminés que sont les protéines sont donc des ensembles *structurés*, dans lesquels on peut délimiter différentes zones fonctionnellement signifiantes. Ces zones sont d'une importance particulière d'abord évidemment parce qu'elles déterminent la fonction d'une protéine, mais également parce qu'elles sont en général plus fortement conservées que le reste des séquences, ce qui en fait des points d'ancrage naturels pour les processus d'alignement que nous allons considérer au chapitre suivant. Parmi les bases que nous avons citées ci-dessus, Pfam est sans conteste l'une des plus connues et des plus utilisées, notamment dans les projets d'annotation des génomes de l'EMBL (European Molecular Biology Laboratory), organisme de recherche européen multi-sites. La spécificité de Pfam est d'avoir été la première base conçue à partir de modèles statistiques markoviens décrivant les séquences, ces « HMM profils » dont nous parlerons plus loin (section 3.3). L'enjeu de la détermination automatique des positions d'intérêt (sites actifs et plus largement domaines protéiques) au sein des séquences et de la modélisation du contenu de celles-ci en termes d'acides aminés est central dans la question des HMM profils, des modèles qui servent de fondement à ceux que nous exposons ici.

Avant de nous pencher sur les modèles permettant de représenter les familles de protéines apparentées, le chapitre suivant nous amène à examiner les voies et moyens per-

mettant de rendre compte de manière lisible des correspondances entre celles-ci, en les *alignant* les unes aux autres.

CHAPITRE

2 Aligner des séquences homologues

On dit d'un ensemble de séquences biologiques (ADN ou protéines) qu'elles sont *homologues* (nous dirons aussi parfois *apparentées*) lorsque les unes et les autres sont issues de la même séquence ancestrale. Nous constatons alors qu'elles *se ressemblent*. Les aligner, c'est alors repérer des correspondances entre les caractères des unes et des autres, de manière à pouvoir ensuite analyser les similarités et les différences entre les séquences. Il s'agit donc là d'une première étape indispensable pour qui souhaite ensuite construire des modèles descriptifs pour un groupe de séquences apparentées. Nous exposons ci-après quelques-unes des techniques courantes pour l'alignement de séquences biologiques.

Sommaire

Depuis que l'on sait obtenir par des moyens dits de *séquençage* la suite de caractères nucléotidiques composant un brin d'ADN, les scientifiques se sont rendu compte que deux séquences très semblables l'une à l'autre mais prises chez deux espèces différentes, sont bien souvent proches du point de vue fonctionnel : une homologie de séquence induit des fonctions biologiques similaires. Considérons par exemple la protéine de la phényléthanolamine-N-méthyltransférase, abréviée PNMT. Il s'agit d'une protéine qui intervient de façon essentielle dans le chemin métabolique de biosynthèse de l'adrénaline [1]. On représente en figure 2.1 une partie de la séquence de PNMT chez l'homme et chez le

1. L'épiphédrine, ou adrénaline, est vitale chez tous les animaux car sa libération rapide dans des situations de stress lié à un danger entraîne différents effets, dont l'accélération de la fréquence cardiaque, la

chien : on peut constater le fort taux d'identité entre les deux séquences.

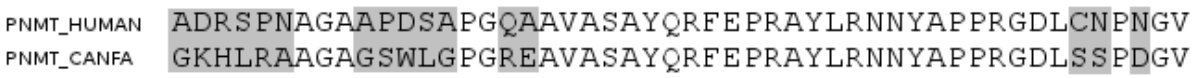


Figure 2.1. *Une fraction de la séquence protéique de la phényléthanolamine-N-transférase chez l'homme (positions 4 à 53 de la séquence UniProtKB P11086) représentée au-dessus de son homologue chez le chien (positions 7 à 56 de F1PW89 selon UniProtKB). Les résidus non identiques chez les deux espèces sont représentés sur fond grisé.*

Ayant constaté cette caractéristique des séquences biologiques, et en accord avec la théorie darwinienne de l'évolution (cf. chapitre 4), on énonce que deux séquences « homologues » sont « alignables », en ceci qu'on peut établir des correspondances injectives entre les caractères de l'une et les caractères de l'autre (l'injectivité impose que deux caractères distincts d'une séquence A ne peuvent correspondre au même caractère de la séquence B. Par contre, des caractères de A peuvent tout à fait se trouver sans correspondance dans B et vice-versa). Bien souvent les caractères entre lesquels on établit ces correspondances sont simplement identiques deux à deux, mais il arrive que ce ne soit pas le cas. Par exemple, les acides aminés A et G, mis l'un en face de l'autre aux positions 1, 10 et 14 de la figure 2.1 partagent plusieurs propriétés physico-chimiques : ce sont tous deux des acides aminés de très petite taille, non polaires et hydrophobes. On postule donc que sur le chemin qui relie l'homme au chien dans l'arbre du vivant, il s'est produit une *substitution* de l'alanine par la glycine ou vice-versa. C'est pourquoi on produit légitimement des *alignements* de séquences, dans lesquels on représente sur la même colonne des acides aminés appariés.

En théorie, deux acides aminés alignés l'un avec l'autre partagent le même rôle fonctionnel et structurel au sein de la protéine *et* sont issus du même acide aminé ancestral. On verra plus loin que dans les alignements courants, dans certaines zones où la fiabilité de l'alignement est contestable, ou bien lorsque la distance évolutive entre les séquences est trop importante, cette double condition peut ne pas être systématiquement respectée.

Bien que l'on soit souvent en mesure d'aligner « à l'œil » des séquences proches, la communauté scientifique s'est très tôt penchée sur le développement d'outils algorithmiques permettant de rechercher automatiquement le « meilleur » alignement entre deux séquences (alignement dit *pairwise* ou deux à deux) ou entre plusieurs séquences (alignement multiple). C'est le paysage formé par ces outils que nous invitons le lecteur à découvrir dans ce chapitre.

hausse de la pression artérielle, la dilatation des bronches et des pupilles, etc. : toutes choses utiles pour que l'individu soit en capacité d'affronter efficacement un danger imminent.

Gaps

Avant d'aller plus loin, il nous faut introduire la notion de « trous », ou *gaps* (dans la suite nous employons souvent directement le terme anglais sans autre forme de procès, parce qu'il est peut-être plus esthétique et surtout d'usage largement répandu dans la communauté, y compris francophone). Les événements mutationnels d'origine évolutive ne sont pas seulement des substitutions de caractères, mais parfois l'évolution sélectionne positivement la suppression ou l'ajout d'un ou de plusieurs caractères consécutifs dans une séquence. Ces événements évolutifs induisent la nécessité de représenter dans un alignement « du vide » en face de résidus qui existaient dans la séquence ancestrale et n'existent plus dans la séquence contemporaine, ou vice-versa. Cet espace est occupé par des caractères tirets '-' appelés gaps, qu'on trouve donc immanquablement dans tout alignement complet contenant des séquences ayant suffisamment divergé évolutivement les unes par rapport aux autres.

2.1 Des matrices de score de similarité entre caractères

Si l'on devait aligner entre elles les séquences `AACWK` et `AAYK`, quel serait l'alignement le plus plausible, de $\begin{smallmatrix}\mathrm{A}&\mathrm{A}&\mathrm{C}&\mathrm{W}&\mathrm{K}\\ \mathrm{A}&\mathrm{A}&\mathrm{Y}&-&\mathrm{K}\end{smallmatrix}$ ou de $\begin{smallmatrix}\mathrm{A}&\mathrm{A}&\mathrm{C}&\mathrm{W}&\mathrm{K}\\ \mathrm{A}&\mathrm{A}&-&\mathrm{Y}&\mathrm{K}\end{smallmatrix}$? Répondre à cette question suppose d'avoir établi une mesure de similarité entre acides aminés. C'est pourquoi on a développé dès le début des années 1970 de telles matrices.

2.1.1 Matrices PAM

Dans le monde des séquences protéiques, la première famille de matrices à voir le jour fut la famille PAM (pour *Point Accepted Mutation* ou *Percent of Accepted Mutations*), publiée en 1978 par Margaret Dayhoff, R.M. Schwartz et B.C. Orcutt [Dayhoff *et al.*, 1978]. La stratégie de mise au point de ces matrices est la suivante.

Les données utilisées par les auteurs sont constituées d'un grand nombre de paires de séquences très proches, alignées deux-à-deux. Les alignements en question ne souffrent d'aucune ambiguïté puisque les paires en question sont choisies de telle façon qu'en moyenne on ait une mutation toutes les 100 positions : 99% des sites alignés mettent en regard deux acides aminés identiques. Une telle démarche est justifiée par le désir des auteurs de dériver leur matrice PAM à partir d'un ensemble de mutations *directes* $a \rightarrow b$: on suppose dans ce qui suit qu'un alignement d'un acide aminé a contre un acide aminé $b \neq a$ dans le jeu de données employé indique une substitution non médiée par un acide aminé intermédiaire (qui serait du type $a \rightarrow c \rightarrow b$).

Sur tous les couples de séquences alignées l'une à l'autre, on dénombre alors dans une matrice A de taille 20×20 le nombre de fois où l'on observe une substitution de

l'acide aminé a vers l'acide aminé b, ce pour tous les couples (a,b). Les substitutions ne sont pas orientées : lorsqu'on observe un alignement entre a et b, on ajoute 1 au terme A_{ab} et également 1 à A_{ba}. Pour estimer la probabilité que a se transforme en b en un temps évolutif caractéristique du jeu de données, il faut encore réaliser une opération de normalisation en divisant les termes de A par la fréquence d'apparition de l'acide aminé de départ dans le jeu de données. On calcule donc une deuxième matrice B selon : $B_{ab} = \Pr(a \to b | a \text{ mute}) = A_{ab} / \sum_{c \neq a} A_{ac}$.

Quelle est la probabilité de voir l'acide aminé a muter ? Soit $f_a = \sum_{c \neq a} A_{ac}$ le nombre total de mutations observées impliquant l'acide aminé a. $f = \sum_a f_a$ est donc le double du nombre total de mutations observées (on rappelle que la mutation impliquant les acides aminés a et b est comptabilisée doublement, une fois dans A_{ab} et l'autre dans A_{ba}). La loi de Bayes nous donne alors :

$$\Pr(\text{mutation}|a) = \frac{\Pr(a|\text{mutation})\Pr(\text{mutation})}{\Pr(a)} = \frac{\left(\frac{f_a}{f}\right)\frac{1}{100}}{\Pr(a)} = \frac{f_a}{100\, f \Pr(a)} \tag{2.1}$$

La probabilité d'une mutation $a \to b$ dans un laps de temps caractéristique du jeu de données employé est donc :

$$\Pr(a \to b) = \Pr(\text{mutation}|a)\Pr(a \to b | a \text{ mute}) = \frac{f_a}{100\, f \Pr(a)} \frac{A_{ab}}{f_a} = \frac{A_{ab}}{100\, f \Pr(a)} \tag{2.2}$$

Enfin, comme on le verra par la suite avec les matrices BLOSUM, ce qu'on appelle en général *score d'alignement* entre les acides aminés a et b est une quantité qui mesure combien l'association (a,b) est plus probable que ce qu'on pourrait attendre d'une situation dans laquelle les appariements entre acides aminés seraient le fruit du hasard. Si $q_a = \Pr(a)$ représente simplement la proportion de caractères a rencontrés dans le jeu de données (telle que $\sum_a q_a = 1$), alors la probabilité d'observer les acides a et b appariés l'un à l'autre serait donnée dans l'hypothèse « hasard » par le produit $q_a q_b$. Soit $p_{ab}(t)$ la probabilité de l'association (a,b) correspondant à un temps d'évolution caractéristique t : $p_{ab}(t) = q_a \Pr(a \xrightarrow{t} b) = q_b \Pr(b \xrightarrow{t} a)$. En utilisant le logarithme du ratio on obtient alors finalement l'expression du *score de similarité* recherchée :

$$s_t(a,b) = \log\left(\frac{p_{ab}(t)}{q_a q_b}\right) = \log\left(\frac{\Pr(a \xrightarrow{t} b)}{q_b}\right)$$

Ainsi, par construction, les scores issus de la matrice PAM dérivée ci-dessus (nous l'appelons PAM_1) correspondent à la durée d'évolution t qui sépare deux séquences entre lesquelles il se produit en moyenne une substitution tous les 100 acides aminés, c'est-à-dire des séquences avec un taux de substitution moyen de 1%. Les matrices de score pour les séquences partageant un taux d'identité moindre sont simplement extrapolées à

partir de la matrice PAM_1, par exponentiation : $\text{PAM}_N = (\text{PAM}_1)^N$. La matrice PAM_{250}, elle, représente les probabilités de mutation entre acides aminés lorsqu'on a en moyenne 2,5 événements de substitution par site (certains pouvant être silencieux, c'est-à-dire du type $a \to a$ ou $a \to b \to a$).

2.1.2 Matrices BLOSUM

En 1992, Steven et Jorja Henikoff ont publié une nouvelle stratégie de construction de matrices de scores, appelée BLOSUM [Henikoff et Henikoff, 1992]. En guise d'argument pour défendre leur approche, les auteurs remarquent que les matrices PAM ont le défaut d'être toutes construites sur la base d'alignements entre séquences proches, avec un taux d'identité systématiquement supérieur à 85 %. Cette approche elle présente l'inconvénient d'inférer les substitutions observables dans les jeux de données avec un fort taux de divergence à partir de substitutions observées sur des jeux de données avec un faible taux de divergence. Entre la fin des années 1970 et le début des années 1990, les données biologiques étaient devenues disponibles en très grand nombre et des alignements distants commençaient à apparaître, rendant possible un changement de paradigme.

Steven et Jorja Henikoff ont procédé à partir d'un grand nombre de blocs alignés, issus de séquences protéiques diverses (base de données BLOCKS, plus de 2000 blocs sans gaps répartis en plus de 500 groupes de protéines apparentées, en tout plus de 15 millions de paires de résidus alignés). Leur idée centrale fut de regrouper les séquences en *clusters* selon leur degré d'identité de séquence. Ce processus se fait de façon aggrégative, avec simple lien :

1. on commence par définir un *seuil* (s%), fixé pour toute la suite du processus de construction de la matrice de score et qui caractérisera celle-ci *in fine*,
2. on agrège progressivement toutes les séquences dans différents clusters, jusqu'à ce que les clusters ne croissent plus en taille : une séquence isolée A rejoint un cluster $\mathscr{C}$ dès lors qu'il se trouve dans $\mathscr{C}$ *au moins une séquence* qui présente plus de s% d'identité de séquence avec A,
3. les paires d'acides aminés observées sur une même colonne dans un même cluster totalisent un poids unitaire : elles ne comptent que pour une paire. On diminue ainsi sérieusement l'importance des substitutions observées entre séquences plus proches que s% d'identité.

Les matrices BLOSUM sont ensuite construites sur la base des comptages de paires alignées, lesquels donnent des probabilités p_{ab} de voir les caractères a et b alignés l'un à

l'autre. Les scores BLOSUM sont simplement :

$$s_{ab} = \log_2\left(\frac{p_{ab}}{q_a q_b}\right)$$

Les termes de la matrice $S = \{s_{ab}\}$ sont enfin multipliés par un facteur constant puis arrondis à l'entier le plus proche.

Les matrices les plus utilisées sont la matrice BLOSUM62 (représentée en figure 2.2) et la matrice BLOSUM50 (pour laquelle les clusters sont plus large et qui représente donc mieux les substitutions ayant lieu entre des séquences plus divergentes).

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	-1	5																		
Asn	-2	0	6																	
Asp	-2	-2	1	6																
Cys	0	-3	-3	-3	9															
Gln	-1	1	0	0	-3	5														
Glu	-1	0	0	2	-4	2	5													
Gly	0	-2	0	-1	-3	-2	-2	6												
His	-2	0	1	-1	-3	0	0	-2	8											
Ile	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Figure 2.2. *La matrice de score BLOSUM62. Sa propriété de symétrie nous permet de ne représenter ici que la demi-matrice inférieure gauche. Les unités sont des demi-bits (c'est-à-dire que le facteur multiplicatif vaut 2). Les acides aminés rares ont un poids (i.e. autoscore) important. Par exemple, W-W score 11 alors que I, beaucoup plus courant, score 4. Un score positif signifie que les acides aminés en question sont fortement interchangeables (par exemple I-L, +4) alors que les associations rares dans un alignement reçoivent un score négatif (e.g. P-W, -4).*

2.2 Aligner deux séquences

Parce qu'elles résument les spécificités physico-chimiques des acides aminés, les matrices de score sont un outil indispensable pour aligner deux séquences l'une à l'autre. Plusieurs algorithmes ont été développés qui utilisent ces matrices afin de calculer de manière automatique (c'est-à-dire non supervisée) l'alignement de plus fort score.

2.2.1 Algorithme de Needleman et Wunsch

Reprenons le problème brièvement énoncé au début de ce chapitre. Il s'agit d'aligner les deux séquences `AACWK` et `AAYK` (nous appelons s_1 la première et s_2 la seconde). Pour ce faire, on peut procéder en examinant en parallèle l'une et l'autre de ces séquences, en progressant de gauche à droite pour construire l'alignement. À chaque pas de l'algorithme, on peut effectuer au choix l'un des trois mouvements suivants :

- consommer le caractère courant de s_1 et le caractère courant de s_2, les décréter alignés l'un à l'autre et les placer comme tels en les ajoutant à la fin de l'alignement en cours de construction,
- consommer le caractère courant de s_1 sans toucher à s_2. On ajoute à la fin de l'alignement en cours de construction le caractère consommé, aligné contre un gap dans la ligne correspondant à s_2,
- consommer le caractère courant de s_2 sans toucher à s_1. On ajoute à la fin de l'alignement en cours de construction le caractère consommé, aligné contre un gap dans la ligne correspondant à s_1.

En progressant ainsi, on construit un alignement. Lorsqu'une des deux séquences vient à épuisement, si l'autre aussi est simultanément tarie alors on a terminé, sinon on doit aligner tous les caractères non consommés de la séquence non vide contre des gaps, et on a ensuite terminé.

Un tel mécanisme se représente logiquement dans un tableau. On peut voir en figure 2.3 les chemins correspondant à l'alignement $\begin{smallmatrix}\mathrm{A}&\mathrm{A}&\mathrm{C}&\mathrm{W}&\mathrm{K}\\ \mathrm{A}&\mathrm{A}&\mathrm{Y}&-&\mathrm{K}\end{smallmatrix}$ (noir et bleu) et à $\begin{smallmatrix}\mathrm{A}&\mathrm{A}&\mathrm{C}&\mathrm{W}&\mathrm{K}\\ \mathrm{A}&\mathrm{A}&-&\mathrm{Y}&\mathrm{K}\end{smallmatrix}$ (noir et rouge).

Le problème des gaps se traite classiquement en affectant une pénalité (score négatif) à l'ouverture d'un gap et une autre pénalité, moins forte, pour la prolongation d'un gap. Avec cette technique, un gap présente un coût affine, fonction de sa longueur. Cette modélisation se base sur le fait que l'ouverture d'un gap est plus rare dans les séquences biologiques que l'extension d'une zone de gaps contigus déjà existante. Dans [Henikoff et Henikoff, 1992], les auteurs proposent un score de -12 pour l'ouverture d'un gap (c'est-à-dire pour le premier caractère aligné contre un caractère '-') et de -4 pour l'extension de gap (c'est-à-dire pour chacun des caractères suivants). L'algorithme d'alignement inté-

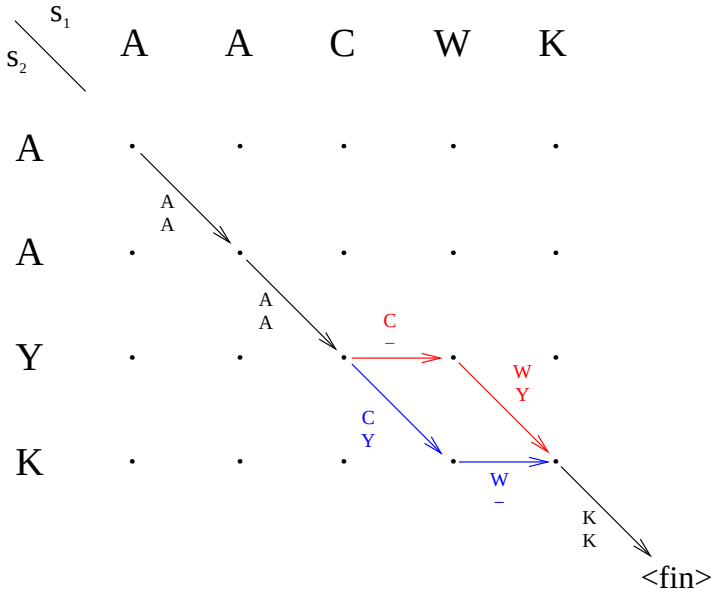


Figure 2.3. *Deux chemins d'alignement (noir/bleu ou noir/rouge) correspondant à l'alignement de s_1 avec s_2.*

grant de tels gaps avec pénalités affines est dû à Osamu Gotoh [Gotoh, 1982].

Nous avons maintenant tous les outils pour calculer le score total d'alignement correspondant à un chemin dans un tableau tel que celui représenté en figure 2.3 : il suffit de sommer les différents scores ou pénalités accumulés le long du chemin (cf. figure 2.4).

L'algorithme de Needleman-Wunsch proprement dit [Needleman et Wunsch, 1970] propose une méthode efficace pour explorer l'espace des chemins. Il s'agit d'abord de construire en une passe le tableau des scores partiels optimaux aboutissant en un point donné du quadrillage présenté en figure 2.3. Chacun de ces scores partiels correspond à l'alignement d'une séquence préfixe de s_1 avec une séquence préfixe de s_2. Le score optimal récolté en bas à droite du tableau correspond au meilleur alignement global possible. Il suffit alors d'avoir pris soin de retenir lors du parcours du quadrillage la direction d'où venait l'alignement partiel optimal pour reconstruire ainsi, en rebroussant chemin à partir du bord inférieur droit, l'alignement global optimum. C'est un exemple de *programmation dynamique*, c'est-à-dire un algorithme qui construit progressivement la solution à un problème en considérant les solutions optimales à des sous-problèmes.

La traduction mathématique de ce qui précède peut s'énoncer ainsi :
- soit la séquence s_1 de taille n à aligner avec la séquence s_2 de taille m,
- soit la matrice de score S,

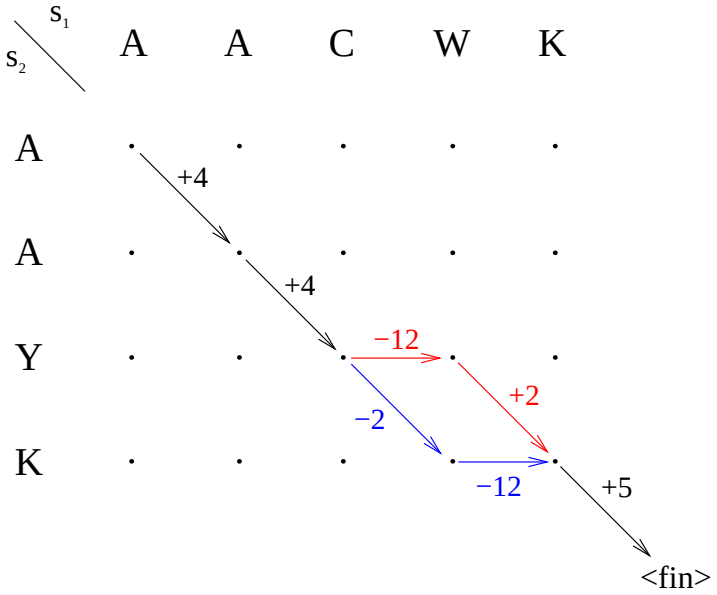


Figure 2.4. *Calcul des scores. Le chemin noir/bleu score* −1 *tandis que le chemin noir/rouge score* +3*. Cette différence s'explique par le fait qu'il est bien plus courant de rencontrer une substitution entre le tryptophane (W) et la tyrosine (Y) (score* +2*) qu'entre cette dernière et la cystéine (C) (score* −2*).*

- soit une matrice M à coefficients $M_{i,j}$ entiers, avec $i \in [1,n]$ et $j \in [1,m]$,
- on utilisera pour initialiser le remplissage de M des éléments fictifs tous nuls $M_{0,j}$ et $M_{i,0}$,
- on construit progressivement M en partant du coin supérieur gauche $M_{1,1}$ pour aboutir au coin inférieur droit $M_{n,m}$, en effectuant de manière itérative des recherches de maximum :

$$M_{i,j} = \max \begin{cases} M_{i-1,j-1} + S_{s_1[i],s_2[j]} \\ M_{i-1,j} + S_{s_1[i],\text{gap}} \\ M_{i,j-1} + S_{\text{gap},s_2[j]} \end{cases} \tag{2.3}$$

- parallèlement, lorsqu'on calcule $M_{i,j}$ on retient simultanément la direction d'où vient le chemin de score partiel maximum, $\text{dir}(i,j) \in \{\text{up}, \text{left}, \text{upleft}\}$.
- enfin, lorsqu'on est arrivé au bout et que $M_{n,m}$ est renseigné (et par construction il inclut déjà l'éventualité de longs gaps finaux depuis les éléments de type $M_{i<n,m}$ ou $M_{n,j<m}$), il suffit de suivre à l'envers le chemin $\text{dir}(i,j)$ à partir de la case de coordonnées (n,m) pour établir l'alignement global de score maximal égal à $M_{n,m}$.

Le caractère optimal du chemin obtenu provient directement de la propriété de décomposition décrite par l'équation (2.3) : le chemin optimal jusqu'à un certain point se

compose du chemin optimal jusqu'à un point antérieur et d'un pas incrémental entre ce point et le point courant.

2.2.2 Algorithme de Smith et Waterman

L'algorithme de Smith-Waterman [Smith et Waterman, 1981] propose une légère modification de l'algorithme de Needleman-Wunsch visant à permettre l'alignement *local* entre deux séquences. Alors que la problématique de l'alignement global consistait à définir un alignement qui consomme coûte que coûte *tous* les caractères des deux séquences, l'alignement local s'attache à rechercher des *fragments* de séquences prises dans l'une et dans l'autre et s'alignant bien entre eux. Bien souvent, deux séquences protéiques apparentées l'une à l'autre ne vont pas nécessairement présenter un alignement global satisfaisant, par exemple parce que l'une aura subi bon nombre d'insertions par rapport à l'autre, ou simplement parce que, autour de sous-unités fonctionnelles appelées *domaines* et relativement bien conservées, le reste des séquences n'est absolument pas conservé et donc difficilement alignable. Dans un tel contexte, l'algorithme de Smith et Waterman permet de découvrir sans information supplémentaire des couples de fragments alignables l'un à l'autre.

La modification proposée par les auteurs au programme de Needleman et Wunsch est très simple : elle consiste à « reprendre à zéro » l'alignement en cours dans la matrice M dès lors qu'on a atteint un score négatif. Par construction, la matrice M dans l'algorithme de Smith et Waterman ne contient donc que des valeurs positives ou nulles, puisqu'on a rajouté un terme (0) à l'opération « max » de l'équation (2.3).

Cette astuce demande cependant quelques ajustements supplémentaires :

- un alignement peut démarrer à n'importe quel endroit dans la table mettant en regard l'une et l'autre des deux séquences, mais également se terminer n'importe où. On ne déterminera donc pas l'alignement à partir du seul coin inférieur droit du tableau, mais on parcourra la totalité du tableau pour y repérer les valeurs maximales,
- le score attendu pour un alignement aléatoire doit nécessairement être négatif, sans quoi les alignements longs recevraient automatiquement un score plus grand que les alignements courts, même étendus aléatoirement,
- on doit au moins avoir un score d'alignement $s(a,b)$ positif, sans quoi le tableau serait irrémédiablement rempli de zéros.

Ces deux algorithmes ont fait l'objet de maintes optimisations depuis leur publication. Avec des sophistications plus ou moins importantes (par exemple pour cibler *a priori* les sous-séquences dans lesquelles on est susceptible de trouver un alignement local), c'est toujours l'algorithme de Smith et Waterman qui est employé pour découvrir des similarités entre séquences, par exemple dans le célèbre logiciel BLAST et ses dérivés.

2.2.3 BLAST et FASTA, deux outils très populaires

Le logiciel BLAST [Altschul *et al.*, 1990] propose à l'utilisateur de trouver à partir d'une séquence requête et d'une base de données, des séquences issues de celle-ci et présentant de fortes similarités locales avec celle-là. Les séquences testées (donc issues de la base de données) sont appelées *cibles* ou encore, lorsqu'elles émergent de la recherche avec des scores de similarité significativement élevés, *hits*. BLAST procède en commençant par rechercher ce qu'il appelle des « graines » (*seeds*), c'est-à-dire de courtes sous-séquences faites de correspondances exactes ou quasi exactes entre séquence requête et séquence cible. Originellement, la taille de ces graines était de trois acides aminés pour ce qui concerne les séquences protéiques. Pour chaque séquence cible ayant donné au moins une graine, le logiciel étend ensuite progressivement à gauche et à droite, simultanément dans la séquence cible et dans la séquence requête, les graines trouvées jusqu'à obtenir un fragment de longueur maximale (i.e. dont toute extension supplémentaire ferait baisser le score de similarité). Le score d'un hit est ensuite calculé à partir de la somme des scores des fragments correspondant à la paire (requête, cible).

La très grande popularité du logiciel BLAST doit beaucoup à sa simplicité d'utilisation et à sa rapidité : la recherche de correspondances exactes de faible longueur est computationnellement bien plus efficace que l'alignement global systématique de la requête avec toutes les cibles, prohibitif lorsqu'il s'agit de repérer des similarités dans de grandes bases de données.

Plus le score d'alignement est élevé entre la requête et la cible, et plus on aura confiance en le fait que la séquence cible appartienne à la famille d'homologie dans laquelle se trouve la séquence requête. Cette approche permet dans de nombreux cas à elle seule de construire une famille de séquences homologues à partir d'une seule séquence requête. Après BLAST, elle a été reprise par de nombreux logiciels dérivés, parmi lesquels on peut citer Wu-Blast [States et Gish, 1994], PSI-Blast [Altschul *et al.*, 1997] ou encore Blat [Kent, 2002].

Le logiciel FASTA [Pearson et Lipman, 1988] procède de façon assez similaire à BLAST, en ajoutant cependant une étape finale d'alignement global de la séquence requête avec chacun des hits, en prenant en compte des coûts pour l'ouverture et l'extension de gaps.

2.3 Alignements multiples : plus de deux séquences

Nous avons présenté dans ce qui précède les techniques existantes pour aligner deux séquences. Le problème appelé « alignement multiple de séquences » consiste, lui, à

aligner entre elles *plus de deux séquences*, en conservant toujours comme objectif de présenter sur une même colonne les différents caractères dérivant tous d'un même caractère ancestral. Nous nous bornons ici au cadre strict de l'alignement de *séquences* et non de *génomes*, c'est-à-dire que l'on suppose qu'il n'y a pas eu de réarrangements (par exemple des inversions de domaines) au cours de l'histoire évolutive de la famille de séquences à aligner : l'ordre des domaines composant les séquences est préservé, même si des délétions ont eu lieu.

2.3.1 Méthodes progressives

La plupart des algorithmes d'alignement multiple fonctionnent de manière *progressive*, c'est-à-dire qu'ils commencent par construire un arbre à partir des séquences non alignées, puis qu'ils construisent progressivement un alignement multiple en alignant à chaque pas une séquence supplémentaire (ou un sous-ensemble déjà aligné) contre l'alignement en cours d'élaboration.

La première étape, celle consistant à concevoir un arbre à partir des séquences, se fait en général de façon rapide et sous-optimale. Le schéma général retenu consiste d'abord à calculer des distances entre séquences en alignant ces dernières deux-à-deux. Puis on construit un arbre en agrégeant progressivement les séquences : d'abord les deux les plus proches, puis en incluant de façon graduelle les plus distantes. Deux méthodes sont utilisées pour construire ce genre d'arbre, les méthodes UPGMA (pour *Unweighted Pair Group Method using arithmetic Averages*, [Sokal et Michener, 1958]) et NJ (*Neighbour Joining*, [Saitou et Nei, 1987]).

Une fois cet arbre construit, il devient un *arbre guide* pour la procédure d'alignement multiple, qui commence par aligner entre elles les feuilles les plus proches puis suit l'arbre guide pour aligner progressivement les séquences, en utilisant des routines d'alignement « deux-à-deux » entre une séquence et un modèle (l'alignement en cours de construction) ou bien entre deux modèles (deux alignements multiples partiels de sous-ensembles de l'ensemble des séquences à aligner). Le logiciel d'alignement multiple progressif le plus connu est sans aucun doute CLUSTALW [Thompson *et al.*, 1994a]. Des successeurs sont apparus par la suite, raffinant l'approche développée dans CLUSTALW. MAFFT [Katoh *et al.*, 2002] et MUSCLE [Edgar, 2004] sont les plus connus. On peut également citer DIALIGN-TX [Subramanian *et al.*, 2008]. Aujourd'hui, la plupart des algorithmes d'alignement multipes, dont ceux que nous venons de citer, proposent cependant une méthode itérative de raffinement (cf ci-dessous) qui vient dans un second temps améliorer l'alignement initial obtenu par une méthode progressive.

DIALIGN-TX

Publié en 2008 [Subramanian *et al.*, 2008], DIALIGN-TX propose une méthode progressive de constructions d'alignements multiples à partir de fragments locaux de forte similarité. L'étape de construction d'un alignement multiple à partir de fragments locaux impliquant deux séquences se fait en maximisant une fonction objectif directement liée à la p-valeur de ces fragments locaux. L'idée de p-valeur est centrale en statistique (notamment pour évaluer la significativité de *hits* obtenus par Blast), la p-valeur d'un fragment étant ici la probabilité que l'on puisse trouver dans deux séquences aléatoires un fragment aussi long donnant un score de similarité au moins aussi élevé.

2.3.2 Méthodes itératives

Un inconvénient majeur des méthodes d'alignement progressif est que celles-ci sont fortement dépendantes de l'arbre guide : puisqu'on ne revient jamais en arrière (les colonnes composant l'alignement des séquences au sein d'un sous-groupe ne sont jamais remises en cause par l'alignement ultérieur de ce sous-groupe avec les séquences restantes), deux choix initiaux différents de l'arbre guide mènent souvent à des alignements différents. Les méthodes itératives ont été mises au point pour pallier ce désavantage [Barton et Sternberg, 1987; Berger et Munson, 1991; Gotoh, 1993].

Les méthodes itératives visent à déterminer le « meilleur » alignement à l'aune d'une *fonction objectif* bien définie. Bien souvent, cette fonction objectif est la somme des scores entre paires alignées (SP-score, pour *Sum of Pairs*). Un alignement de longueur L comprenant m séquences induisant $L\sum_{k=1}^{m-1} k = \frac{Lm(m-1)}{2}$ paires de caractères alignés (« caractères » est ici à interpréter au sens large : il s'agit d'acides aminés ou de gaps), la combinaison d'une matrice de scores de similarité entre acides aminés et d'une stratégie de pénalisation des gaps permet d'attribuer un SP-score à tout alignement multiple de séquences. C'est en général ce score SP que les méthodes itératives se proposent d'optimiser.

Une fois déterminée la fonction objectif, le propre des méthodes itératives est de revenir à plusieurs reprises sur l'alignement multiple, en partitionnant l'alignement en deux sous-groupes de séquences, puis en réalignant les sous-groupes l'un à l'autre après en avoir supprimé les colonnes composées exclusivement de gaps. L'itération suivante partira alors de l'alignement obtenu par le réalignement précédent si et seulement si cet alignement possède un meilleur SP-score que celui à partir duquel il a été calculé.

Pour économiser du temps de calcul et choisir des partitions susceptibles d'apporter le plus d'améliorations, il a été suggéré [Hirosawa *et al.*, 1995] que les partitions en deux sous-groupes correspondent à une bipartition selon une branche d'un arbre supportant

les espèces à aligner. C'est notamment l'approche retenue par les versions itératives du logiciel MAFFT.

MAFFT

Publié en 2002, MAFFT [Katoh *et al.*, 2002] apporte une innovation originale : le processus d'alignement démarre avec la recherche de fortes similarités entre séquences, laquelle s'appuie sur un changement de représentation des objets alignés. En effet, les acides aminés composant chacune des séquences à aligner sont d'abord résumés en autant de vecteurs comportant seulement deux coordonnées : l'une représentant le volume de l'acide aminé, l'autre sa polarité. Cette projection des acides aminés sur un espace bidimensionnel est due à Grantham [Grantham, 1974]. Katoh et coauteurs calculent alors la corrélation entre deux séquences décalées l'une par rapport à l'autre de k positions comme un produit de transformées de Fourier. Les valeurs de k donnant une corrélation importante signalent un alignement potentiel (au moins local) d'une séquence par rapport à l'autre. Une analyse à fenêtre glissante courant le long de l'alignement induit par le décalage k permet ensuite de déterminer le ou les blocs de forte similarité locale entre séquences.

Le reste du processus d'alignement est plus classique : programmation dynamique pour obtenir le chemin optimal fait de segments (c'est-à-dire de blocs de forte similarité) cohérents, extension de la méthode basée sur le transformées de Fourier à des alignements entre groupes de séquences déjà alignées, stratégie globale d'alignement ou bien progressive (versions FFT-NS-1 et FFT-NS-2), ou bien itérative (FFT-NS-i).

Sous l'influence de T-Coffee, les auteurs de MAFFT ont publié en 2005 des améliorations introduites par la version 5 de leur logiciel : les stratégies d'alignement multiple F-INS-i, G-INS-i et H-INS-i incluent dans la fonction objectif de l'information provenant des alignements deux-à-deux réalisés entre les séquences à aligner. La fonction objectif n'est plus le SP-score, mais le WSP-score (pour *Weighted Sum of Pairs*). La pondération réalisée dans MAFFT est complexe, elle se résume en disant qu'une paire de caractères (a,b) apparaissant respectivement aux positions i et j des séquences respectives s et t verra son score d'alignement (dépendant uniquement de a et de b dans le schéma SP simple) pondéré simultanément par :

- la fréquence avec laquelle le i^{e} site de la séquence s se trouve impliqué dans un segment sans gap dans les différents alignements entre paires de séquences,
- la fréquence avec laquelle le j^{e} site de la séquence t se trouve impliqué dans un segment sans gap dans les différents alignements entre paires de séquences,
- le score d'alignement pour le segment sans gap issu de l'alignement entre les séquences s et t et comprenant à la fois s_i et t_j,
- la *confiance* $w_{a,b}$ placée dans l'alignement entre les séquences a et b, en général

plus ou moins inversement proportionnelle à la distance séparant les deux taxa. Le sujet est débattu, et MAFFT utilise une fonction de pondération w issue des travaux de Gotoh [Gotoh, 1995].

Enfin, de nouvelles améliorations de MAFFT ont été publiées en 2008 [Katoh et Toh, 2008], essentiellement dédiées à l'efficacité de l'alignement sur un très grand nombre de séquences ou à l'alignement d'ARN non codants. Ces améliorations sortent du cadre de notre exposé.

2.3.3 Méthodes basées sur la cohérence

Une façon très différente de procéder a été introduite par Cédric Notredame avec T-COFFEE [Notredame *et al.*, 2000], puis poursuivie dans les logiciels PROBCONS [Do *et al.*, 2005] et MAFFT dans ses versions L-INS-i, G-INS-i et E-INS-i [Katoh *et al.*, 2005; Katoh et Toh, 2008]. Il s'agit pour ces logiciels de défendre le point de vue suivant : le meilleur alignement est celui qui respecte au mieux les contraintes provenant des alignements deux-à-deux de toutes les séquences impliquéees. Dans le meilleur des cas, les alignements en question sont tous compatibles : les résidus alignés entre eux forment des cliques (une arête entre les résidus x_i et y_j signifiant que la position i de la séquence x est alignée à la position j de la séquence y dans l'alignement de x avec y). Mais bien souvent, dans les alignements non triviaux tout au moins, ce n'est pas toujours le cas. on aura par exemple les liens $x_i \leftrightarrow y_j$ et $y_j \leftrightarrow z_k$ sans que x_i ne soit aligné à z_k dans l'alignement de x avec z. C'est l'examen complet de l'ensemble de ces relations entre résidus qui permet aux méthodes dites « basées sur la cohérence » de déterminer un alignement optimal, c'est-à-dire qui soit le plus en accord possible avec les alignements deux-à-deux. La décision de privilégier telle relation entre résidus plutôt que telle autre se prend par exemple en attribuant à chaque relation un poids égal au pourcentage d'identité de séquence de la paire alignée dont ils proviennent.

T-Coffee

Publié en 2000 [Notredame *et al.*, 2000], T-Coffee est un logiciel reprenant l'idée des méthodes progressives mais intégrant tout au long du processus la prise en compte de contraintes issues des alignements deux-à-deux des séquences composant l'ensemble à aligner. Bien que plusieurs évolutions soient venues enrichir le programme depuis sa première version, nous résumons ici la méthode publiée en 2000.

T-Coffee commence par aligner toutes les paires de séquences, à la fois par une méthode d'alignement global (ClustalW [Thompson *et al.*, 1994a]) et par une méthode de recherche de similarités locales (programme Lalign de la suite FASTA [Huang et Miller, 1991]). Les auteurs désignent par le nom « librairie » un ensemble de contraintes pondé-

rées, chacune correspondant à l'alignement d'un caractère précis au sein d'une séquence donnée, contre un autre caractère présent dans une autre séquence. On obtient donc deux librairies, le poids de chacune des contraintes étant donné par le pourcentage d'identité entre les deux séquences en jeu, calculé sur les segments sans gap de l'alignement *pairwise* en question. On combine les deux librairies (l'une issue des alignements deux-à-deux globaux et l'autre issue des alignements deux-à-deux locaux) en sommant les poids associés à une contrainte particulière lorsque cette contrainte se trouve à la fois dans l'une et dans l'autre des deux librairies.

Une fois la librairie construite, une idée originale de Cédric Notredame consiste à l'*étendre* en examinant pour chaque élément de la librairie (c'est-à-dire pour chaque contrainte, donc chaque paire (s_i, t_j) de caractères alignés) sa cohérence avec les autres contraintes de la librairie. Ceci se fait en examinant les alignements de *triplets* de séquences : les paires de caractères alignés entre les séquences s et t sont évalués à l'aune de tous les triplets formés en superposant les alignements de s avec une séquence $s' \neq t$ et de s' avec t. Si s_i est aligné avec s'_k et que s'_k est aligné avec t_j, alors la paire (s_i, t_j) voit son poids augmenter du minimum des pourcentages d'identité de séquence entre s et s' et entre s' et t. Ainsi « étendue », la librairie des contraintes est telle que chacune des contraintes possède un poids qui résume une partie de l'information issue de *toutes* les séquences de l'ensemble, et pas seulement des 2 séquences dont est issue la paire en jeu.

L'alignement se fait ensuite de manière progressive classique en suivant un arbre guide, mais le point essentiel est que la matrice de similarité utilisée n'est pas une matrice générique du type BLOSUM, mais est la matrice des scores que sont les poids des paires alignées issues de la librairie étendue. Ainsi, le score d'alignement entre, par exemple, un tryptophane et une tyrosine, n'est pas donné de manière absolue. On évalue séparément le score d'alignement entre le tryptophane situé en position i dans la séquence s et la tyrosine en position j dans la séquence t, etc.

L'une des caractéristiques de T-Coffee qui en rendent l'usage particulièrement appréciable est sa grande flexibilité. T-Coffee fournit une multitude d'options à l'utilisateur, et accepte en particulier qu'on lui fournisse en entrée une ou des librairies partielles, par exemple pré-calculées à partir d'alignements structuraux deux-à-deux. Cette approche peut se révéler la plus fructueuse lorsqu'il s'agit de construire des alignements multiples d'homologues très distants, et nous l'utiliserons ici dans l'un de nos bancs de test.

ProbCons

Les auteurs se servent de manière centrale d'un pair-HMM [Durbin *et al.*, 1998] pour calculer les probabilités des différents alignements *pairwise* impliquant deux séquences s et t. À partir de ces probabilités, ils calculent la *probabilité postérieure* que s_i soit ali-

gné avec t_j. En faisant l'hypothèse que la distribution des probabilités des alignements *pairwise* donnée par le pair-HMM est une bonne approximation de la probabilité pour un alignement donné de correspondre à l'alignement biologiquement « juste », les auteurs calculent pour chaque alignement *pairwise* une valeur appelée « précision attendue » censée être une mesure probabiliste de la corrélation entre l'alignement en question et l'alignement « correct ». De tous les alignements *pairwise* entre les séquences s et t, celui qui maximise la précision attendue définit la mesure de similarité entre s et t dont les auteurs se servent pour construire l'arbre guide de l'alignement progressif.

Un peu à la manière de T-Coffee, Do et Batzoglou réestiment pour toutes les paires de séquences s et t et pour toutes les positions i de s et toutes les potisions j de t, la probabilité postérieure d'avoir s_i aligné avec t_j en examinant toutes les séquences tierces z, selon :

$$Pr(s_i \sim t_j \in a^* | s,t) \leftarrow \frac{1}{|S|} \sum_{z \in S} \sum_{z_k} \Pr(s_i \sim z_k \in a^* | z,k) \Pr(z_k \sim t_j \in a^* | z,t)$$

L'alignement progressif utilise ensuite comme score d'alignement entre résidus les expressions $Pr(s_i \sim t_j \in a^* | s,t)$, puis une procédure itérative (voir ci-avant) est utilisée pour raffiner l'alignement.

2.3.4 Alignements respectant la phylogénie

Les méthodes progressives ne se servent d'un arbre reliant les séquences entre elles qu'afin d'ordonnancer les tâches d'alignement partiel. Elles effectuent ensuite celles-ci en commençant par les séquences les plus proches pour finir avec les plus distantes. Rien n'est fait de façon rigoureuse pour que les événements phylogénétiques (insertions, délétions et substitutions) induisant l'alignement construit soient à la fois clairement localisables sur l'arbre guide et relativement peu nombreux. Le logiciel PRANK [Löytynoja et Goldman, 2005, 2008] développé par Ari Löytynoja et Nick Goldman a pour ambition d'être un logiciel d'alignement multiple étroitement guidé par la phylogénie : tout en alignant progressivement les séquences le long de son arbre guide, PRANK marque les positions correspondant à une insertion rencontrée plus bas dans l'arbre, pour les *interdire* pour toute la suite de l'alignement progressif : on n'y alignera plus de caractères. Ainsi, deux événements d'insertion situés à la même position dans l'alignement multiple mais *phylogénétiquement distincts* (c'est-à-dire intervenant sur deux branches distinctes) correspondront donc à deux colonnes distinctes dans l'alignement final, là où les méthodes traditionnelles d'alignement multiple ont justement plutôt tendance à agglutiner sur un nombre restreint de colonnes successives les résidus présents dans les zones de faible résolution (i.e. avec de nombreux gaps), sans prendre garde au fait que les résidus en

question soient phylogénétiquement issus du même résidu ancestral ou non [2].

Nous avons survolé dans ce chapitre les différentes techniques menant à l'alignement d'un ensemble de séquences entre elles. Cet alignement est en soi un objet d'étude : en faisant l'hypothèse que tout alignement de séquences homologues est une représentation partielle d'une famille de séquences dérivant de la même séquence ancestrale et partageant donc des caractéristiques communes, on est amené à décrire cette famille par l'intermédiaire de modèles statistiques, à partir des observations que constituent ces séquences « d'apprentissage ». L'objectif est donc d'apprendre un modèle stochastique à partir de quelques représentants alignés entre eux. C'est ce type de modèles que nous décrivons dans le chapitre suivant.

2. Ce comportement observé tient notamment à la gestion des pénalités d'ouverture de gap, que les logiciels classiques d'alignement multiple *minorent* aux positions où une insertion a déjà été réalisée dans l'alignement en cours de construction.

CHAPITRE

3 Des modèles pour décrire un alignement

Comme on l'a vu dans le chapitre précédent, un ensemble de séquences homologues peut être décrit par un alignement, c'est-à-dire que (s'il s'agit de séquences protéiques) les acides aminés composant les différentes séquences peuvent être alignés colonne par colonne, avec éventuellement l'insertion de caractères dits « gaps » pour combler les trous. À l'issue de la procédure d'alignement, on s'attend à ce que les caractères (nucléotides ou acides aminés) présents sur une même colonne soient tous issus de la même position ancestrale par le biais de processus évolutifs et/ou partagent des caractéristiques physico-chimiques semblables et remplissent donc le même rôle structurel et/ou fonctionnel dans leurs séquences respectives. Dans la suite, on parlera indifféremment de « sites » ou de « colonnes ». Que l'on souhaite rechercher dans des bases de données *d'autres* séquences apparentées ou que l'on cherche à caractériser les domaines fonctionnels communs aux séquences en question, il est utile de donner une modélisation nécessairement probabiliste (car l'ensemble des séquences observées n'est qu'un sous-ensemble de la réalité observable) des séquences alignées. C'est à de tels modèles que nous nous intéressons dans ce chapitre.

Sommaire

Nous présentons dans ce qui suit l'essentiel des modèles conçus pour rendre compte d'un alignement, en faisant la part belle aux modèles de Markov cachés (HMM) puisque ce sont ceux dont nous avons retenu la structure pour développer les modèles que nous exposons dans cette thèse. En fin de chapitre, nous mettons en exergue deux sections qui ont trait à deux aspects importants de la construction de modèles : la pondération relative des séquences d'apprentissage et la sélection des sites d'intérêt au sein d'un alignement. Ces deux points sont traités spécifiquement car ils nous intéressent tout particulièrement. En effet, le travail que nous présentons dans cette thèse résoud de manière élégante, par une approche différente, les deux problématiques sous-jacentes :

- la pondération des séquences d'apprentissage, ou comment prendre en compte les liens existant entre celles-ci,
- la sélection d'un sous-ensemble de colonnes au sein d'un alignement, c'est-à-dire la désignation des sites que l'on juge utile de modéliser pour caractériser la famille de séquences en jeu.

Un alignement de séquences décrit fondamentalement de l'*observé* : on y trouve des séquences connues (par exemple des hémoglobines de vertébrés), mises ensemble parce que l'on connaît les liens de similarité (structurelle ou fonctionnelle) qui les unissent, ou bien parce que ces similarités nous auront été fortement suggérées par un collègue chercheur ou des outils probabilistes (par exemple BLAST, cf. le chapitre précédent). Mais lorsqu'il s'agira de déterminer à partir de ces observations la structure canonique permettant de décrire la *totalité* de cette famille de séquences (observées ou non) tout en excluant les séquences qui n'en font définitivement pas partie, on voudra se donner les moyens de repérer dans une base de données *d'autres séquences* qui auraient toutes les raisons de faire partie de la famille en question mais qui n'y figuraient pas au départ. C'est là tout le problème de l'inférence d'un modèle à partir d'une connaissance partielle de la réalité : il faut à la fois apprendre suffisamment des données observées (pour tenter d'en extraire les caractéristiques propres), et éviter l'écueil du sur-apprentissage (consistant à particulariser le modèle à tel point qu'il ne soit plus capable de représenter autre chose que les données d'apprentissage elles-mêmes).

Depuis le milieu des années 1980, plusieurs modèles ont été proposés pour décrire de façon statistique le contenu d'un alignement. Tous ces modèles intègrent évidemment une part d'incertitude stochastique : ce sont des modèles probabilistes. Avec ceux-ci, on a accès de façon directe à une première mesure de similarité : Pr(séquence|modèle), pour toute séquence candidate à l'homologie. En première approximation, si une telle probabilité est élevée, alors le modèle en question aura de grandes chances de générer la séquence donnée, et on aura tendance à accepter l'hypothèse selon laquelle la séquence

testée appartient à la famille modélisée.

3.1 Les précurseurs : tables de scores position-spécifiques

Gribskov, MacLachlan et Eisenberg ont publié en 1987 une méthodologie novatrice [Gribskov *et al.*, 1987] pour construire des modèles probabilistes à partir d'un alignement de séquences protéiques, se servant ensuite de tels modèles afin d'effectuer des recherches d'homologues dans une base de données. Les auteurs partent d'un ensemble de séquences alignées (incluant donc possiblement des gaps). Pour chaque colonne p de l'alignement d'entrée, ils construisent une série de 20 scores $M(p,\alpha)$ (α représentant l'un des 20 acides aminés), en utilisant la formule

$$M(p,\alpha) = \sum_{\beta=1}^{20} W(p,\beta)\, S(\alpha,\beta) \tag{3.1}$$

dans laquelle β représente un acide aminé, $W(p,\beta)$ la proportion d'apparition (ou poids relatif) de cet acide aminé dans la colonne p, et $S(\alpha,\beta)$ est le terme d'une matrice de scores de substitution correspondant à la substitution de α par β. Cette formule combine donc une observation réalisée sur l'alignement d'apprentissage, $W(p,\beta)$, avec une connaissance *a priori* des scores d'alignement entre acides aminés, $S(\alpha,\beta)$. Dans leur article, Gribskov et coauteurs utilisent pour S l'une des toutes premières matrices de scores d'alignement entre acides aminés, la matrice MDM78 due à Margaret Dayhoff [Dayhoff *et al.*, 1979], tout en faisant remarquer que toute autre matrice de score pourrait être utilisée dans le même schéma général. Si $\overrightarrow{\text{obs}_p}$ est le vecteur des proportions observées à la position p et M_p le vecteur des 20 scores de « match » calculés pour cette même position, la méthode de Gribskov et al. s'exprime matriciellement de la façon suivante : $M_p = S\overrightarrow{\text{obs}_p}$.

Un vingt-et-unième score pen_p est calculé pour chacune des positions : c'est la pénalité associée à l'ouverture d'un gap ou d'une délétion à cet endroit. Alors que l'utilisateur fournit lui-même une pénalité o_{gap} d'ouverture d'un gap et une pénalité e_{gap} d'extension d'un gap d'un acide aminé supplémentaire, la pénalité retenue pour un gap de longueur L sera égale à $(\text{pen}_p(o_{\text{gap}} + L\, e_{\text{gap}}))$. Les pénalités pen_p sont dérivées de l'alignement d'apprentissage par une méthode que nous n'exposons pas ici, mais qui tend à diminuer la pénalité dans le voisinage immédiat des positions dans lesquelles on recontre un ou des gaps dans les séquences de l'alignement d'apprentissage.

Une fois le modèle établi sous la forme d'une matrice de scores de taille $M \times 21$ (si M est la taille de l'alignement fourni en entrée), la méthodologie de recherche de séquences homologues dans les bases de données consiste à aligner (par un algorithme

de programmation dynamique) les séquences de la base contre le modèle, pour ensuite retenir les séquences donnant les meilleurs scores. Le processus d'alignement consiste à associer chacun des acides aminés de la séquence cible à une position p du profil (c'est le nom donné par les auteurs à leur matrice M) ou à un gap, qui déclenchera une pénalité fonction de sa position et de sa longueur. Insertions et délétions par rapport au modèle sont comptées de la même façon, ce qui fait que les positions orphelines du profil (c'est-à-dire n'ayant pas été associées à un acide aminé de la séquence cible) engendreront des pénalités calculées de la même façon. On entend alors par « score » de la séquence cible la somme des scores $M(p,a)$ déclenchés par son alignement aux positions non orphelines, diminuée du total des pénalités d'insertions et de délétions.

Le modèle de « profil » ou « position-specific scoring table » développé par Gribskov et coauteurs se résume graphiquement à ce qui est présenté en figure 3.1 : une chaîne d'états « match » donnant des scores calculés selon (3.1), dont on peut sortir à tout moment pour une insertion ou une délétion par rapport au consensus. Il est à noter que ce consensus, directement représenté par la suite des états match avec leur acide aminé de plus fort score, est de même longueur que l'alignement fourni en entrée.

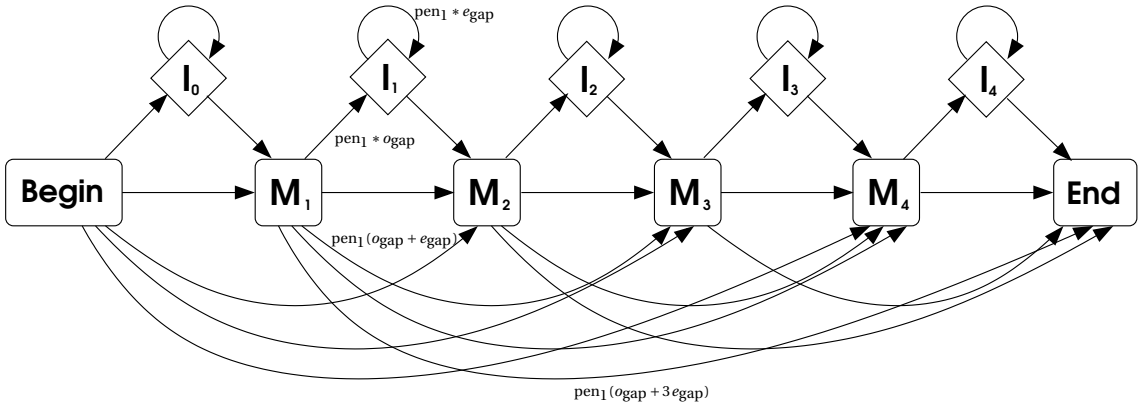


Figure 3.1. *Profil selon Gribskov et al. Il y a une correspondance bijective entre les états M_p du modèle et les positions p de l'alignement d'apprentissage. On a explicité uniquement quelques-unes des différentes pénalités pour ne pas surcharger le schéma.*

3.2 Modèles de Markov cachés (HMM)

Dans [Krogh *et al.*, 1994] et [Krogh, 1994], Anders Krogh et David Haussler, notamment, ont apporté une petite révolution dans le monde de la modélisation en bioinformatique. Ils ont eu l'idée d'importer dans notre champ scientifique des modèles génériques connus dans le domaine de l'apprentissage statistique depuis la fin des années 1960 et appliqués

notamment à des problématiques de reconnaissance de la parole (cf. [Rabiner, 1989]). Ces modèles sont connus sous le nom de « chaînes de Markov à états cachés », ou plus simplement « modèles de Markov cachés ». Nous emploierons ici régulièrement l'acronyme issu de l'anglais pour *Hidden Markov Model*, soit HMM.

Un HMM est simplement un automate fini probabiliste. Le passage par l'un de ses états produit un *caractère* selon une distribution de probabilités sur un alphabet fini. Cette distribution est définie séparément pour chaque état. On représente de tels automates par des graphes dont les sommets forment les états de l'automate et les arêtes indiquent les transitions possibles entre états. On donne un exemple de représentation d'un automate probabiliste à deux états en figure 3.2. On désigne par t_{11}, t_{12}, t_{21} et t_{22} les différentes probabilités de transition entre les états 1 et 2 (e.g. $t_{12} = 1-p$). $\Pi_i(\alpha)$ est la probabilité d'émettre le caractère α lorsqu'on se trouve dans l'état i. Ainsi envisagé, le HMM est un modèle qui génère des données selon un certain nombre de paramètres que sont la topologie du modèle, les probabilités de transitions et les distributions de probabilités correspondant aux émissions de caractères sur les états du modèle. On peut en effet construire une séquence suivant le modèle donnée en figure 3.2 en itérant la boucle ci-dessous à chaque instant d'un temps discret :

1. soit i l'état courant,
2. on tire aléatoirement un acide aminé selon la distribution de probabilités Π_i. Soit α cet acide aminé,
3. on ajoute α à la fin de la séquence en cours de construction,
4. on tire aléatoirement une des transitions sortantes de l'état i selon la distribution donnée par les $\{t_{i\cdot}\}$,
5. le nouvel état courant est l'état de destination de la transition tirée.

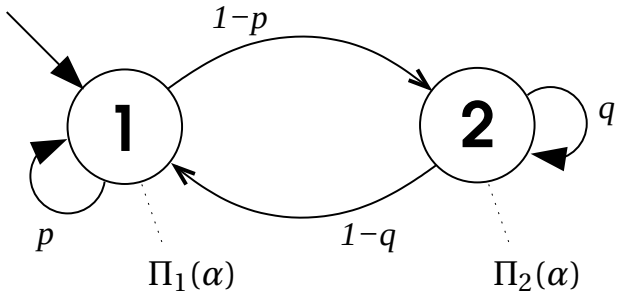


Figure 3.2. *Un automate fini à deux états. Les transitions entre états (arêtes du graphe) sont pondérées par des probabilités. Les deux états (sommets du graphe) génèrent des caractères α lorsqu'on les traverse, selon une distribution de probabilités Π propre à chaque état. L'état initial est indiqué par une arête entrante sans origine.*

Mais on peut également utiliser le HMM pour évaluer la pertinence d'une séquence de caractères donnée par rapport au modèle : si les deux sont statistiquement proches, alors c'est que la séquence de caractères en question aura pu être générée par le modèle, avec une forte probabilité. Précisons ce discours en introduisant la notion de *score* d'une séquence dans un HMM. Soit $X = X_1, X_2, \ldots, X_n$ une séquence de n caractères et $\mathscr{H}$ le HMM décrit en figure 3.2. Soit $h : [1,n] \longrightarrow \{1,2\}$ le *chemin* emprunté par la séquence X dans le HMM $\mathscr{H}$. Cela signifie que l'on fait l'hypothèse que pour tout i, le caractère X_i a été engendré par l'état $h(i)$. On peut alors exprimer la probabilité que la séquence ait été générée par le modèle, *sachant* ce chemin :

$$\Pr(X|\mathscr{H},h) = \prod_{i=1}^{n-1} \left(\Pi_{h(i)}(X_i)\, t_{h(i)h(i+1)}\right) \Pi_{h(n)}(X_n) \tag{3.2}$$

La détermination du score réalisé par la séquence X dans le HMM $\mathscr{H}$, $\Pr(X|\mathscr{H})$, sans information a priori sur le chemin suivi par la séquence dans le modèle se fait selon l'une ou l'autre de deux stratégies :

1. ou bien on détermine le chemin optimal h^*, c'est-à-dire $h^* = \arg\max_h \Pr(X|\mathscr{H},h)$, et alors le score retenu pour la séquence est justement $\Pr(X|\mathscr{H},h^*)$: c'est la stratégie dite de Viterbi,
2. ou bien on somme sur tous les chemins possibles, et l'on retient le score $\sum_h \Pr(X|\mathscr{H},h)$: c'est la stratégie basée sur l'algorithme dit « forward ».

Le terme « modèles de Markov cachés » fait référence au fait que l'on part de données observables (la séquence de caractères X) pour déterminer (avec une certaine incertitude) la séquence d'états $h(i)$ du HMM la plus susceptible d'avoir engendré X. C'est cette séquence d'états qui est « cachée » parce qu'inobservée, mais inférée du couple { modèle, séquence }. Cette stratégie d'inférence est utile par exemple si l'on souhaite découvrir dans une séquence biologique les zones codantes et non codantes : on peut établir un modèle de génération de caractères pour les zones codantes, un modèle de génération de caractères pour les zones non codantes, et essayer ainsi de déterminer automatiquement dans une séquence donnée en entrée du HMM, les deux types de zone. Un autre exemple classique est celui du « casino parfois malhonnête », consistant à modéliser un HMM du type de celui représenté en figure 3.2 dont l'un des états modélise des tirages aléatoires de chiffres entre 1 et 6 par un dé non pipé, et l'autre des tirages aléatoires par un dé pipé pour donner plus souvent le chiffre six. Sachant que le casino passe parfois d'un dé à l'autre, on peut se servir du HMM pour détecter à partir d'une longue série de tirages, quelles sont les phases pendant lesquelles le dé pipé a été utilisé (cf. [Durbin *et al.*, 1998], chap. 3).

3.3 HMM profils

Avec d'autres, Richard Durbin, David Haussler et Anders Krogh ont eu l'idée en 1993 de fabriquer des modèles dérivés des profils de Gribskov et coauteurs, en intégrant ceux-

ci dans un cadre probabiliste de type HMM. Partant de l'idée de [Gribskov *et al.*, 1987] et des modèles comme décrits en figure 3.1, Durbin, Haussler, Krogh et al. [Krogh *et al.*, 1994; Krogh, 1994] ont proposé des modèles qu'ils ont appelés « HMM profils ». Ces modèles sont des HMM à part entière, c'est-à-dire des modèles avec des transitions probabilistes (la somme des probabilités associées aux transitions quittant un état donné est toujours égale à 1, ces probabilités étant toutes positives) ainsi que des états décrits entièrement par une distribution de probabilités sur les caractères de l'alphabet cible (i.e. nucléotides ou acides aminés). Mais contrairement aux modèles HMM décrits dans la section précédente, les HMM profils ont tous la même structure canonique, assemblage linéaire de *nœuds* de trois états (Match, Délétion et Insertion). Chaque nœud est associé à un site de l'alignement, mais la taille du HMM profil (c'est-à-dire son nombre de nœuds) est *inférieure ou égale* à la taille de l'alignement d'apprentissage (alors qu'avec Gribskov elle lui était égale). Ainsi, la première phase de la construction du HMM profil consiste à sélectionner le sous-ensemble des positions de l'alignement d'entrée qui va faire l'objet d'une modélisation par des états Match. Les autres positions de l'alignement seront modélisées par les états d'Insertion, tandis que les états muets (i.e. n'engendrant pas de caractères) de Délétion permettront d'aligner contre le modèle une séquence n'exprimant pas toutes les positions Match. On représente un tel HMM profil à quatre nœuds en figure 3.3.

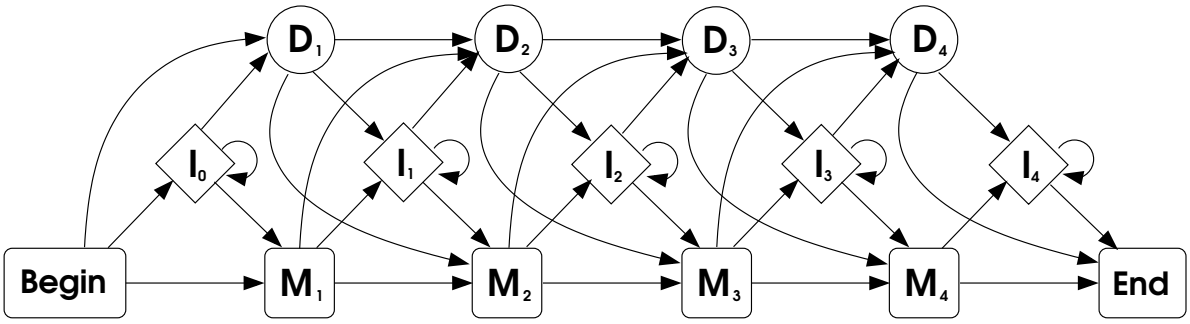


Figure 3.3. *Un HMM profil de longueur 4 structuré selon les idées de Krogh et Haussler. Chaque nœud introduit neuf transitions (MM, MI, MD, IM, II, ID, DM, DI, DD). Les états Match et Insertion sont associés chacun à une distribution de probabilités sur l'alphabet des caractères : ils génèrent un caractère lorsqu'on les traverse. Les états de Délétion, eux, sont silencieux : ils ne génèrent rien d'autre qu'un gap (« - ») qui n'a d'autre réalité que celle d'occuper une place dans un alignement tel qu'on les représente habituellement.*

3.3.1 Phases de conception d'un HMM profil

Quelle que soit l'implémentation logicielle utilisée, un HMM profil se conçoit en différentes étapes que nous décrivons brièvement ci-dessous :

1. analyse des séquences d'apprentissage pour déterminer les poids relatifs des séquences. Cette phase de pondération (cf. section 3.4) permet de diminuer l'importance qu'auront les séquences identiques ou quasi identiques pour la suite du processus d'apprentissage. En effet, on trouve souvent une certaine redondance dans les jeux de données, cette redondance étant due à divers facteurs (échantillonnage taxonomique, épissage alternatif [1], redondance des identifiants dans les bases, etc). En pondérant les séquences, on augmente l'importance des séquences singulières. Il faut noter qu'en cas d'erreur dans la construction de la famille d'apprentissage, ce mécanisme porte préjudice puisqu'il renforce l'importance des éventuels intrus.
2. détermination du sous-ensemble des sites qui vont être modélisés par des états Match. On peut par exemple adopter la règle simple consistant à retenir une colonne dès lors qu'elle comporte une proportion de gaps inférieure à un seuil donné.
3. détermination du poids total à attribuer à l'alignement de séquences d'apprentissage. Ce poids total représente la confiance donnée aux observations par rapport à un modèle *a priori* des séquences. Plus ce poids est élevé, plus on fera confiance aux observations dans la suite de l'apprentissage. Plus il est faible, et plus on privilégiera l'information a priori (par exemple la composition standard en acides aminés telle qu'observée dans les bases de données, la probabilité a priori d'ouvrir un gap, etc). La pondération totale de l'ensemble d'apprentissage peut se faire par exemple en fixant la quantité d'information moyenne que l'on veut tirer des sites alignés sur des états Match, par rapport à la composition de fond (par exemple 0,5 bit d'information par position).
4. apprentissage des paramètres du modèle :
   – distributions de probabilités d'émission de caractères sur les états Match,
   – distributions de probabilités d'émission de caractères sur les états Insertion,
   – probabilités de transition entre états.

   Cet apprentissage se fait soit par la combinaison directe des observations et des connaissances a priori, soit de manière itérative par un processus de recherche des paramètres optimaux appelé Expectation-Maximisation (EM).

Expectation-Maximisation

Publiée en 1977 par Dempster, Laird et Rubin [Dempster *et al.*, 1977], la méthode consiste à rechercher un ensemble θ^* de paramètres optimaux pour un modèle $\mathscr{M}$, à partir d'observations $\mathscr{D}$. Le modèle inclut des variables cachées, H. Dans le cas qui nous

1. Un transcrit est une succession d'introns et d'exons. Les introns sont des morceaux de séquence qui sont évacués lors de la traduction de l'ARNm en une séquence d'acides aminés, par un processus appelé *épissage*. Chez les eukaryotes, les signaux biologiques d'épissage sont variables selon le contexte et peuvent donc mener à des séquences de coupures et de ligatures différentes, produisant ainsi à partir d'un même gène des protéines différentes bien que proches. C'est ce phénomène que l'on appelle épissage *alternatif*.

intéresse ici, $\mathscr{D}$ correspond à un alignement de séquences, $\mathscr{M}$ est un HMM profil dont les paramètres θ sont toutes les probabilités d'émission et de transition correspondant aux différents états. Enfin, H est l'ensemble (discret) des *chemins cachés* possiblement empruntés par les séquences dans le HMM. Ils sont dits « cachés » car inconnus de l'utlisateur : si l'on fait l'hypothèse que le modèle représente fidèlement la réalité, les « vrais » chemins cachés sont les séquences d'états ayant effectivement généré les séquences observées.

L'objectif de l'algorithme d'Expectation-Maximisation est de donner au moins une approximation du jeu de paramètres optimal θ^* tel que la vraisemblance $\text{Lk}(\theta^*|\mathscr{M},\mathscr{D}) = Pr(\mathscr{D}|\mathscr{M},\theta^*)$ soit maximale. Il procède par une série d'itérations successives de raffinement d'un jeu de paramètres θ_t, chaque itération se composant de deux phases que nous décrivons ci-dessous.

La phase *Expectation* (E) consiste à calculer la vraisemblance attendue pour un jeu de paramètres θ en faisant certaines hypothèses liées à l'état courant du jeu de paramètres, θ_t. L'état des variables cachés étant inconnu, le calcul de la vraisemblance d'un jeu de paramètres induit une sommation :

$$\Pr(\mathscr{D}|\mathscr{M},\theta) = \sum_H \Pr(\mathscr{D},H|\mathscr{M},\theta) \tag{3.3}$$

Cette sommation sur les variables cachées H étant computationnellement difficile à établir, l'algorithme EM propose de calculer la vraisemblance attendue pour θ en utilisant un jeu de paramètres présupposé (c'est le rôle de θ_t) afin de déterminer la distribution de probabilité des variables cachées. Ainsi, on détermine une quantité $Q(\theta|\theta_t)$ correspondant à la log-vraisemblance de θ « sachant θ_t » :

$$\begin{aligned} Q(\theta|\theta_t) &= \sum_H \Pr(H|\mathscr{D},\mathscr{M},\theta_t)\log\Pr(\mathscr{D},H|\mathscr{M},\theta) &(3.4)\\ &= \mathbb{E}_{H|\mathscr{D},\theta_t}\left[\log\Pr(\mathscr{D},H|\mathscr{M},\theta)\right] &(3.5)\end{aligned}$$

La phase de *Maximisation* (M) consiste alors à déterminer la valeur θ_{t+1} qui sera utilisée pour l'itération suivante. Cette valeur est obtenue en maximisant $Q(\theta|\theta_t)$:

$$\theta_{t+1} = \arg\max_\theta (Q(\theta|\theta_t)) \tag{3.6}$$

Il faut noter qu'en général le problème correspondant à l'étape de maximisation est difficile à résoudre. La classe des algorithmes dits « Generalised Expectation Maximisation » se contente de déterminer à chaque itération un jeu de paramètres θ_{t+1} tel que $\text{Lk}(\theta_{t+1}|\mathscr{D},\mathscr{M}) > \text{Lk}(\theta_t|\mathscr{D},\mathscr{M})$. C'est à cette classe qu'appartient l'algorithme de Baum-Welch [Baum, 1972] qui est une implémentation de l'algorithme EM pour l'apprentissage

des paramètres d'un HMM profil.

Dans le cadre de ces modèles, les variables cachées correspondent comme on l'a dit plus haut aux chemins empruntés par les différentes séquences d'apprentissage dans le HMM profil. Les paramètres θ sont les probabilités d'émission et de transition décorant respectivempent les états et les arêtes du modèle. L'étape (E) consiste alors à calculer le nombre attendu d'utilisations de chacune des transitions ou des émissions du modèle. Cette étape utilise directement les données issues de l'algorithme *forward* qui permet de calculer l'alignement optimal d'une séquence au modèle ou encore les probabilités postérieures pour un caractère d'avoir été engendré par l'un ou l'autre des états du HMM.

L'étape de maximisation est ensuite relativement triviale puisqu'elle consiste à affecter à chaque transition ou chaque émission une probabilité proportionnelle à son utilisation attendue calculée lors de l'étape (E). Par exemple, si A_{kl} est le nombre d'utilisations attendues (en se basant sur le jeu de paramètres θ_t) pour la transition allant de l'état k vers l'état l, alors la probabilité correspondante a_{kl} prendra dans le jeu de paramètres θ_{t+1} la valeur $a_{kl} = A_{kl} / \sum_j A_{kj}$.

3.3.2 Score d'une séquence dans un HMM profil

Comme en ce qui concerne les HMM, le score brut d'une séquence X dans un HMM profil $\mathscr{M}$ est soit $\Pr(X|\mathscr{M}) = \sum_h \Pr(X|\mathscr{M}, h)$ (stratégie forward), soit $\Pr(X|\mathscr{M}, h^*)$ (stratégie de Viterbi). Pour un chemin h fixé, $\Pr(X|\mathscr{M}, h)$ s'obtient directement en effectuant le produit de toutes les probabilités d'émission et de toutes les probabilités de transition rencontrées sur le chemin h. Il faut remarquer qu'étant donné la séquence X et le HMM profil $\mathscr{M}$, déterminer un chemin dans le HMM revient exactement à *aligner* X contre $\mathscr{M}$, c'est-à-dire à associer chacun des nœuds du HMM à une position dans la séquence, et à dire s'il s'agit d'une association « Match » (auquel cas on associe le noeud du HMM à un caractère de la séquence) ou d'une association « Delete » (auquel cas on associe le nœud du HMM à un intervalle entre deux caractères consécutifs de X). Cette notion d'alignement d'une séquence à un HMM profil est explicitée graphiquement sur un exemple en figure 3.4. Les algorithmes permettant d'obtenir cet alignement (algorithme *forward* ou algorithme de Viterbi) appartiennent à la classe des algorithmes de programmation dynamique, et sont proches de l'algorithme de Needleman et Wunsch évoqué au chapitre précédent.

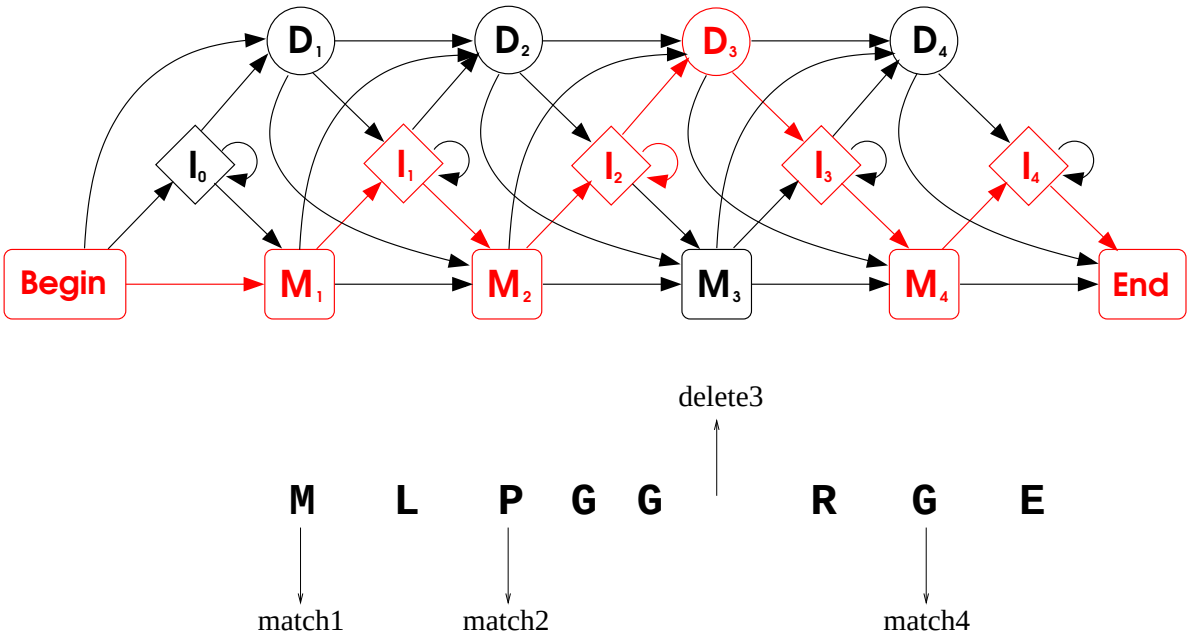


Figure 3.4. *Un HMM profil de longueur 4 et une séquence X pour laquelle on propose en dessous un alignement possible contre le HMM profil. L'association de certaines positions de la séquence à un état Match* ou bien *à un état Délétion, en n'ignorant aucun des nœuds du HMM, suffit à déterminer complètement le chemin suivi par la séquence dans le HMM (en rouge) : les insertions se déduisent du reste. Lorsqu'on voudra inclure la séquence X dans un alignement multiple de séquences contre ce HMM, un caractère de gap viendra s'insérer dans X à la place marquée « delete3 », sur la colonne où les autres séquences exprimeront un caractère alignable avec l'état* M_3.

3.3.3 SAM, première implémentation de HMM profil pour les séquences biologiques

SAM est un acronyme pour « Sequence Alignment and Modeling system ». Cette suite d'outils logiciels (http://compbio.soe.ucsc.edu/sam.html) a été développée dès 1993 à l'Université de Californie Santa Cruz (UCSC) par une équipe composée de Michael Brown, David Haussler, Anders Krogh, I. Saira Mian et Kimmen Sjölander. Le papier original est [Krogh *et al.*, 1994] et les principaux contributeurs et mainteneurs du code de SAM sont Richard Hughey, Kevin Karplus et Anders Krogh. Aujourd'hui SAM n'est plus réellement maintenu, ayant cédé la place à son rival HMMER en partie à cause du fait que les auteurs de SAM n'ont pas souhaité en publier le code source (K. Sjölander, communication personnelle). La dernière version en date en juillet 2011 est la version 3.5, datant de juillet 2005. Nous présentons ci-dessous en quelques points les caractéristiques essentielles de SAM.

Structure des HMM

SAM retient la structure linéaire proposée par Krogh et Haussler et présentée plus haut en figure 3.3 : un enchaînement de nœuds composés chacun de 3 états (Match, Délétion et Insertion) et de 9 arêtes (cf. fig 3.5). Des modules appelés « Free Insertion Modules » peuvent être insérés à différents endroits de la chaîne (notamment avant le premier nœud et après le dernier) pour permettre la reconnaissance de motifs locaux (modélisés par la sucession des nœuds du HMM) sans la pénalisation qu'engendrerait le passage par de multiples états d'Insertion.

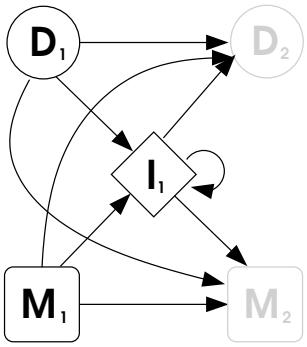


Figure 3.5. *Un nœud et ses 9 transitions pour les HMM de SAM*

Taille du modèle

La taille du HMM appris à partir d'un alignement de longueur N est inférieure ou égale à N. Elle est déterminée au fur et à mesure du processus d'apprentissage. Ce dernier étant principalement constitué par un algorithme de type *Expectation-Maximisation*, il consiste essentiellement à obtenir un jeu de paramètres (probabilités d'émission et de transition) θ_t à l'étape t, pour ensuite dénombrer le nombre d'utilisations des différentes transitions et émissions faites par les séquences alignées contre le modèle θ_t de manière optimale. Ces comptages donnent des fréquences relatives qui servent à déterminer θ_{t+1}, les paramètres du modèle à l'étape suivante. On met fin aux itérations lorsqu'un critère de stationarité est satisfait.
Ce mécanisme itératif joue aussi son rôle pour la détermination de la taille du HMM : un état d'insertion utilisé à l'étape t par plus de la moitié des séquences deviendra un état match à l'étape $t+1$, de même qu'un état match utilisé par moins de la moitié des séquences à l'étape t sera commué en un état d'insertion à l'étape $t+1$. C'est ainsi que l'on converge vers un certain nombre d'états match, utilisés par une majorité des séquences d'apprentissage.

Lissage des distributions

On appelle *lissage* des distributions la phase de l'apprentissage statistique consistant à éviter les problèmes de sur-apprentissage en mitigeant les observations réalisées par l'adjonction de probabilités *a priori*. Cette étape porte également le nom « régularisation ». SAM fait un usage important des travaux de Kimmen Sjölander en la matière [Sjölander *et al.*, 1996], en proposant plusieurs mélanges de Dirichlet (notamment développées par Kevin Karplus). Les mélanges de Dirichlet sont des compositions de plusieurs distributions de Dirichlet, chacune d'elles ayant pour objet de décrire des pseudocomptes à ajouter aux observations pour modérer le biais apporté par des observations faites en petit nombre. Supposons qu'on observe pour un état match M_i donné les comptes d'acides aminés émis par cet état match : soit $c_{i,a}$ le nombre de fois où l'on observe dans l'alignement d'apprentissage que l'état M_i du HMM produit le caractère a. L'apprentissage le plus simple indiquerait des probabilités d'émission proportionnelles aux observations :

$$\Pr(a|M_i) = \frac{c_{i,a}}{\sum_b c_{i,b}}$$

Mais alors, quid des probabilités $\Pr(a|M_i)$ lorsqu'aucune observation dans le jeu de données en entrée ne vient indiquer que l'état M_i a produit le caractère a ? On a inexorablement $c_{i,a} = 0 \Rightarrow Pr(a|M_i) = 0$. Sauf si l'on ajoute artificiellement aux $c_{i,a}$ ce qu'on appelle des *pseudocomptes*, sortes d'observations virtuelles destinées à éviter les probabilités nulles en sortie de la phase d'apprentissage.

La stratégie la plus simple pour produire des probabilités d'émission pour M_i en contournant le problème des probabilités nulles là où on a un compte nul pour un ou plusieurs caractère(s) consiste à adopter des pseudocomptes égaux à 1 pour tous les caractères : ce sont les pseudocomptes dits de Laplace. Dans le cas d'acides aminés (alphabets de longueur 20), cette approche s'écrit :

$$\Pr(a|M_i) = \frac{c_{i,a} + 1}{\left(\sum_b c_{i,b}\right) + 20}$$

Il faut noter que lorsque tous les comptes $c_{i,a}$ sont suffisamment grands devant 1, l'effet de la régularisation devient négligeable. Cet effet est valable quel que soit le lissage employé.

On peut raffiner ce *modus operandi* des pseudocomptes de Laplace en tenant compte de la connaissance que l'on a éventuellement des probabilités *a priori* $\Pr(a)$: si l'on sait que de manière générale un acide aminé est bien plus courant qu'un autre (voir plus loin les statistiques de la base de données UniProtKB en page 66), alors on peut établir des pseudocomptes différents α_a pour les différents caractères a. Au demeurant, ces pseudocomptes n'étant pas des observations réelles, ils n'ont nul besoin d'être entiers mais simplement tous positifs stricts. Si le caractère a est a priori plus courant que le caractère b,

alors on a $\alpha_a > \alpha_b$. La formule de régularisation avec ces probabilités a priori (dites « de Dirichlet ») s'écrit :

$$\Pr(a|M_i) = \frac{c_{i,a} + \alpha_a}{\sum_b \left(c_{i,b} + \alpha_b\right)}$$

Enfin, on en arrive aux *mélanges* de Dirichlet si l'on pousse la réflexion jusqu'à considérer que les sites de l'alignement ont des propriétés physico-chimiques bien déterminées, et qu'il est inutile de noyer un signal bien conservé sous des probabilités a priori qui vont ajouter trop de bruit. Par exemple, un site contenant exclusivement des isoleucines et des leucines sera un site avec un fort biais vers les acides aminés aliphatiques (essentiellement I, L, V et M). Si la nature d'un tel site est avérée et sous-tendue par un nombre raisonnable d'observations (e.g. 3 I, 4 L et deux gaps pour un site de taille 9), alors on peut faire confiance à cette caractéristique physico-chimique et ne pas ajouter aveuglément des probabilités a priori « bonnes pour tout », mais plutôt des probabilités a priori correspondant à un grand nombre d'observations toutes issues de sites aliphatiques dans une grande base de données. Ainsi, un mélange de Dirichlet décrit-il une collection de K jeux de n pseudocomptes, où n est la taille de l'alphabet considéré et où chacun des K jeux de pseudocomptes α^k est adapté plus ou moins exclusivement à un type de site. Le mélange de Dirichlet n'est pas complet sans que l'on indique les probabilités a priori de chacune des K classes d'appartenance de site, p_k. La règle de régularisation est alors légèrement plus complexe que précédemment, mais s'écrit encore assez directement :

$$\Pr(a|M_i) = \sum_{k=1}^{K} \Pr(k|\vec{c_i}) \frac{c_{i,a} + \alpha_a^k}{\left(\sum_b c_{i,b} + \alpha_b^k\right)}$$

Les probabilités $\Pr(k|\vec{c_i})$ que le site i appartienne à la classe k sachant les observations, s'obtiennent par la loi de Bayes et le calcul des $\Pr(\vec{c_i}|k)$ (cf. [Brown *et al.*, 1995]) :

- soit $c_{i,j}$ le nombre de caractères d'indice j dans l'alphabet des acides aminés que l'on peut observer sur la colonne $\vec{c_i}$,
- soit $c_i = \sum_{j=1}^{20} c_{i,j}$ le nombre total de caractères (non gaps) observés sur la colonne $\vec{c_i}$,
- si k est l'indice correspondant à l'une des composantes du mélange de Dirichlet,
- soit $\alpha_j^{(k)}$ la prior (ou pseudocompte) liée au caractère d'indice j dans la k^{e} composante du mélange,
- soit $\alpha^{(k)} = \sum_{j=1}^{20} \alpha_j^{(k)}$ le poids total (somme des pseudocomptes) correspondant à la k^{e} composante du mélange,

la loi de Bayes s'écrit alors :

$$\Pr(k|\vec{c_i}) = \frac{\Pr(\vec{c_i}|k)\Pr(k)}{\Pr(\vec{c_i})} = \frac{\Pr(\vec{c_i}|k)\Pr(k)}{\sum_{\kappa=1}^{K} \Pr(\vec{c_i}|\kappa)\Pr(\kappa)} \tag{3.7}$$

et enfin, les expressions de la forme $\Pr(\vec{c_i}|\kappa)$ s'écrivent à l'aide de la fonction Gamma :

$$\Pr(\vec{c_i}|\kappa) = \frac{\Gamma(c_i+1)\Gamma(\alpha^{(\kappa)})}{\Gamma(c_i+\alpha^{(\kappa)})} \prod_{j=1}^{20} \frac{\Gamma(c_{i,j}+\alpha_j^{(\kappa)})}{\Gamma(c_{i,j}+1)\Gamma(\alpha_j^{(\kappa)})} \tag{3.8}$$

3.3.4 HMMER 2.x

Le logiciel HMMER a été développé principalement par Sean R. Eddy (Howard Hughes Medical Institute, Janelia Farm, état de Virginie, US). Au départ Sean Eddy, ancien étudiant de Richard Durbin, envisageait ce projet comme une simple réimplémentation des idées originales de Durbin, Krogh et Mitchison, exposées dans le livre qu'ils ont écrit en commun [Durbin *et al.*, 1998]. La première version publique, HMMER 1.8, date d'avril 1995. L'écriture de HMMER 2 a démarré fin 1996 et reconsidérait beaucoup d'aspects du logiciel en repartant de zéro. La dernière version stable, 2.3.2, est sortie en octobre 2003 et constitue la version de référence lorsqu'on parle aujourd'hui de « HMMER 2 ». C'est de cette version que nous parlons ci-dessous.

Architecture Plan7

Les modèles conçus par le logiciel HMMER 2 le sont selon une architecture canonique appelée *Plan 7* car chaque nœud du HMM y introduit sept transitions (là où SAM présentait neuf transitions par nœud). La raison de ce passage de neuf à sept réside dans l'abandon pur et simple par HMMER 2 des transitions de l'état Insertion d'un nœud à l'état Délétion du nœud suivant, ainsi que des transitions de l'état Délétion d'un nœud à l'état Insertion du même nœud. Cet abandon fait évidemment diminuer le nombre de paramètres du HMM, mais il est peu argumenté par son auteur. On trouve dans [Durbin *et al.*, 1998] la mention :

> We have added transitions between insert and delete states, as [Krogh and Haussler] did, although these are usually very improbable. Leaving them out has negligible effect on scoring a match, but can create problems when building the model.

Il semble que Sean Eddy ait eu l'occasion de constater effectivement le fait que les transitions I→D et D→I sont de peu d'utilité en règle générale, mais à notre connaissance on ne retrouve ceci dans aucune publication. L'intéressé n'a pas répondu à un courriel dans lequel nous lui demandions sur quelle base concrète s'était opéré l'abandon desdites transitions pour aboutir à l'architecture Plan7. La déclaration d'« inutilité » semble d'ailleurs contradictoire avec ce qu'a prouvé une étude comparant SAM et HMMER et publiée en 2002, à savoir que les modèles produits par SAM (version 3.2) étaient significativement meilleurs que ceux produits par HMMER version 2.2g [Madera et Gough, 2002]. Quoi qu'il en soit, cette question de savoir si le choix de sept transitions seulement par nœud est devenue une question de sophiste, étant donné l'immense succès qu'a rencontré HMMER,

alors que SAM n'est plus beaucoup utilisé aujourd'hui [2].

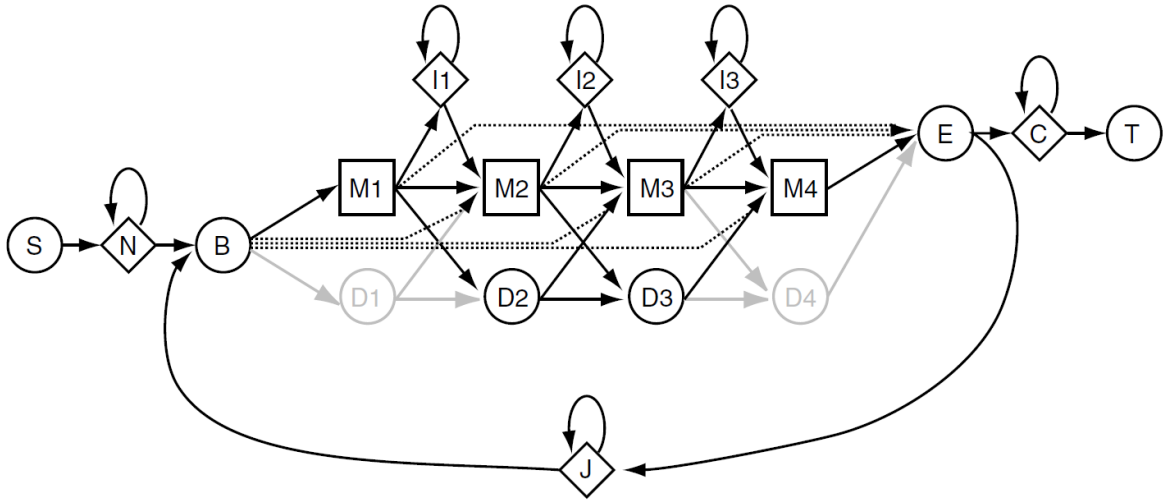


Figure 3.6. *L'architecture complète Plan7 des HMM profils des suites* HMMER*2.x et* HMMER*3.0. Les états N, C et J permettent de boucler autour du cœur du modèle et donc de faire des recherches capables de détecter plusieurs occurrences du même modèle au sein de la séquence cible.*

L'architecture Plan7 employée par HMMER apporte également l'introduction de nouveaux états *autour* du cœur du modèle. Ces états sont les états N, C et J de la figure 3.6 (représentées en traits pointillés, les transitions sortantes de l'état B et les transitions entrantes dans l'état E jouent aussi un rôle dans ce qui suit). Ils permettent, selon les valeurs données à leurs transitions entrantes et sortantes, d'effectuer à partir du même cœur de modèle (les n nœuds comprenant chacun un état Match, un état de Délétion et un état d'Insertion) plusieurs types de modélisations et donc de recherches subséquentes :

1. On peut vouloir rechercher à faire correspondre tout ou partie d'une séquence cible potentielle à la *totalité* du modèle. C'est le mode dit « glocal », qui est employé par défaut. Les correspondances entre modèle et séquence seront *globales* par rapport au modèle et *locales* par rapport à la séquence. Dans ce cadre, une séquence dans laquelle on ne trouverait une correspondance réelle qu'à une partie du modèle verrait son score pénalisé par le passage à travers de nombreux états de délétion (rappelons

2. Cette désaffection de SAM au profit de HMMER tient à plusieurs choses, en particulier au fait que le code source de SAM n'a pas été rendu public, que la documentation accompagnant HMMER est autrement plus lisible que celle accompagnant SAM, que HMMER est d'usage plus simple que SAM et enfin que le dynamisme de Sean Eddy et de sa petite équipe de programmeurs ont rendu HMMER incroyablement efficace en termes de temps de calcul, tout en faisant l'objet d'une diffusion et d'un support assez exemplaires.

que le score d'une séquence dans le HMM se calcule à partir du produit des probabilités d'émission et de transition déclenchées le long du chemin emprunté par la séquence dans le HMM, et qu'en général pour un MHM « normal », par construction les transitions d'un état Match vers le suivant sont toujours les plus probables). Par contre, les états N et C « absorbent » les préfixes et suffixes de la séquence cible flanquant l'occurrence statistique du modèle sans pénalités de score (les deux états se comportent comme le modèle nul, cf. supra). De plus, pour permettre de trouver plusieurs occurrences du modèle dans la séquence, les transitions E→J, J→J et J→B ne sont pas à 0.

2. On peut rechercher éventuellement *plusieurs* occurrences statistiques *partielles* du modèle dans une seule séquence cible, lesdites occurrences étant non chevauchantes. Ce mode correspond à un cadre « totalement local », c'est-à-dire local par rapport à la séquence *et* local par rapport au modèle : les transitions en traits pointillés de la figure 3.6 ne sont pas à 0, et on peut donc entrer ou sortir du modèle à tout moment sans pénalité.
3. L'alignement local de type Smith-Waterman est possible aussi, en reprenant les mêmes paramètres que précédemment mais en annulant E→J. On a alors un mode « totalement local » mais simple hit.
4. Enfin, l'alignement de type Needleman et Wunsch est possible aussi, en reprenant les mêmes paramètres qu'en (1) mais en imposant une seule occurrence du modèle dans la séquence ($t_{EJ} = 0$).

Comme on le voit, l'architecture Plan7 introduit dans le modèle lui-même une flexibilité qu'on n'a pas aussi explicitement avec SAM, lequel introduit seulement le concept de « Free Insertion Modules » correspondant plus ou moins aux états N et C de Plan7.

Étapes de la construction d'un modèle avec *hmmbuild*

hmmbuild est le composant de la suite HMMER chargé de la construction de modèles. Il procède en plusieurs étapes :

1. *pondération globale* des séquences par rapport aux paramètres du modèle a priori. On mesure le degré global d'informativité contenu dans l'alignement, en terme de nombre « effectif » de séquences. Par exemple, un alignement d'entrée contenant N séquences toutes identiques recevra un poids global équivalent à celui d'une séquence unique. Plusieurs méthodes existent pour déterminer cette pondération. Pour HMMER2 ce nombre effectif est égal au nombre de clusters de séquences construits par une approche de lien simple entre séquences partageant plus d'un certain pourcentage d'identité [Henikoff et Henikoff, 1992]. On peut considérer ce poids total comme étant le reflet du crédit que l'on accorde aux données observées

dans l'alignement, par rapport à la confiance que l'on a placée dans la distribution de fond,

2. *pondération relative* des séquences les unes par rapport aux autres (cf. un peu plus loin en section 3.4). C'est la méthode des poids basés sur l'arbre de Gestein/Sonnhammer/Chothia qui est employée par défaut dans HMMER2,
3. construction de l'architecture du modèle, c'est-à-dire détermination des colonnes de l'alignement d'entrée qui vont faire l'objet d'une modélisation sous la forme d'un état Match. Par défaut HMMER 2 utilise un algorithme peu intuitif de « maximum a posteriori ». Cet algorithme est censé construire l'architecture qui maximise la probabilité des séquences en entrée, mais il contient pour ce faire des étapes d'estimation de probabilités qui s'appuient sur des distributions a priori, et il n'est pas rare que cette technique conduise à des modèles dont certains états Match sont construits à partir d'un seul acide aminé observé sur une colonne pour le reste pleine de gaps. Cette méthode par défaut de construction de l'architecture a d'ailleurs été abondonnée par HMMER 3, où elle ne figure pas même sous la forme d'une option.
4. calcul des paramètres du HMM (émissions et transitions) en fonction des comptages effectués à partir de l'alignement en entrée, connaissant les poids relatifs des séquences, les trois distributions de fond (l'une pour les émissions issues des états Match, la deuxième pour les émissions issues des états d'Insertion et la troisième pour les transitions) et le poids total de l'alignement.

Distributions a priori

Les distributions a priori sont encodées « en dur » dans HMMER. Elles ont été proposées par Karplus, Sjölander, Mitchison ou Eddy lui-même.

En ce qui concerne les émissions d'acides aminés issues des états Match, la distribution a priori est constituée par un mélange de Dirichlet à 9 composantes (Blocks9) déterminé par Kimmen Sjölander [Sjölander *et al.*, 1996] en 1996 à partir d'alignements issus de la base BLOCKS [Henikoff et Henikoff, 1991]. Les mélanges de Dirichlet offrent l'avantage de donner des pseudocomptes adaptés au profil de chaque site : les acides aminés observés font pencher la balance vers une ou plusieurs des composantes du mélange, selon par exemple l'hydrophobicité ou la polarité des acides aminés observés. C'est à notre connaissance à ce jour la technique de pseudocomptes la plus aboutie qui soit utilisée pour la paramétrisation de modèles de séquences protéiques (cf. supra).

La distribution a priori pour les émissions sur les états d'Insertion est une distribution apprise à partir d'une ancienne version de Pfam (Pfam 1.0, novembre 1996). La distribution résultante présente un biais en faveur des résidus hydrophiles et fuit plutôt les hydrophobes. Ce comportement représente en effet plutôt bien les acides aminés pré-

sents dans les boucles et les coudes, ces sous-structures qui permettent des changements de direction de la structure tertiaire et se trouvent en général à la surface des protéines, donc plutôt en milieu hydrophile.

La distribution a priori pour les transitions a été établie par Graeme J. Mitchison « probablement sur une des premières versions de Pfam » (Sean Eddy lui-même avoue dans le code source de HMMER que les notes décrivant l'origine exacte de cette distribution ont été égarées). Ce travail a donné trois densités de Dirichlet (l'une pour les transitions quittant les états Match, une deuxième pour les transitions quittant les états d'Insertion et la troisième pour les états de Délétion) à une composante chacune. Elles donnent bien sûr le poids le plus important aux transitions M→M, un ratio $\frac{M\to I}{M\to D}$ égal à 2, à peu près autant de $I \to I$ que de $I \to M$ et environ deux fois plus de transitions $D \to M$ que de transitions $D \to D$. Ceci signifie que la croyance a priori tend à concevoir des modèles adaptés à des alignements où les délétions par rapport au consensus sont courtes et peu nombreuses en regard des insertions par rapport à ce même consensus.

Détermination des propriétés statistiques des distributions de score avec *hmmcalibrate*

Une recherche de protéines homologues à partir d'un modèle statistique construit sur une famille de protéines va affecter à toute séquence *seq* testée contre le modèle un *score* S, qui est donné pour HMMER par la formule suivante :

$$S = \log_2 \left(\frac{\Pr(\text{seq}|\mathscr{M})}{\Pr(\text{seq}|\mathscr{M}_{\text{nul}})} \right) \quad , \tag{3.9}$$

dans laquelle $\mathscr{M}_{\text{nul}}$ désigne ce qu'on appelle le *modèle nul*, censé représenter des séquences aléatoires. Pour HMMER, le modèle nul est constitué d'un seul état bouclant sur lui-même selon une probabilité qui dépend directement de la longueur attendue a priori pour les séquences (probabilité fixée à 350/351, c'est-à-dire pour une longueur moyenne de séquence fixée à 350 acides aminés), et émettant des acides aminés selon une distribution de fond correspondant à leurs fréquences d'apparition dans la base Swiss-Prot 34. Le rapport présent dans l'équation (3.9) permet donc de ne pas pénaliser automatiquement les séquences longues (le score de celles-ci étant un produit de nombres inférieurs à 1 en quantité plus importante que pour les séquences courtes).

Toute séquence testée contre le modèle reçoit donc un score qui reflète combien le modèle présenté est plus adapté que le modèle nul pour décrire ladite séquence. Étant donné un score, on peut se poser la question de sa *significativité statistique*. Pour répondre à cette question, il faut avoir une idée de la distribution des scores produits par le HMM

lorsqu'on lui présente un grand nombre de séquences aléatoires.

La vaste majorité des séquences aléatoires va obtenir des scores « moyens » à travers le HMM. Ce qui nous intéresse tout particulièrement, c'est en revanche la distribution des scores aux valeurs extrêmes, et plus particulièrement celle des meilleurs scores. On se trouve alors dans le cadre de la théorie dite des « valeurs extrêmes », qui a proposé plusieurs distributions convenant à la représentation statistiques d'événements rares à extrêmement rares. De telles distributions peuvent être décrites à l'aide de paramètres, et ce sont ces paramètres que le programme *hmmcalibrate* calcule à partir d'un modèle et de tirages aléatoires de séquences. Une fois ces paramètres établis, on pourra déterminer pour tout score S significativement élevé, combien il sera statistiquement rare : la *p-valeur* nous dira quelle était la probabilité au départ de tirer aléatoirement une séquence qui atteigne ou dépasse le score S, tandis que l'*e-valeur* nous dira combien de séquences on pouvait s'attendre à voir produire un score supérieur ou égal à S compte tenu de la taille de la base de séquences sur laquelle on aura fait chercher le HMM. Dans la famille des distributions aux valeurs extrêmes, HMMER 2.x utilise des distributions de Gumbel [Karlin et Altschul, 1990; Mott, 1992; Bundschuh, 2002].

hmmsearch, pour rechercher des *hits* dans les bases

C'est précisément le rôle du programme *hmmsearch* : ce dernier produit à partir d'un HMM et d'une base de séquences, une liste de ce qu'on appelle des *hits*, c'est-à-dire la liste des séquences de la base ayant produit les plus forts scores et les *e*-valeurs les moins élevées. Si une séquence combine score élevé et *e*-valeur faible, alors c'est très probablement une séquence homologue aux séquences ayant servi à concevoir le HMM. Une *e*-valeur faible associée à un score pas si élevé que cela (parfois même négatif) est souvent aussi l'indice d'une séquence apparentée aux séquences d'apprentissage, mais avec un modèle qui est tout de même relativement mal conçu pour la séquence cible. En effet, il peut arriver dans certains cas (surtout lorsque les séquences d'apprentissage sont très divergentes) que le modèle nul représente mieux la séquence cible que le modèle HMM, mais que les paramètres de l'EVD (*Extreme Value Distribution*) soient tels que même un score assez négatif soit déjà significativement élevé.

On contrôle le nombre de séquences renvoyées par la recherche en donnant à *hmmsearch* un seuil d'*e*-valeur (classiquement 10). Comme d'habitude en statistique, on parlera de *vrais positifs* lorsque les séquences apparaissant sous le seuil de recherche sont effectivement des séquences homologues au modèle, de faux positifs lorsque des séquences non homologues apparaissent sous le seuil, et de faux négatifs lorsque des séquences homologues arrivent au-delà du seuil et ne sont donc pas incluses dans le résultat d'une recherche effectuée avec *hmmsearch*. Le but de tout modèle est d'être capable de détecter tous les vrais positifs sans inclure aucun faux positif sous le seuil.

3.3.5 HMMER 3.0

La version 3.0 de HMMER, sortie le 28 mars 2010, a été brièvement décrite par Sean Eddy dans [Eddy, 2009]. Elle apporte plusieurs nouveautés, parmi lesquelles :

- HMMER 3.0 a abandonné totalement la détermination des scores par l'algorithme de Viterbi, pour lui préférer l'algorithme dit de « forward ». En effet, l'approche de Viterbi consiste à calculer la probabilité d'une séquence sachant le modèle en s'en tenant exclusivement au meilleur alignement possible pour cette séquence, appelé ici π^* : $S_{M,\text{Viterbi}}(\text{seq}) = \log_2\left(\frac{\Pr(\text{seq},\pi^*|M)}{\Pr(\text{seq}|M_{\text{nul}})}\right)$. L'approche dite « forward », elle, prend en compte la totalité des alignements possibles de la séquence dans le modèle M : $S_{M,\text{forward}}(\text{seq}) = \log_2\left(\frac{i\sum_\pi \Pr(\text{seq},\pi|M)}{\Pr(\text{seq}|M_{\text{nul}})}\right)$. On peut donc légitimement argumenter que l'approche *forward* est plus juste que sa contrepartie de Viterbi, même si dans la plupart des cas, tout le « poids » de la distribution des probabilités jointes $\Pr(\text{seq},\pi)$ se trouve concentré sur le terme $\Pr(\text{seq},\pi^*)$, si bien qu'alors $S_{M,\text{forward}}(\text{seq}) \simeq S_{M,\text{Viterbi}}(\text{seq})$.
- Dans la même logique, HMMER 3.0 présente les résultats des recherches à partir d'un modèle en donnant non seulement le meilleur alignement, mais en fournissant également une *enveloppe* pour les alignements les plus probables : on ne déclare plus « les positions i à j de la séquence cible correspondent au modèle », mais : « le meilleur alignement rencontre les positions i à j de la séquence, mais on trouve des alignements qu'on ne saurait ignorer, qui partent au plus tôt de la position $i' <= i$ et qui vont au plus loin jusqu'à la position $j' >= j$. » Cette notion d'incertitude dans un alignement, ainsi présentée, est nouvelle pour ce qui concerne la classe d'outils qui nous intéresse ici.
- la recherche dans les bases de données à partir d'un HMM est devenue extrêmement rapide (sachant que la construction d'un modèle avec HMMER n'a jamais été un point consommateur de ressources processeur). Ceci est dû essentiellement au fait que HMMER 3 incorpore des heuristiques pour filtrer progressivement la base de séquences qu'on fournit en entrée de *hmmsearch*. La première et la plus importante de ces heuristiques consiste en une recherche initiale d'alignements partiels sans gaps, faisant intervenir des états Match successifs du modèle. Sean Eddy s'est directement inspiré des heuristiques présentes dans BLAST (les *graines* à partir desquelles BLAST construit des correspondances de séquences) pour tenter de faire aussi bien en termes de rapidité.
- enfin, l'étape *hmmcalibrate* disparaît de HMMER 3.0, suite à des avancées théoriques [Olsen *et al.*, 1999; Altschul *et al.*, 2001; Eddy, 2008] qui font qu'on peut se passer de l'estimation coûteuse des deux paramètres de la distribution de Gumbel : [Eddy, 2008] conjecture en effet que pour des modèles probabilistes d'alignements locaux, les scores de Viterbi sont distribués selon une loi de Gumbel de paramètre constant $\lambda = \log(2)$, alors que les scores issus de l'algorithme *forward* sont, eux, ex-

ponentiellement distribués avec le même paramètre λ, du moins en ce qui concerne la queue de distribution correspondant aux plus forts scores. Ces conjectures n'ont à notre connaissance pas été remises en cause à ce jour. Une partie de la rapidité accrue des analyses avec HMMER 3.0 tient à l'économie de temps de calcul réalisée grâce à ces conjectures.

À partir d'un alignement multiple fourni en entrée, les étapes de construction d'un HMM avec HMMER3.0 sont les suivantes :

1. *pondération relative* des séquences les unes par rapport aux autres (cf. plus loin, section 3.4). C'est la méthode des poids *position-based* développée par les époux Henikoff qui est employée par défaut,
2. construction de l'architecture du modèle, c'est-à-dire détermination des colonnes de l'alignement d'entrée qui vont faire l'objet d'une modélisation sous la forme d'un état Match. Par défaut on sélectionne les colonnes qui comportent plus d'un certain pourcentage de caractères qui ne sont pas des gaps (50% par défaut). Dans HMMER 3.0, les séquences considérées comme des *fragments* (leur taille est inférieure de moitié au moins par rapport à la taille moyenne des séquences de l'alignement) font l'objet d'un traitement particulier : les gaps situés en début et en fin de cette séquence alignée seront ignorés dans le calcul de la « fraction de remplissage » des colonnes en question,
3. *pondération globale* des séquences par rapport aux paramètres du modèle a priori. On mesure le degré global d'informativité contenu dans l'alignement, en terme de nombre « effectif » de séquences. Par exemple, un alignement d'entrée contenant N séquences toutes identiques recevra un poids global équivalent à celui d'une séquence unique. Plusieurs méthodes existent pour déterminer cette pondération. Alors que dans HMMER2 ce nombre effectif était égal au nombre de clusters de séquences construits par une approche de lien simple entre séquences partageant plus d'un certain pourcentage d'identité [Henikoff et Henikoff, 1992], HMMER 3.0 définit ce poids total de l'alignement de façon à ce que l'entropie relative moyenne des états Match (calculée par rapport à le distribution de background, encodée dans les routines de construction du HMM) soit égale à une certaine quantité (exprimée en bits par position, par défaut 0,59). L'alignement étant fixé, si l'on fait l'hypothèse minimale consistant à supposer qu'au moins une des colonnes sélectionnées pour fabriquer les états Match diffère dans sa composition de la distribution de background, alors augmenter cette quantité demandée d'entropie relative moyenne revient à augmenter le poids total accordé aux séquences. On peut considérer ce poids total comme étant le reflet du crédit que l'on accorde aux données observées dans l'alignement, par rapport à la confiance que l'on a placée dans la distribution de fond. En 1999, les auteurs de SAM préconisaient déjà de fixer un gain moyen d'entropie relative de $0,5$ bit par colonne [Cline *et al.*, 1999],

4. calcul des paramètres du HMM (émissions et transitions) en fonction des comptages effectués à partir de l'alignement en entrée, connaissant les poids relatifs des séquences, les différentes distributions de fond (émissions sur les états Match et sur les états d'Insertion, transitions) et le poids total de l'alignement, par une approche basée sur les mélanges de Dirichlet,
5. calcul des paramètres des distributions statistiques qui vont permettre de donner des p-valeurs et des e-valeurs aux cibles trouvées lors d'une recherche avec ce HMM. Grâce à la conjecture sur le paramètre λ de la loi de Gumbel (cf. plus haut), il s'agit en fait simplement d'estimer le second des deux paramètres de la loi.

3.4 Pondérer les séquences d'apprentissage pour maximiser l'informativité du modèle

L'ensemble des séquences d'apprentissage pour un HMM est souvent au moins partiellement redondant : on y trouve plusieurs séquences très semblables les unes aux autres, à côté d'autres plus singulières. Cette inégalité de représentation peut avoir plusieurs causes, la plus courante étant celle d'un échantillonnage taxonomique inégal dans les bases de données : on peut par exemple avoir recensé de nombreuses séquences de globines chez les mammifères, alors que peu de globines appartenant à d'autres classes auront été séquencées et annotées. En général, la volonté du modélisateur lorsqu'il construit un HMM pour une famille de séquences est de concevoir un modèle statistique capable de repérer *toutes* les séquences homologues (c'est-à-dire en leur attribuant un score qui soit le plus grand possible), jusqu'à un degré élevé de mutations accumulées. Pour ce faire, il est important de compenser le biais de construction de l'ensemble d'apprentissage lorsqu'on détermine les paramètres du modèle, lesquels sont des distributions de probabilités, qu'il s'agisse d'émissions ou de transitions [Lüthy *et al.*, 1994; Thompson *et al.*, 1994b]. Le cadre général adopté pour ce faire est celui de la *pondération des séquences* les unes par rapport aux autres. Une fois que l'on aura déterminé des poids (valeurs positives dont la somme n'est pas nécessairement normalisée) pour chacune des séquences de l'ensemble d'apprentissage, la suite du processus de construction du modèle se fera à partir d'un ensemble d'observations obtenu en multipliant les observations faites sur les séquences (émissions de caractères ou transitions entre états du HMM) par les poids respectifs de celles-ci.

Deux classes d'algorithmes de pondération sont en vigueur aujourd'hui dans les processus d'apprentissage de HMMs profils : d'un côté ceux qui s'appuient sur une phylogénie sous-jacente reliant les séquences entre elles, et de l'autre ceux qui ne supposent pas l'existence d'un arbre mais travaillent à partir de distances (deux-à-deux) entre séquences. Nous présentons dans ce qui suit quelques-uns des algorithmes de pondération

de séquence les plus répandus, trois d'entre eux étant implémentés dans le populaire HMMER3, d'autres dans son prédécesseur HMMER2.

Il faut remarquer que dans cette section on passe sous silence le traitement des gaps dans les alignements, qu'il s'agisse de calculer des distances d'édition entre séquences ou de construire un arbre à partir de l'ensemble d'apprentissage. Historiquement, les auteurs s'étant penchés sur la question ont systématiquement présenté des algorithmes fonctionnant sur des séquences sans trous. La prise en compte de ces derniers se fait habituellement en adoptant une stratégie simple : ou bien on ignore les colonnes contenant des gaps dans les alignements deux à deux, ou bien le coût d'édition entre un gap et une lettre quelconque est nul, ou bien il est fixe quelle que soit la lettre alignée contre un gap. Lorsque les poids dérivent de distances entre séquences calculées sur les alignements globaux deux-à-deux, la présence de gaps peut être traitée de façon classique en définissant par exemple des pénalités d'ouverture et d'élongation de gap. En revanche, lorsque les poids sont calculés à partir de comparaison site par site, la littérature n'est pas claire quant à la prise en compte des gaps, et les auteurs semblent systématiquement ignorer les gaps [Vingron et Sibbald, 1993; Henikoff et Henikoff, 1994; Krogh et Mitchison, 1995].

On note $s_i,\ i \in [1,n]$ les séquences d'apprentissage et $d(a,b)$ la distance d'édition entre deux séquences a et b. Les w_i désignent les poids calculés.

3.4.1 Pondération sans construction d'arbre

Une première méthode basée sur les distances

La première méthode de pondération des séquences à partir de distances d'édition a été publiée par Vingron & Argos [Vingron et Argos, 1989]. Les auteurs établissent une matrice de distances entre séquences, constituée simplement des distances d'édition deux à deux $d(s_i,s_j)$. Ils calculent ensuite pour chaque séquence d'apprentissage sa distance aux autres séquences, et la somme de telles distances donne le poids relatif de cette séquence, après normalisation :

$$w_i = \frac{\sum_{k\neq i} d(s_i,s_k)}{\sum_j \sum_{k\neq j} d(j,k)} \tag{3.10}$$

Selon cette première méthode, plus une séquence diffère des autres et plus on lui accordera d'importance lorsqu'il s'agira d'estimer sa contribution à l'ensemble pendant la phase d'apprentissage des paramètres du modèle. Mais comme on peut s'en rendre compte rapidement, une telle méthode n'est pas assez « sévère » pour réduire le poids de séquences plus ou moins identiques. Prenons par exemple une famille de cinq séquences a, b, c, d et e dans laquelle deux séquences figurent chacune en double exemplaire, en dehors de quoi les séquences sont équidistantes entre elles ($a=b$, $c=d$ et $d(a,c)=d(a,e)=d(c,e)$).

On obtient alors pour les séquences présentes en double un poids individuel de $\frac{3}{16}$ contre seulement $\frac{1}{4}$ pour la séquence *e* qui est singulière. Or on s'attendrait plutôt à voir $\frac{1}{6}$ pour chacune des séquences en double contre $\frac{1}{3}$ pour la séquence unique.

La méthode dite de Voronoï

Ultérieurement, d'autres algorithmes de pondération par méthodes de distance ont été conçues pour tenter de corriger de tels problèmes. Sibbald et Argos ont ainsi publié en 1990 une autre méthode, basée sur les diagrammes de Voronoï [Sibbald et Argos, 1990]. L'idée de tels diagrammes est de partir d'un ensemble de points x_i dans un espace multidimensionnel $\mathscr{E}$ pour partitionner ensuite cet espace en un certain nombre de polyèdres Ω_i centrés chacun en un des points x_i. Les partitions Ω_i sont déterminées de telle manière que :

$$\forall i,\ \Omega_i = \{s \in \mathscr{E},\ \forall j \neq i\ d(s,x_i) < d(s,x_j)\} \tag{3.11}$$

Le *volume* du polyèdre Ω_i est ensuite pris comme poids affecté à x_i.

Ici l'espace $\mathscr{E}$ est l'ensemble des séquences biologiques et les points x_i sont les séquences d'apprentissage. $\mathscr{E}$ n'étant pas naturellement défini comme un espace métrique de dimension finie, il est très difficile d'estimer mathématiquement les frontières entre les Ω_i et donc les volumes ou poids affectés aux séquences. L'idée de [Sibbald et Argos, 1990] est d'échantillonner très partiellement l'espace $\mathscr{E}$ en effectuant des altérations élémentaires des séquences d'apprentissage x_i : les auteurs construisent ainsi l'ensemble de toutes les séquences possibles obtenues par une mutation par rapport à l'une des séquences d'apprentissage. Pour travailler avec un ensemble de séquences virtuelles qui soit à la fois de taille raisonnable et composé de séquences ayant une certaine chance d'apparaître dans la réalité, ils se cantonnent à fabriquer des séquences virtuelles en sélectionnant aléatoirement pour chaque position un caractère *vu au moins une fois* dans l'alignement sur le site en question. L'ensemble des combinaisons possibles est examiné. Pour chacune des séquences virtuelles ainsi créées, on calcule sa distance d'édition avec chacune des séquences de l'ensemble d'apprentissage, après quoi celle des séquences d'apprentissage qui est la plus proche de la séquence mutée gagne un point : celle-ci rentre dans son voisinage Ω. En cas d'ex-æquo, le point est partagé. Après normalisation, la somme des points gagnés par une séquence donne son poids relatif.

Cette méthode présente l'attrait d'une description théorique de l'espace des séquences accompagnée de l'idée de son partitionnement, mais en pratique, la chose devient vite computationnellement impraticable lorsque les séquences sont nombreuses et très divergentes. Le recours à des méthodes de Monte Carlo pour échantillonner l'espace $\mathscr{E}$ est possible, mais la pauvreté de l'échantillonnage peut rendre contestable toute la méthode.

Clustering des séquences

Henikoff et Henikoff ont publié en 1992 le jeu de matrices sans doute le plus populaire jusqu'ici parmi les structuralistes et très largement utilisé dans de nombreux logiciels d'alignement, les matrices BLOSUM [Henikoff et Henikoff, 1992]. Celles-ci ont été construites à partir de blocs de sites extraits d'alignements multiples. Pour réduire l'importance de séquences très proches lors du processus de comptage des paires d'acides aminés alignés, Henikoff & Henikoff ont proposé de grouper les séquences les plus proches en un certain nombre de *clusters* pour ensuite attribuer un poids de 1 à chacun de ces clusters.

Cette approche a perduré et a été notamment retenue pour être implémentée dans la suite HMMER, de la façon suivante : un seuil de similarité est défini tout d'abord par l'utilisateur ($s = 0,62$ par défaut). On forme ensuite des groupes de séquences en regroupant progressivement celles présentant une proportion au moins égale à s de sites identiques l'une par rapport à l'autre. Ce processus itératif d'agglomération est dit « à simple lien » (*single linkage*) car on intègre une séquence x dans le cluster C en formation dès lors qu'il existe au moins une séquence y de C avec laquelle x présente au moins une proportion s de sites identiques. Enfin, le poids d'un cluster est réparti uniformément entre toutes les séquences le composant.

Henikoff & Henikoff : poids basés sur les positions

Dans un papier de 1994, Henikoff et Henikoff [Henikoff et Henikoff, 1994] ont proposé une méthode rapide de pondération des séquences qui ne partirait pas de distances deux-à-deux calculées globalement sur toute la longueur de l'alignement, mais plutôt d'un schéma consistant à mesurer site par site la diversité constatée dans l'alignement d'apprentissage, et à « récompenser » chacune des séquences proportionnellement à leur apport à la diversité du site. Pour ce faire, ils attribuent le même « crédit unitaire » à chacun des résidus vus sur un site donné. Par exemple, sur un site composé uniquement de I, de L et de V, l'isoleucine, la leucine et la valine ont chacune un crédit de $\frac{1}{3}$. Ce crédit est ensuite partagé à égalité entre les séquences qui présentent une isoleucine, tandis que celles qui présentent une leucine se partagent également entre elles le crédit « leucine ». Au bout du compte, une séquence qui aura été la seule à présenter une isoleucine recevra l'intégralité du crédit de $\frac{1}{3}$, alors que les autres séquences se partageront les crédits accordés à L et à V. On obtient le poids relatif pour une séquence en sommant les crédits ainsi attribués position par position, tout le long de la séquence.

Ce schéma de pondération des séquences est celui qui est adopté par défaut par HMMER3.

Approche basée sur la maximisation de l'entropie

Enfin, pour clore le chapitre concernant les méthodes de pondération sans construction d'arbre, signalons le travail de Krogh et Mitchison [Krogh et Mitchison, 1995], qui ont publié en 1995 une méthodologie de pondération des séquences dont le but était de maximiser l'entropie informationnelle (entropie de Shannon) des distributions pondérées d'acides aminés observés sur les différents sites de l'alignement. Proposée dans HMMER2, cette méthode a été abandonnée dans HMMER3.

3.4.2 Approches arborées

Dans tout ce qui précède, les séquences biologiques sont pondérées en fonction de leurs distances d'édition deux à deux, ou bien en fonction de l'apport informationnel des caractères qu'elles portent, lequel apport est évalué site par site. Toutes ces méthodes reviennent d'une certaine façon à évaluer la distance de chacune des séquences d'apprentissage à une séquence hypothétique qui serait une sorte de centroïde pour l'ensemble de ces séquences.

Si l'on sait construire un arbre dont les séquences d'apprentissage sont les feuilles, alors on peut envisager la racine de cette arbre comme portant une séquence virtuelle de référence, et évaluer la distance évolutive entre la racine et chacune des séquences d'apprentissage pour pondérer ces dernières. L'idée est alors de donner un poids plus élevé aux séquences se trouvant au bout de longues branches, alors que deux séquences très voisines se partageront un poids inférieur.

Plusieurs techniques ont été mises au point qui se chargent de résoudre le problème de la pondération des séquences en s'appuyant sur un arbre. Dans tout ce qui suit, on supposera que l'arbre en question est déjà construit. En pratique il est souvent inféré rapidement à partir de l'alignement des séquences d'apprentissage, en général par des méthodes de *Neighbour-Joining* ou de clustering hiérarchique de type UPGMA (cf. chapitre 4).

Une application de la loi de Kirchhoff : pondération à la Thompson et al.

L'idée de Thompson et coauteurs dans [Thompson *et al.*, 1994b] est de calculer les poids des séquences aux feuilles d'un arbre en faisant l'analogie suivante : on assimile l'arbre à un réseau électrique dont les branches sont des conducteurs de courant faits d'un même matériau. La résistance qu'oppose une branche au passage du courant est donc strictement proportionnelle à sa longueur. Les feuilles étant au potentiel nul et la racine de l'arbre à un potentiel non nul arbitraire, la chute de potentiel entre deux points du réseau se calcule simplement par la loi d'Ohm : U = R x I. Le poids d'une séquence sera

proportionnel à l'intensité qui arrive sur la feuille la portant.

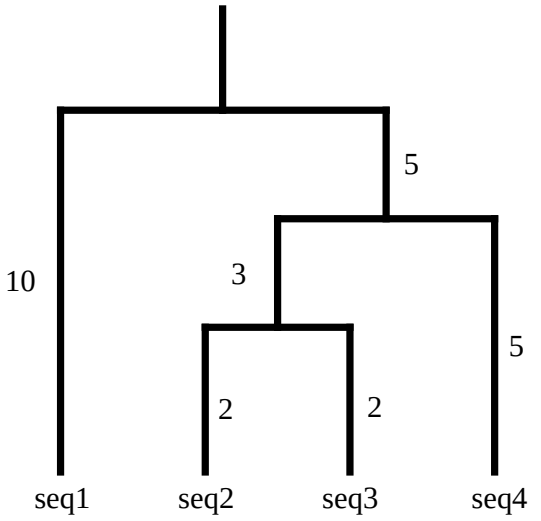


Figure 3.7. *Kirchhoff : arbre avec ses longueurs de branche*

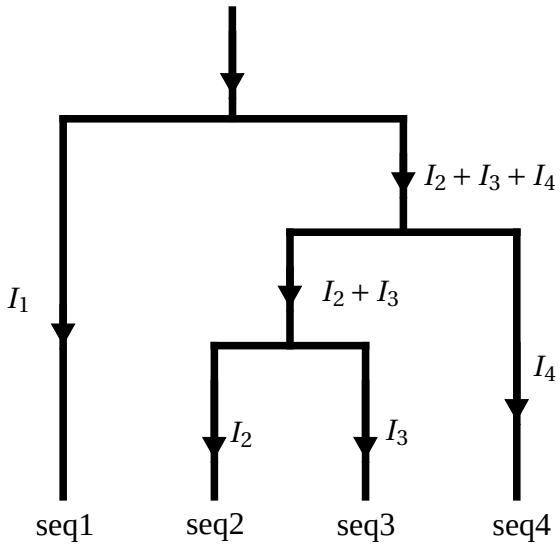


Figure 3.8. *Kirchhoff : intensités*

Le théorème de Kirchhoff énonce que les intensités se somment lors d'une bifurcation du réseau (cf. figure 3.8). En assimilant la longueur d'une branche à sa résistance, on obtient donc dans l'exemple de la figure 3.7 le jeu d'équations suivant :

$$
\begin{aligned}
V_{\text{racine}} &= 10I_1 && (3.12)\\
&= 5(I_2+I_3+I_4)+3(I_2+I_3)+2I_2 && (3.13)\\
&= 5(I_2+I_3+I_4)+3(I_2+I_3)+2I_3 && (3.14)\\
&= 5(I_2+I_3+I_4)+5I_4 && (3.15)\\
& && (3.16)
\end{aligned}
$$

En fixant le potentiel à la racine V_{racine} à une valeur arbitraire (par exemple 10), on a un système de quatre équations reliant les quatre inconnues que sont les intensités I_1 à I_4, qui se résoud en donnant : $I_1 = 1$, $I_2 = I_3 = \frac{1}{3}$ et $I_4 = \frac{2}{3}$. Les poids des séquences ne sont autres que les intensités normalisées à 1.

Le problème de cette approche est qu'elle est parfois contre-intuitive : la résistance étant plus élevée pour un conducteur plus long, si on considère une bifurcation avec une branche longue et une autre plus courte, l'intensité sera plus élevée dans la branche courte et la pondération favorisera les feuilles en bout de branche courte. C'est tout l'inverse de ce qu'on veut faire en donnant une prime aux séquences très divergentes. Cette anomalie

montre à quel point cet algorithme est sensible au positionnement de la racine de l'arbre.

Les poids à la Gerstein, Sonnhammer et Chothia

À l'occasion d'une publication [Gerstein *et al.*, 1994] concernant l'évolution du volume des protéines, Gerstein, Sonnhammer et Chothia (à qui l'on doit la base de données structurale SCOP) ont décrit une nouvelle méthode de pondération des séquences. Comme celle que nous venons de décrire ci-dessus, la méthode que nous appellerons GSC s'appuie sur un arbre reliant les séquences entre elles. L'algorithme proposé procède de bas en haut, des feuilles jusqu'à la racine. La première étape consiste à attribuer à chaque séquence un poids initial égal à la longueur de la branche qui la porte. Ensuite on procède itérativement de la façon suivante en remontant vers la racine : lorsqu'on rencontre un nœud, on partage le poids (longueur) de la branche située juste au-dessus de ce nœud entre toutes les séquences situées en dessous. Le partage de ce poids additionnel s'effectue en proportion de ce qu'étaient les poids des séquences à l'étape précédente. On présente le processus en images en figure 3.9.

Alors que la méthode des poids GSC était la méthode par défaut de pondération des séquences dans HMMER2, elle n'est qu'une option dans HMMER3, ce dernier lui préférant par défaut la méthode des poids définis par position (Henikoff & Henikoff, cf. section 3.4.1).

On peut se rendre compte de la variabilité des mesures de pondération ne serait-ce que sur le simple exemple introduit ici, en consultant le tableau 3.10, qui présente les pondérations normalisées (i.e. sommant à 1) pour les deux méthodes arborées vues ci-dessus et pour l'arbre donnée en figure 3.7.

Toutes les méthodes de pondération récapitulées ici, qu'elles s'apuient ou non sur un arbre sous-jacent, sont des méthodes « ad-hoc » qui ne jouissent en général *d'aucune justification théorique* proprement établie en lien avec les séquences biologiques, et la variabilité des poids issus des différentes méthodes induit évidemment des différences dans les modèles construits sur les alignements de séquences.

La notion de pondération des séquences, ou en tout cas de prise en compte différentielle des séquences, est pourtant centrale dans le processus de construction d'un modèle à partir d'un alignement, car elle pose directement la question de la *cible* des efforts du modélisateur : s'agit-il pour lui de concevoir un modèle de reconstruction ancestrale de la séquence à la racine de l'arbre ? Un modèle pour une séquence virtuelle qui serait au centre d'un volume défini par les points que constituent les séquences d'apprentissage ? Ou bien encore un modèle qui maximise la somme des vraisemblances des séquences d'apprentissage (maximum de discrimination) ? Tant que la cible (c'est-à-dire la fonction

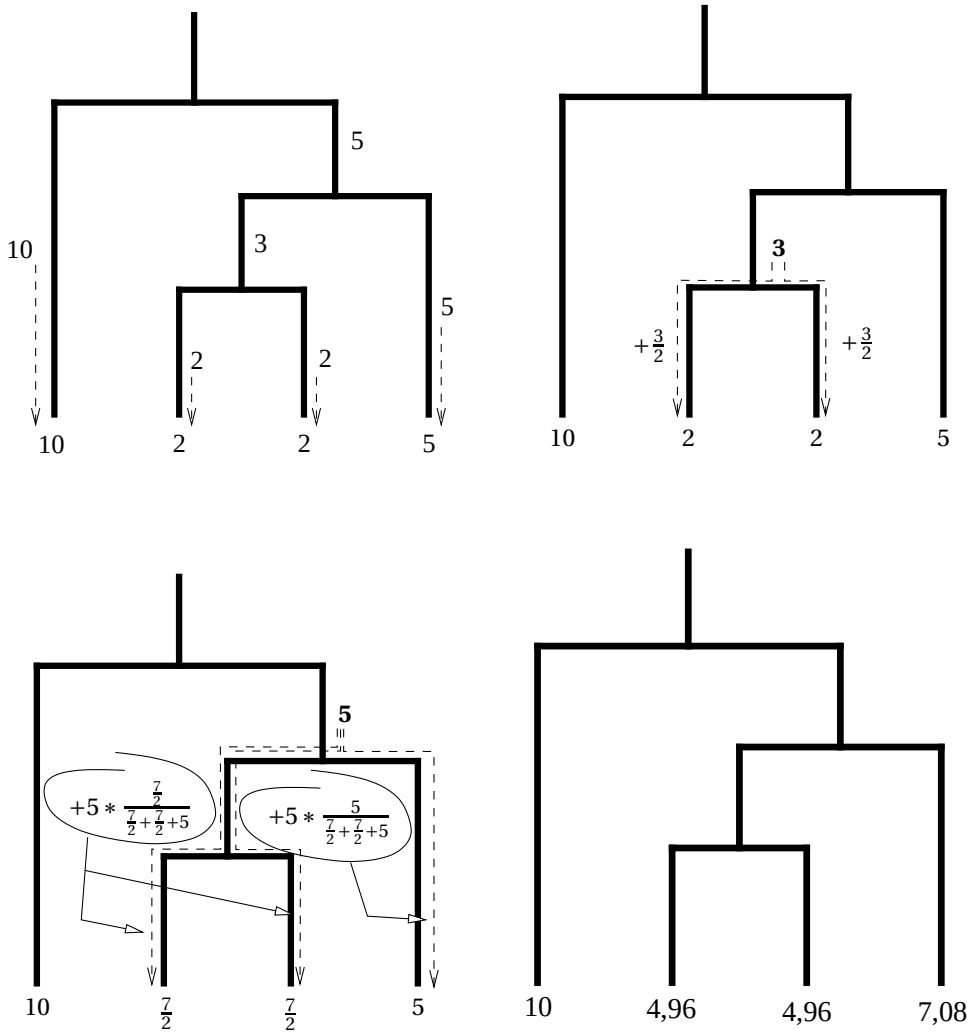


Figure 3.9. *Pondération selon Gerstein/Sonnhammer/Chothia. À chaque étape figurent les poids courants aux feuilles.*

objectif que l'étape de pondération est censée maximiser) n'est pas clairement établie, on ne sait dire quelle méthode est la plus appropirée.

Quoi qu'il en soit, la possibilité de pondérer des séquences en tirant parti de l'information disponible dans un arbre a rencontré une certaine popularité, sans que l'on soit parvenu jusqu'ici à en donner une implémentation qui ne souffre aucune contestation. Sean R. Eddy écrivait fort à propos en 2003 dans le manuel utilisateur de la suite HMMER version 2.3.2 (c'est nous qui traduisons) :

	seq1	seq2	seq3	seq4
Thompson	0.429	0.143	0.143	0.286
GSC	0.370	0.184	0.184	0.262

Figure 3.10. *Une comparaison des méthodes de pondération basées sur un arbre*

> Il n'existe aucune théorie satisfaisante derrière le choix d'une pondération pour les séquences, c'est là une étape *ad hoc*.[...] Un modèle correct incorporerait un modèle phylogénétique explicite (e.g. les « tree HMMs » introduits par Mitchison et Durbin en 1995). On a aujourd'hui un corpus croissant d'articles publiés par Jotun Hein, Ian Holmes, Bill Bruno, Richard Goldstein et d'autres, qui avancent vers la combinaison de HMMs profils et d'arbres phylogénétiques.

Et plus loin, lorsqu'il expose les méthodes de pondération *totale* de l'ensemble des séquences (vs. l'information a priori) :

> Il n'existe aucune théorie satisfaisante derrière l'approche totalement empirique de l'évaluation du poids global d'un alignement. Un jour, des méthodes explicitement phylogénétiques rendront tout cela obsolète.

Ces prédictions optimistes quant à l'avenir des approches conjuguant modèles probabilistes à la « HMM profil » et phylogénies ont totalement disparu du manuel que le même Sean R. Eddy a rédigé en 2010 pour la version 3 de HMMER, mais la littérature dont il parlait est néanmoins abondante et mérite que l'on s'y intéresse, ce que nous avons modestement fait dans cette thèse.

3.5 Sélectionner des colonnes d'intérêt dans un alignement, première étape du processus d'inférence d'un modèle

Qu'il s'agisse d'estimer une phylogénie ou de construire un HMM profil à partir d'un alignement de séquences, on en passe toujours au préalable par une étape de dé-bruitage de l'alignement consistant à repérer et à extraire les colonnes de celui-ci qui contiennent le plus de signal phylogénétique (si l'objectif est de construire une phylogénie) ou qui sont le mieux conservées (si l'objectif est de modéliser la famille de séquences alignées par un modèle de type HMM). Plusieurs auteurs ont montré en effet [Morrison et Ellis, 1997; Castresana, 2000; Talavera et Castresana, 2007] que les sites qui sont trop variables et/ou qui contiennent trop de gaps nuisent à la description verticale (phylogénie) ou longitudinale (HMM profil) de l'alignement, ce pour trois raisons principales :

- Les sites ayant évolué trop rapidement sont dits « saturés » du point de vue du signal phylogénétique, c'est-à-dire que la distribution des caractères constatée sur ces sites dans les séquences contemporaines informe plus sur la distribution d'équilibre du processus de substitution que sur les caractéristiques physico-chimiques propres au site considéré (qui sont d'ailleurs plus ou moins indéfinies, sans quoi le site n'aurait pas évolué aussi rapidement). Ce genre de sites saturés a tendance à provoquer le phénomène connu dit « d'attraction des longues branches », par lequel deux taxa vont avoir tendance à être vus comme proches l'un de l'autre lors du processus d'inférence phylogénétique justement à cause de la présence de tels sites saturés qui noient le signal phylogénétique derrière la ressemblance trompeuse de deux taxa à la distribution d'équilibre du processus évolutif.
- La présence d'un grand nombre de gaps au sein d'une même colonne de l'alignement pose un problème bien connu en statistique, celui de l'« overfitting » ou surapprentissage : ceci se produit lorsque l'on tente d'apprendre un modèle en partant de trop peu d'observations. On a alors tendance à construire un modèle qui s'approche trop du peu qu'on a observé, et qui ne permet pas de représenter avec suffisamment peu d'incertitude statistique un profil pertinent pour le site en question. Ce problème n'est que partiellement corrigé par le lissage des distributions estimées via l'ajout d'une information a priori, par exemple en utilisant un mélange de densités de Dirichlet (cf. page 45).
- Enfin, et ce n'est pas la moindre des raisons, un site au sein duquel les caractères montrent une trop grande variabilité peut tout à fait être le résultat d'une ou de plusieurs erreurs d'alignement. Les retirer en amont du processus d'inférence d'un modèle permet donc de ne pas construire sur de telles erreurs.

Nous présentons ci-dessous les quelques mesures le plus couramment utilisées pour mesurer l'informativité d'un site donné dans un alignement. Bon nombre d'entre elles s'appuient sur la notion d'entropie informationnelle, que nous décrivons ci-dessous. Dans toute cette section, nous considérons un site donné, qui contient ou non des gaps et qui est construit sur un alphabet de cardinal N (par exemple $N = 4$ s'il s'agit d'un alignement de séquence nucléotidiques ou $N = 20$ si l'on parle d'un alignement de protéines, c'est-à-dire d'acides aminés). Le vecteur $\mathbf{p} = (p_i)_{i\in[1,N]}$ représente le comptage des fréquences relatives des caractères observés sur le site en question, et on a donc $\sum_{i=1}^{N} p_i = 1$ avec $\forall i,\ 0 \leq p_i \leq 1$. On écrit $\log_k$ pour le logarithme à base k et on convient que $p_i \log(p_i)$ vaut 0 lorsque p_i est nul.

3.5.1 Mesures d'informativité basées sur l'entropie

Entropie de Shannon

C'est à Claude Elwood Shannon, mathématicien américain (1916–2001), que l'on doit la première mesure formelle de la *quantité d'information* contenue dans un message. Tout en introduisant lui-même ce concept de quantité d'information, Shannon en donnait la première définition en 1948 [Shannon, 1948] :

$$\mathrm{E}_{\mathrm{Sh}}(\mathbf{p}) = -\sum_i p_i \log_2(p_i) \tag{3.17}$$

Cette mesure s'apparente, comme l'a suggéré plus tard John Von Neumann, aux mesures d'entropie qui existent dans le champ de la thermodynamique physique, et qui décrivent l'état d'un ensemble de particules : plus l'entropie est élevée, plus les particules sont instables et plus le corps considéré est chaud. L'entropie d'un système de particules est minimale et nulle uniquement au zéro absolu (0K = −273 °C). Quoique incomplètement analogue, l'entropie informationnelle de Shannon respecte un certain nombre de propriétés :

1. $\forall \mathbf{p}, \quad 0 \leq \mathrm{E}_{\mathrm{Sh}}(\mathbf{p}) \leq \log_2(N)$
2. $\mathrm{E}_{\mathrm{Sh}}(\mathbf{p}) = 0 \quad \Leftrightarrow \quad \exists i,\ p_i = 1$
3. $\mathrm{E}_{\mathrm{Sh}}(\mathbf{p}) = \log_2(N) \quad \Leftrightarrow \quad \forall i,\ p_i = \frac{1}{N}$

Ainsi, l'entropie de Shannon donne une mesure simple de la diversité des caractères constituant un site : si chacun des 20 acides aminés apparaît le même nombre de fois, alors l'entropie est maximale, tandis qu'elle est nulle si le site est parfaitement conservé (i.e. constant).

On peut remarquer qu'en utilisant le logarithme à base N au lieu de logarithme à base 2 dans (3.17), on obtient une entropie évoluant dans $[0,1]$, ce qui est commode. Dans tout ce qui suit, N sera la base en vigueur (donc 20 pour les exemples de colonnes d'acides aminés.

On donne en figure 3.11 quelques exemples de valeurs d'entropie en fonction de la distribution des caractères observés sur un site au sein d'un alignement d'acides aminés.

composition	$\frac{1}{3}/\frac{1}{3}/\frac{1}{3}$	$\frac{1}{2}/\frac{1}{2}$	$\frac{9}{10}/\frac{1}{10}$
E_{Sh}	0,367	0,231	0,109

Figure 3.11. *Quelques valeurs d'entropie pour différentes fréquences observées ($N = 20$)*

Entropie de Shannon relative

L'un des problèmes que pose l'entropie de Shannon « classique » est qu'elle ne tient aucun compte de la proportion des caractères attendue, c'est-à-dire de l'information *a priori* dont on dispose et qui provient par exemple d'une connaissance globale des proportions de caractères rencontrés dans le Vivant. Si l'on s'intéresse aux acides aminés, on peut considérer par exemple la distribution observée sur la totalité de la base de données UniProtKB/TrEMBL au 28 juin 2011 (plus de 16 millions de séquences, totalisant quelque 5 milliards d'acides aminés) donnée en figure 3.12.

Ala (A)	8,64	Gln (Q)	3,87	Leu (L)	9,85	Ser (S)	6,70
Arg (R)	5,46	Glu (E)	6,12	Lys (K)	5,24	Thr (T)	5,62
Asn (N)	4,13	Gly (G)	7,12	Met (M)	2,48	Trp (W)	1,31
Asp (D)	5,30	His (H)	2,19	Phe (F)	4,03	Tyr (Y)	3,04
Cys (C)	1,26	Ile (I)	6,01	Pro (P)	4,72	Val (V)	6,75

Figure 3.12. *Composition en acides aminés de la base UniProtKB/TrEMBL release 2011_07, donnée en pourcentage pour chaque acide aminé*

On y voit par exemple que l'alanine (A) et la leucine (L) sont deux acides aminés très fréquents, alors que l'on rencontre rarement la cystéine (C) ou le tryptophane (W). Ainsi, on peut légitimement considérer qu'une colonne comportant des cystéines bien conservées recèle un signal plus « significatif » qu'une colonne pleine de leucines. Il est donc souhaitable de tenir compte de l'information a priori lorsque l'on veut mesurer la quantité d'information contenue dans une colonne.

C'est très exactement le rôle de la *divergence de Kullback-Leibler*[Kullback et Leibler, 1951] dont l'entropie *relative* de Shannon n'est qu'une traduction. Soit $\mathbf{p}$ le vecteur des fréquences observées et $\mathbf{q}$ celui correspondant à l'information a priori, on a :

$$\mathrm{E}_{\mathrm{Sh,rel}}(\mathbf{p}|\mathbf{q}) = D_{\mathrm{KL}}(\mathbf{p}||\mathbf{q}) = \sum_i p_i \log(p_i) - p_i \log(q_i) = \sum_i p_i \log\left(\frac{p_i}{q_i}\right) \tag{3.18}$$

C'est l'inégalité de Gibbs (issue de la propriété de concavité du logarithme) qui assure que la divergence de Kullback-Leibler est toujours une quantité positive : l'entropie de Shannon d'une distribution $\mathbf{p}$ est toujours inférieure ou égale à l'entropie croisée de $\mathbf{p}$ et $\mathbf{q}$ (i.e. $-\sum_i p_i \log(q_i)$), avec égalité si et seulement si $\mathbf{p} = \mathbf{q}$. Non bornée, la divergence de Kullback-Leibler donne l'opposée de la log-vraisemblance moyenne associée à l'observation d'une quantité infinie de données distribuées selon $\mathbf{p}$ mais générées par le modèle induit par $\mathbf{q}$ [Shlens, 2007].

Il faut donc faire attention lorsque l'on compare entropie de Shannon et entropie de Shannon *relative*, car l'une et l'autre évoluent dans des sens opposés :

– plus un site est conservé, plus il est informatif et plus son entropie de Shannon est faible,
– plus la distribution observée diffère de la distribution attendue a priori, et plus son entropie relative est élevée.

On attend donc d'un site « informatif » qu'il ait à la fois une entropie de Shannon faible et une entropie relative élevée. Cette dernière mesure s'apparente à une mesure de l'informativité, ou encore de l'utilité d'un site en termes de modélisation.

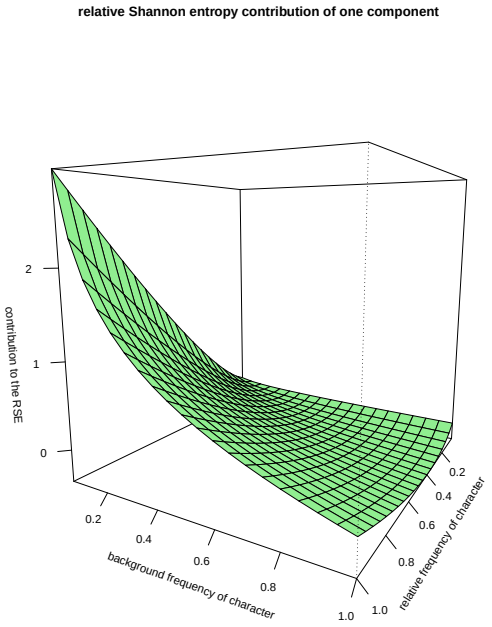


Figure 3.13. *Contribution individuelle d'une composante p_i dans l'entropie relative de Shannon*

composition	$\frac{1}{20}$ (unif.)	$\frac{1}{3}/\frac{1}{3}/\frac{1}{3}$	$\frac{1}{2}/\frac{1}{2}$	$\frac{9}{10}/\frac{1}{10}$
I L [V]	0,043	0,504	0,625	0,814
W C [H]	0,043	1,028	1,222	1,340
Shannon	1	0,367	0,231	0,109

Figure 3.14. *Quelques valeurs d'entropie relative de Shannon pour différentes fréquences observées. Un acide aminé entre crochets ne vaut que pour la colonne des « tiers ».*

On peut voir en figure 3.13 et en figure 3.14 comment l'entropie relative de Shannon mesure efficacement la divergence entre les fréquences constatées et les fréquences de fond, ou a priori.

Entropie de Von Neumann

Si l'entropie relative de Shannon prend bien en compte la distribution a priori, elle ne fait en revanche aucun cas des échangeabilités plus ou moins grandes entre acides aminés. Or, si l'on sait que la leucine (L) est par ses caractéristiques physico-chimiques très proche de l'isoleucine (I), et donc plutôt susceptible de remplacer celle-ci par l'action d'un mécanisme évolutif, alors on comprend qu'une colonne composée pour moitié d'isoleucines et pour moitié de leucines présente un signal de conservation plus important qu'une colonne comprenant moitié de tryptophanes (W) et moitié de prolines (P), très différents l'un de l'autre.

La mesure d'entropie de Von Neumann était à l'origine un outil développé pour la physique quantique. L'état d'un système quantique étant en fait un mélange d'états dits « purs », il peut être décrit par une *matrice de densité*. L'entropie de Von Neumann est une mesure sur cette matrice, qui dénote le niveau de mélange de l'état quantique : plus cette entropie est faible, plus le système quantique se rapproche d'un état pur (par exemple, si l'entropie d'un système décrivant de la lumière non polarisée est élevée, en revanche celle du système correspondant à de la lumière polarisée est nulle).

L'entropie de Von Neumann d'un système décrit par une matrice de densité Ω sur un espace d'états de cardinal N s'écrit :

$$h_{\mathrm{VN}}(c) = -\operatorname{Tr}(\Omega \log_N(\Omega)) \tag{3.19}$$

où Tr est l'opérateur matriciel de trace et $\log_N$ désigne le logarithme matriciel ($A = \log B$ ssi. $B = e^A$).

Transposée dans le domaine de la biologie moléculaire par Caffrey et al. [Caffrey *et al.*, 2004] et considérée par rapport à l'entropie de Shannon, la mesure d'entropie de Von Neumann introduit en plus la notion de similarité entre caractères par la prise en compte d'une matrice S de substitution. Les éléments S_{ij} d'une telle matrice représentent le rapport entre la probabilité de voir les caractères d'indices i et j alignés l'un à l'autre, et le produit des probabilités *a priori* de les voir l'un et l'autre isolément. On appelle ce rapport « odds ratio » (cf. chapitre 2). La matrice de densité de la physique quantique est ici à une normalisation près le produit du vecteur des fréquences relatives observées par la matrice de substitution S. Si l'on considère un site c donné, on a :

$$\Omega = \mu \Pi S \tag{3.20}$$

où $\Pi = \operatorname{diag}\left(P^{(c)}\right)$ est la matrice carrée diagonale dont les éléments sont les fréquences relatives observées sur le site c, et μ un facteur de normalisation tel que $\operatorname{Tr}(\mu \Pi S) = 1$.

L'introduction de la matrice de similarité S permet de donner par exemple une entropie plus faible à un site composé essentiellement d'acides aminés aliphatiques qu'à un site composé d'acides aminés aux propriétés physico-chimiques contrastées. S peut être directement une matrice de substitution PAM ou BLOSUM, ou encore dériver d'une matrice de substitution e^{Qt} elle-même calculée à partir d'un modèle Q et d'un temps d'évolution t (cf. chapitre 4).

L'idée essentielle de l'entropie de Von Neumann appliquée aux états que sont les colonnes d'un alignement est donc de baser la mesure d'entropie non plus sur les comptages bruts issus de l'observation directe et immédiate du site, mais plutôt sur une estimation de ce que pourrait être ce site après un certain laps de temps d'évolution. On tient compte ainsi des échangeabilités ou taux de substitution constatés entre acides aminés, lesquels sont très différents d'une paire d'acides aminés à l'autre.

L'équation (3.19) faisant intervenir deux opérateurs d'algèbre linéaire (la trace d'une matrice est la somme de ses éléments diagonaux et le logarithme d'une matrice est l'opérateur inverse de l'exponentielle) se calcule plus aisément après diagonalisation de S et extraction de ses valeurs propres, puisque la trace d'une matrice est invariante par similitude.

Ainsi on obtient :

$$h_{\mathrm{VN}}(c) = -\sum_{i=1}^{n} \lambda_i \log_N(\lambda_i) \tag{3.21}$$

où les λ_i sont les valeurs propres de $\Omega = \mu\Pi S$.

On montre dans la table 3.1 l'influence du choix de S dans le calcul de l'entropie de Von Neumann, ainsi que la différence fondamentale entre entropies de Shannon et de Von Neumann.

	Shannon	Von Neumann		
		BLOSUM50	LG ; t=0,7	LG ; t=0,3
site `ILMV`	0,463	0,359	0,425	0,456
site `CQWY`	0,463	0,412	0,457	0,462

Table 3.1. *Comparaison de différentes mesures d'entropie sur un site cohérent au niveau physico-chimique et sur un autre qui ne l'est pas.*

Entropie relative de Von Neumann

Johansson et Toh ont publié en 2010 un article intéressant [Johansson et Toh, 2010] dans lequel ils démontrent que l'entropie de Von Neumann telle que dérivée par Caffrey et al. souffrait d'un défaut de conception : la matrice Ω que ces derniers proposent n'est pas symétrique, donc non auto-adjointe. De plus, ils mettent en œuvre une nouvelle mesure d'entropie, l'entropie relative de Von Neumann, laquelle combine enfin les avantages d'une approche à la Von Neumann (prise en compte des échangeabilités) et ceux d'une approche relative (prise en compte de la distribution a priori).

L'entropie relative de Von Neumann s'écrit :

$$\mathrm{RVNE}(P,\Pi) = \mathrm{Tr}(P\log P) - \mathrm{Tr}(P\log\Pi) \tag{3.22}$$

où P et Π sont deux matrices *symétriques* définies respectivement à partir du vecteur des observations $\vec{p}$ et de celui des fréquences correspondant à la distribution de fond, $\vec{\pi}$. En utilisant une matrice symétrique A décrivant les similarités entre acides aminés (par exemple une matrice du type BLOSUM), Johansson et Toh définissent les matrices P et Π de la façon suivante :

$$\begin{array}{rclcrcl} P & = & P'AP', & & P' & = & \mathrm{diag}(\sqrt{p_1},\ldots,\sqrt{p_n}), \\ \Pi & = & \Pi' A \Pi', & & \Pi' & = & \mathrm{diag}(\sqrt{\pi_1},\ldots,\sqrt{\pi_n}), \end{array}$$

Dans [Johansson et Toh, 2010], la matrice A est de plus définie via une normalisation de BLOSUM62 qui induit une diagonale uniquement composée de '1', et par là deux matrices P et Π de trace égale à 1, ce qui en fait des matrices densité valides. Johansson et Toh montrent l'efficacité de la mesure d'entropie relative de Von Neumann pour prédire les sites catalytiques au sein d'un alignement, des sites qui montrent systématiquement des distributions bien différentes de la distribution de fond des acides aminés.

Un point que nous n'avons pas abordé ici concerne la prise en compte des gaps. Une première idée est de pénaliser systématiquement les sites contenant des gaps en réduisant drastiquement leur *score informationnel*[3]. On peut par exemple appliquer un coefficient multiplicateur égal au ratio de caractères non gaps dans le site considéré. Le problème de cette mesure est qu'elle pénalise trop sévèrement des sites qui présentent un profil en acides aminés assez bien conservé, mais seulement sur une partie des séquences. Alors que cette information peut être significative, le coefficient multiplicateur a tendance à la noyer sous le bruit. Lorsque nous avons tenté de fabriquer des HMM profils en sélectionnant les états Match uniquement sur la base des entropies relatives de Von Neumann des

3. Pour obtenir une mesure de l'informativité d'un site qui soit croissante lorsque le site est conservé, il est commode d'introduire la notion de « score informationnel », ce dernier étant simplement égal à $1-H$ lorsque H est la mesure d'entropie considérée.

sites coefficientées par un tel ratio, nous n'avons pas obtenu de résultats significativement bons (cf. partie *Résultats*).

Globalement, on peut dire que la sélection des sites d'intérêt par des mesures basées sur leur entropie informationnelle sert de nombreuses méthodes de construction de modèles (notamment la détermination d'arbres phylogénétiques ou la construction de HMM profils) qui se montrent plus efficaces lorsqu'elles travaillent à partir d'alignements composés le plus possible de sites sans gaps ou avec peu de gaps, et présentant un signal informationnel fort. Les HMM profils, eux, « écrèment » indirectement de façon systématique les zones contenant un grand nombre de gaps, puisque ces dernières se trouvent « modélisées » par des états d'Insertion qui doivent beaucoup plus aux priors qu'aux observations (cf. section 3.2). Nous tentons dans cette thèse de nous affranchir de ces méthodes basées sur l'entropie pour utiliser pleinement l'information phylogénétique dans la désignation des sites d'intérêt, lors de la construction d'un HMM profil destiné à des recherches dans un voisinage phylogénétique donné.

CHAPITRE

4 Processus évolutifs et phylogénies

Sommaire

Le naturaliste anglais Charles Robert Darwin (1809–1882) est universellement célébré pour avoir été le découvreur de la « théorie de l'évolution », selon laquelle les espèces et individus évoluent au cours du temps, et que l'accumulation des générations s'accompagne de mutations souvent irréversibles, de créations de nouvelles espèces et d'extinctions : le vivant a une *histoire*, les espèces vivantes et les caractères morphologiques ne sont pas immuables. De nos jours, peu de scientifiques s'opposent à cette théorie [Meyer, 2004], souvent avec une démarche argumentative qui ne correspond pas à ce qu'on entend par « méthode scientifique ». Le cadre de la théorie de l'évolution fournit des outils puissants pour modéliser le vivant selon une approche historique ou « verticale » qui s'applique tout autant à l'échelle macroscopique des caractères morphologiques qu'à celle, microscopique, des séquences biologiques. Ce dernier point intéresse tout particulièrement notre propos, puisque nous tentons dans cette thèse de nous appuyer sur des représentations quantitatives de l'histoire évolutive des séquences pour construire des modèles de prédiction des séquences ancestrales, utilisant donc à la fois l'information verticale (contraintes

évolutives) et longitudinale (contraintes séquentielles ou structurelles).

Nous décrivons dans ce chapitre les outils permettant de modéliser l'évolution des séquences biologiques. Après avoir présenté l'idée d'évolution biologique, nous exposons les méthodes de construction d'arbre par maximum de vraisemblance, en particulier l'algorithme de pruning de Felsenstein. C'est en effet une adaptation de cet algorithme qui nous permet de calculer les paramètres des modèles que nous exposons dans cette thèse. Nous présentons enfin la méthode standard de prise en compte de la variabilité des vitesses d'évolution des sites, un outil important pour l'inférence phylogénétique.

4.1 La révolution darwinienne

Charles Darwin est connu avant tout pour son ouvrage publié en 1859 et intitulé *L'origine des espèces par le moyen de la sélection naturelle, ou la préservation des races favorisées dans la lutte pour la vie*. Peu avant lui, Jean-Baptiste de Lamarck (1744–1829) avait déjà eu l'intuition de la nécessité d'une théorie de l'évolution des espèces, car il avait compris que les êtres vivants possèdent une capacité d'adaptation à leur milieu, sans pouvoir expliquer les mécanismes sous-jacents à cette adaptation. Ces derniers seront affirmés et progressivement mis au jour par Darwin. On comprendra plus tard qu'au niveau des gènes, les mécanismes en question se produisent dans le cadre des lois d'hybridation définies par le père de la génétique et contemporain de Charles Darwin, Johann Gregor Mendel (1822–1884).

Darwin présente dans son ouvrage *L'origine des espèces* une construction théorique qui n'a cessé d'être validée depuis lors. On peut la décrire brièvement en quelques points :

1. tous les êtres vivants sont différents les uns des autres, chacun ayant des caractéristiques spécifiques (propres à son espèce) et individuelles (propres à chaque individu),
2. chez tous les êtres vivants se produit sous une forme ou sous une autre une lutte pour la survie (*struggle for life*). Cette lutte est illustrée par la prédation, l'exposition à des conditions climatiques défavorables ou catastrophiques, la concurrence sur des ressources limitées, etc,
3. seuls s'en sortent les individus les mieux *adaptés* aux conditions dans lesquelles ils vivent (*survival of the fittest*),
4. les mutations accumulées dans le patrimoine (qu'on appellera plus tard *génétique*) d'un individu sont transmises à sa descendance,

5. il est statistiquement plus probable pour un individu *adapté* de parvenir à se reproduire et donc à transmettre son patrimoine, que pour un individu moins adapté (mécanisme de *sélection naturelle* des mutations *positives*),
6. l'accumulation d'un nombre important de mutations aboutit à la création de nouvelles espèces.

Ce « programme darwinien », toujours parfaitement valable pour la communauté scientifique au vingt-et-unième siècle, a inspiré à son auteur une petite illustration très explicite que nous reproduisons en figure 4.1.

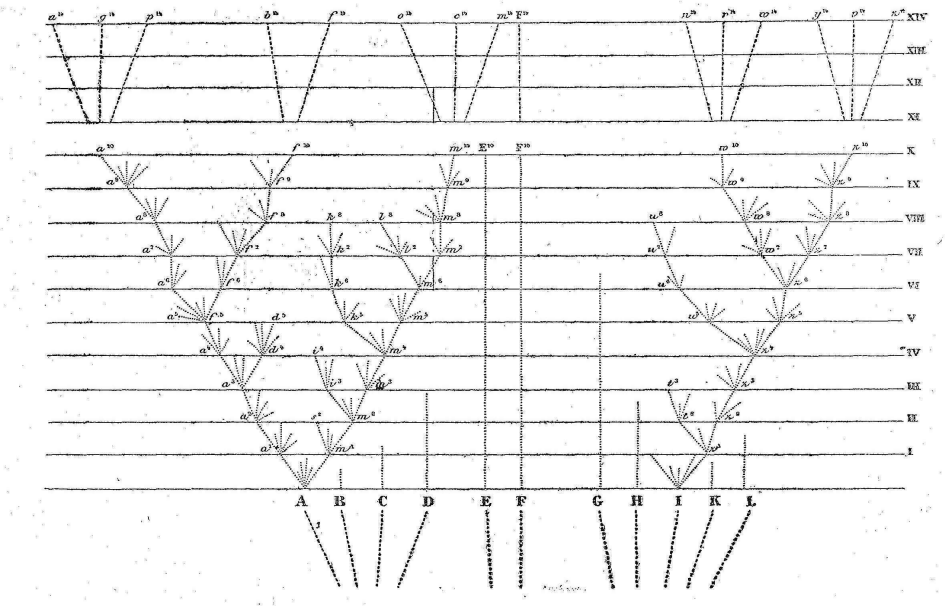


Figure 4.1. *Le concept de phylogénie représenté par Darwin dans son livre* L'origine des espèces. *Darwin illustrait ainsi l'idée selon laquelle des espèces disparaissent au cours de l'histoire (extinctions) tandis que d'autres apparaissent (spéciations) à partir d'une accumulation progressive de mutations. Les espèces présentes descendent donc nécessairement d'espèces passées.*

Cette figure représente ce qu'on appelle aujourd'hui un *arbre phylogénétique*, ou encore une *phylogénie* : les feuilles de l'arbre (situées en haut de la figure 4.1 représentent les taxa [1] observables actuellement. Certaines branches ne parviennent pas jusqu'en haut

1. Un *taxon* est un ensemble d'individus (en général une espèce, un genre, un ordre, etc) susceptible d'apparaître comme tel dans une classification, parce que tous les individus de cet ensemble possèdent les mêmes caractéristiques taxonomiques (e.g. symétrie bilatérale du corps, nombre de membres, système respiratoire, etc.)

de la figure : elles correspondent à des espèces qui sont aujourd'hui éteintes. Les bifurcations dans l'arbre sont autant d'événements de *spéciation*, c'est-à-dire d'apparition de deux espèces distinctes à partir d'une seule et même espèce. On s'accorde en général aujourd'hui à définir l'espèce comme décrivant un groupe maximal d'individus vivants possédant la capacité de s'interféconder en donnant une progéniture viable et elle-même féconde (définition due à Ernst Mayr, 1942). Deux espèces apparaissent donc lorsque des sous-groupes perdent progressivement la propriété d'interfécondité, parce qu'au sein de chacun de ces sous-groupes se sont accumulées des caractéristiques génétiques incompatibles (par exemple à la suite d'une séparation géographique des sous-groupes). L'arbre proposé par Darwin est orienté : en bas se trouve « l'origine de la vie » et en haut les espèces observables actuellement. Les lignes horizontales en pointillés correspondent à des moments de croissance soudaine de la biodiversité, l'apparition de nombreuses espèces nouvelles se faisant effectivement parfois de façon assez localisée dans le temps (cf. l'explosion cambrienne, autour de 540 millions d'années de cela, qui a donne la plupart des embranchements actuels d'animaux pluricellulaires).

Si nous disions plus haut que cette vision proposée par Charles Darwin n'a cessé d'être validée depuis sa publication, c'est parce que les scientifiques ont pu *constuire* de telles phylogénies sur la base de modèles probabilistes, lesquelles phylogénies ont été validées par des données expérimentales (fossiles, données géologiques, etc). Comment fait-on aujourd'hui pour concevoir des phylogénies à partir de la seule connaissance des séquences actuelles, c'est ce que nous allons brièvement examiner dans ce qui suit.

4.2 Deux approches de la phylogénie : maximum de parcimonie et maximum de vraisemblance

Deux grands types d'approche sous-tendent la construction d'arbres phylogénétiques à partir des séquences observées : les approches parcimonieuses et les approches basées sur la vraisemblance.

La parcimonie défend l'hypothèse selon laquelle l'histoire évolutive la plus probable est celle qui peut s'exposer à l'aide d'un *nombre minimal d'événements de substitution*. Ainsi, dans le cadre de cette approche, lorsqu'on observe l'arbre représenté en figure 4.2, on infère *un seul* événement de substitution L↔M sur la branche marquée d'un astérisque. C'est l'algorithme de Fitch [Fitch, 1971] qui permet de calculer efficacement le score de parcimonie d'un arbre et, partant, de rechercher la meilleure topologie possible pour un jeu de séquences donné, c'est-à-dire celle qui minimise le score de parcimonie de l'arbre (différentes techniques classiques d'optimisation sont applicables à ce problème, mais

elles sortent du cadre de notre exposé).

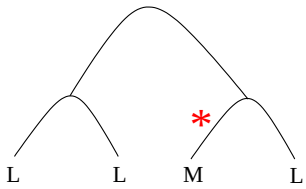


Figure 4.2. *Un arbre à quatre taxa dans lequel l'approche de maximum de parcimonie infère un seul événement évolutif, ignorant les éventuelles mutations silencieuses. L'histoire évolutive expliquant cet arbre avec le moins d'événements possible fait en effet apparaître un événement L↔M sur la branche marquée d'un astérisque rouge. Toutes les autres histoires possibles s'appuient au minimum sur deux événements évolutifs.*

Les choses ne sont probablement pas aussi simples dans la réalité où, sur des branches relativement longues, on peut avoir une série de mutations successives, parfois silencieuses (par exemple $I \rightarrow L \rightarrow I$). De telles séries d'événements sont ignorées par la parcimonie, qui méconnaît également (au moins dans sa version initiale) les échangeabilités différentes entre acides aminés. L'approche de maximum de vraisemblance, elle, s'appuie fortement sur un modèle de substitution Q pour attribuer un score à une phylogénie qui est fonction de la capacité du modèle à engendrer les séquences observées. La *vraisemblance* d'un arbre phylogénétique $\mathscr{T}$ est alors la probabilité que les données observées (Obs) aient été engendrées par le « travail » du processus de substitution Q le long des branches de $\mathscr{T}$: $\mathrm{Lk}(\mathscr{T}) = \Pr(\mathrm{Obs}|\mathscr{T}, Q)$.

Dans la suite nous nous concentrons sur l'exposition des approches basées sur la vraisemblance et des processus markoviens qui les sous-tendent. Ce sont en effet celles-ci, réputées plus pertinentes [Yang, 1996; Zhang et Nei, 1997; Guindon et Gascuel, 2003], que nous utilisons dans cette thèse.

4.3 Des modèles de substitution pour quantifier l'évolution

Que se passe-t-il le long des branches d'un arbre phylogénétique ? C'est pour répondre à cette question qu'ont été inventés les processus de substitution. Pour décrire ces derniers, nous reprenons quatre hypothèses communément adoptées :

1. on peut modéliser l'évolution des séquences site par site, en supposant ceux-ci indépendants les uns des autres,
2. l'évolution des séquences se fait sans mémoire : le devenir d'un acide aminé au cours de l'évolution peut se modéliser sans connaissance de son passé, simplement à par-

tir de l'état courant. C'est l'hypothèse *markovienne*, du nom du mathématicien russe Andreï Markov (1856–1922),

3. le processus de substitution est *homogène*, c'est-à-dire que ses caractéristiques ne changent pas au cours du temps,
4. le processus de substitution est *stationnaire*, c'est-à-dire qu'il admet une unique distribution horizon : la probabilité qu'un acide aminé a se retrouve transformé en un acide aminé b au bout d'un temps infini ne dépend pas de a mais uniquement de b. Elle est égale à une valeur qu'on notera π_b. On suppose par ailleurs que l'on se trouve *effectivement* dans l'état stationnaire, c'est-à-dire que les fréquences des différents caractères dans les séquences correspondent à la distribution π.

La première hypothèse, bien que reconnue comme étant fausse, est une hypothèse « de commodité » rendant les modèles conçus utilisables pour calculer des phylogénies en un temps raisonnable sur les machines que nous utilisons actuellement. En ignorant les possibilités de délétion et d'insertion (respectivement la disparition de caractères à un moment donné de la phylogénie et leur apparition spontanée), nous devons donc travailler avec des probabilités de *substitution* de la forme $\Pr(a \xrightarrow{t} b)$. Cette expression représente la probabilité qu'un caractère a soit remplacé par un caractère b au cours de l'évolution sur un temps t.

En considérant l'évolution d'un caractère x sur une durée t positive, la propriété markovienne s'écrit :

$$\Pr\left(x(s+t) = b | x(s) = a, \{x(s')\}_{s'<s}\right) = \Pr\left(x(s+t) = b | x(s) = a\right) \tag{4.1}$$

Si $\mathscr{A}$ est l'ensemble des valeurs possibles pour les caractères (par exemple l'ensemble des vingt acides aminés du vivant), la propriété d'homogénéité s'écrit quant à elle :

$$\forall (s_1, s_2, t) \in \mathbb{R}_+^3 \, , \forall (a,b) \in \mathscr{A}^2 \quad \Pr\left(x(s_1+t) = b | x(s_1) = a\right) = \Pr\left(x(s_2+t) = b | x(s_2) = a\right) \tag{4.2}$$

Enfin, la stationnarité du processus induit :

$$\forall (a,b) \in \mathscr{A}^2 \quad \lim_{t\to\infty} \Pr\left(a \xrightarrow{t} b\right) = \pi_b \tag{4.3}$$

Les trois conditions (4.1), (4.2) et (4.3) définissent ce qu'on appelle une chaîne (ou processus) de Markov à temps continu, homogène et stationnaire. Un tel processus est entièrement décrit par une matrice carrée Q de dimension $n \times n$ (où n est la taille de l'alphabet $\mathscr{A}$) et à coefficients dans $\mathbb{R}$. Q respecte plusieurs propriétés découlant des conditions décrites ci-dessus, en particulier on a $\pi Q = \pi$ (π est le vecteur des probabilités stationnaires, appelées aussi « probabilités de fond » ou « de background » si l'on cède à l'anglicisme).

La combinaison de la propriété de Markov et de la propriété d'homogénéité fait que les probabilités de substitution ne dépendent ni du passé, ni de l'endroit où l'on se trouve sur la droite du temps. On écrit donc, sans tenir compte de la date s : $\Pr(x(s+t) = b | x(s) = a) = \Pr(b|a,t) = \Pr\left(a \xrightarrow{t} b\right)$. L'intérêt de la matrice Q est de donner une expression littérale simple à ces probabilités de substitution :

$$\forall t \in \mathbb{R}_+ , \forall (a,b) \in \mathscr{A}^2 \quad \Pr\left(a \xrightarrow{t} b\right) = e^{Qt} \tag{4.4}$$

Il « suffit » donc de calculer l'exponentielle de la matrice Qt (matrice composée des éléments de Q multipliés chacun par t) pour obtenir d'un coup toutes les probabilités de substitution décrivant de façon probabiliste l'évolution biologique sur une durée t. Ces probabilités de substitution seront décrites par la matrice carrée $P_t = e^{Qt}$. Les outils mathématiques permettant de diagonaliser Q sont ici précieux, sans quoi le calcul de l'exponentielle serait malaisé.

Le cadre théorique étant fixé, il nous reste à établir Q, la matrice décrivant le processus de Markov. On appelle souvent cette matrice « matrice des taux instantanés », en effet pour des durées t proches de 0, on a le comportement à la limite $P_{t\to 0} = I + Qt$ (où I est la matrice identité). En ce qui concerne les substitutions entre acides aminés, de telles matrices Q ont été construites sur la base d'analyses de grands nombres d'alignements. Parmi les plus populaires, on peut citer dans l'ordre chronologique d'apparition les matrices PAM [Dayhoff *et al.*, 1978], JTT [Jones *et al.*, 1992], WAG [Whelan et Goldman, 2001] et plus récemment LG [Le et Gascuel, 2008]. Toutes ces matrices décrivent les substitutions observables entre acides aminés en général, sans être biaisées pour tel ou tel type particulier de séquences.

Contrainte de réversibilité

Certains modèles sont dits temps-réversibles, c'est-à-dire qu'ils n'imposent pas une orientation particulière pour les branches d'un arbre phylogénétique. Dans ce cas, si l'on imagine une branche de longueur l portant à une extrémité le caractère a et à l'autre le caractère b, on doit avoir « la même » probabilité pour les deux histoires évolutives $a \xrightarrow{l} b$ et $b \xrightarrow{l} a$, selon que l'on considère la branche dans un sens ou dans l'autre. C'est-à-dire : $\forall (a,b) \in \mathscr{A}^2$, $\pi_a \Pr(a \xrightarrow{l} b) = \pi_b \Pr(b \xrightarrow{l} a)$. Les matrices de substitution respectant cette contrainte sont dites *réversibles*. C'est le cas pour la plupart. Dès lors, on peut écrire : $\forall (a,b) \in \mathscr{A}^2$, $\pi_a q_{ab} = \pi_b q_{ba}$, le terme q_{ij} correspondant à l'élément de la matrice Q situé sur la i^{e} ligne et la j^{e} colonne. On a donc $\frac{q_{ab}}{\pi_b} = \frac{q_{ba}}{\pi_a} = R_{a\leftrightarrow b}$. On choisit d'appeler ce dernier terme *échangeabilité* entre les acides aminés a et b, ce qui permet de représenter la matrice de substitution comme une demi-matrice (puisque par construction $R_{a\leftrightarrow b} = R_{b\leftrightarrow a}$),

accompagnée de la distribution d'équilibre π.

4.4 Algorithme de Felsenstein

L'algorithme dit « algorithme de pruning de Felsenstein » [Felsenstein, 1973, 1981] permet de calculer la vraisemblance d'un modèle phylogénétique à partir de données observées, c'est-à-dire la probabilité avec laquelle les données observées ont pu être engendrées par le modèle en question. Nous présentons ci-dessous les fondements théoriques de cet algorithme.

4.4.1 Présentation dans un contexte raciné

Soit $\mathscr{T}$ un arbre binaire enraciné avec m feuilles. Soit D l'ensemble des données aux feuilles et Q le processus substitutionnel. La *vraisemblance* de l'arbre s'écrit :

$$\mathrm{Lk}(Q,\mathscr{T}|D) \overset{\text{déf.}}{=} \Pr(D|\mathscr{T},Q) \tag{4.5}$$

Quelle est alors la probabilité des données sachant l'arbre et le processus substitutionnel ? Soit $D = \{X_i\}_{i\in[1,n]}$ où n est la longueur de l'alignement et X_i la i^{e} colonne de l'alignement sur lequel on a construit la phylogénie. On numérote les feuilles de 1 à m et X_i^j se trouve donc être l'acide aminé présent chez le taxon j (la j^{e} feuille) et sur la colonne i. On fait de plus l'hypothèse classique d'indépendance des sites :

$$\Pr(D|\mathscr{T},Q) = \prod_{i\in[1,n]} \Pr(X_i|\mathscr{T},Q) \tag{4.6}$$

Comment calculer $\Pr(X_i|\mathscr{T},Q)$? On présente ci-dessous un algorithme récursif à partir de la situation présentée en figure 4.3.

L'arbre $\mathscr{U}$ est constitué du nœud u et de ses sous-arbres gauche ($\mathscr{V}$) et droit ($\mathscr{W}$). Les nœuds v et w sont les racines respectives des sous-arbres $\mathscr{V}$ et $\mathscr{W}$. Les branches reliant u à v et u à w sont de longueurs respectives l_v et l_w. Quelle est alors la vraisemblance de l'arbre $\mathscr{U}$ sachant les données aux feuilles de $\mathscr{U}$, c'est-à-dire de $\mathscr{V}$ et de $\mathscr{W}$? On peut répondre à cette question en réécrivant $\mathrm{Lk}(Q,\mathscr{U}|D_\mathscr{V} \cup D_\mathscr{W})$. On paramétrise selon la valeur (inconnue) du caractère porté par le nœud u en utilisant simplement le théorème de Bayes :

$$\begin{aligned} \mathrm{Lk}(Q,\mathscr{U}|D_\mathscr{V} \cup D_\mathscr{W}) &\overset{\text{déf.}}{=} \Pr(D_\mathscr{V} \cup D_\mathscr{W}|\mathscr{U},Q) && (4.7)\\ &= \sum_\alpha \Pr(u=\alpha)\Pr(D_\mathscr{V} \cup D_\mathscr{W}|\mathscr{U},Q,u=\alpha) && (4.8) \end{aligned}$$

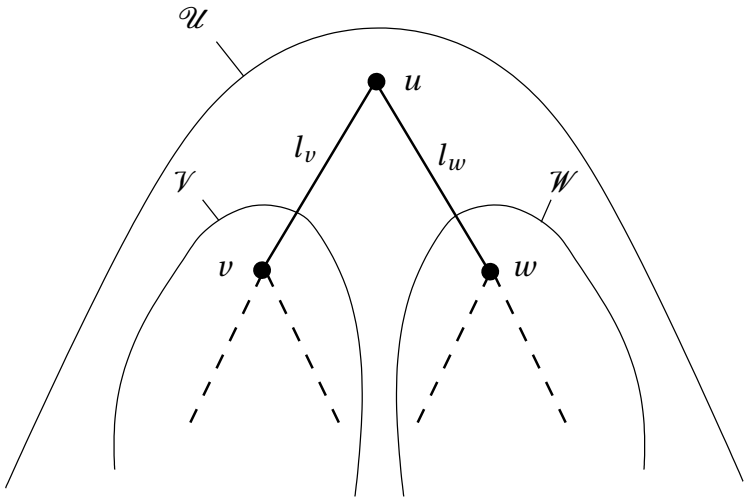


Figure 4.3. *Un arbre phylogénétique de racine u et ses deux sous-arbres. Cette illustration sert de support à la présentation de l'algorithme de Felsenstein que l'on trouve dans ces lignes.*

Dans l'équation qui précède, la sommation sur α se fait en parcourant les vingt acides aminés. L'histoire évolutive ayant découlé de la présence du caractère α au nœud u s'écrit ensuite en descendant dans l'arbre et en paramétrisant suivant les caractères portés par v et w :

$$\Pr(D_{\mathscr{V}} \cup D_{\mathscr{W}} | \mathscr{U}, Q, u = \alpha) = \sum_{\beta} \left[\Pr(\alpha \xrightarrow{l_v} \beta) \Pr(D_{\mathscr{V}} | \mathscr{V}, Q, v = \beta) \right] \sum_{\gamma} \left[\Pr(\alpha \xrightarrow{l_w} \gamma) \Pr(D_{\mathscr{W}} | \mathscr{W}, Q, w = \gamma) \right] \tag{4.9}$$

En écrivant ce qui précède, on fait l'hypothèse logique que l'histoire découlant du nœud v (resp. w) ne dépend que du sous-arbre $\mathscr{V}$ (resp. $\mathscr{W}$), du processus Q et du caractère porté par v (resp. w). Ceci est cohérent avec la structure d'arbre elle-même, qui présente l'évolution de l'une et de l'autre de deux espèces issues d'un événement de spéciation, comme *indépendantes* après ledit événement.

Si l'on sait calculer les expressions de la forme $\Pr(\alpha \xrightarrow{l_v} \beta)$, alors on a atteint un schéma de *récurrence* puisque le calcul de $\Pr(D_{\mathscr{U}} | \mathscr{U}, Q, u = \alpha)$ a engendré deux expressions de la même forme, faisant intervenir les sous-arbres $\mathscr{V}$ et $\mathscr{W}$ à la place de l'arbre $\mathscr{U}$. Reste à savoir comment cette récurrence va prendre fin : lorsque le sous-arbre $\mathscr{V}$ est réduit à une simple feuille ($\mathscr{V} = \{v\}$), que vaut $\Pr(D_{\mathscr{V}} | \{v = \beta\}, Q)$? La réponse à cette question est simple :

puisqu'on *connaît* le caractère x porté par la feuille v dans l'alignement ($D_{\mathcal{V}} = \{x\}$), on a :

$$\Pr(D_{\mathcal{V}}|\{v=\beta\},Q) = \begin{cases} 1 & \text{si } \beta = x \\ 0 & \text{sinon} \end{cases} \tag{4.10}$$

Les feuilles portant un gap (ou un caractère 'X' ou '?', qui dénotent tous deux l'indétermination totale quant au caractère effectivement présent dans la séquence biologique) ont une vraisemblance partielle égale à 1 quel que soit le caractère testé : $\Pr(D_{\mathcal{V}}|\{v=\beta\},Q) = 1\ \forall\beta$.

En se rappelant de plus que les processus substitutionnels markoviens Q donnent l'expression de $\Pr(\alpha \xrightarrow{l_v} \beta) = \left[e^{Ql_v}\right]_{\alpha,\beta}$, on sait donc calculer $\Pr(D_{\mathcal{U}}|\mathcal{U},Q,u=\alpha)$ pour tout acide aminé α. Mais pour calculer finalement la vraisemblance de l'arbre $\mathcal{U}$ selon l'équation (4.8), il nous faut encore avoir connaissance des termes $\Pr(u=\alpha)$, probabilités a priori d'avoir le caractère α à la racine de l'arbre. Ces probabilités a priori sont données par la distribution stationnaire π du processus Q :

$$\mathrm{Lk}(Q,\mathcal{U}|D_{\mathcal{U}}) = \sum_{\alpha} \pi(\alpha)\Pr(D_{\mathcal{U}}|\mathcal{U},Q,u=\alpha) \tag{4.11}$$

On souligne ici l'importance de la condition de stationnarité, laquelle n'est pas toujours faite (voir par exemple [Galtier et Gouy, 1998], repris par [Boussau *et al.*, 2008]).

L'algorithme récursif détaillé ci-dessus, dit algorithme de *pruning* de Felsenstein [Felsenstein, 1973, 1981], nous permet donc de calculer la vraisemblance d'un arbre enraciné étant donné les caractères aux feuilles et le processus substitutionnel employé, et ce pour chacun des sites d'un alignement. De par l'hypothèse d'indépendance des sites, la vraisemblance calculée sur l'alignement est le produit des vraisemblances calculées sur les sites. Les vraisemblances étant toujours comprises entre 0 et 1 (et en règle général très proches de zéro pour des alignements de données biologiques), il est d'usage de manipuler non pas les vraisemblances elle-mêmes, mais le logarithme de ces vraisemblances. Le produit sur les n colonnes de l'alignement se transforme alors en une sommation :

$$\mathrm{Lk}(Q,\mathcal{T}|D) = \prod_{i=1}^{n} \mathrm{Lk}(Q,\mathcal{T}|X_i) \quad \Rightarrow \quad \mathrm{logLk}(Q,\mathcal{T}|D) = \sum_{i=1}^{n} \mathrm{logLk}(Q,\mathcal{T}|X_i) \tag{4.12}$$

4.4.2 Algorithme dans un contexte non raciné

Nous présentons maintenant l'algorithme précédent dans un contexte *non raciné*, c'est-à-dire dans lequel l'arbre ne possède pas de racine identifiée. Chaque nœud n'a donc plus ni père ni fils, mais que des voisins (un seul s'il s'agit d'une feuille, trois sinon). Ce contexte correspond à l'arbre phylogénétique $\mathcal{U}$ tel que présenté en figure 4.4, où

l'on a singularisé une branche b_0 de longueur l reliant deux nœuds que l'on appelle v et w. Comme précédemment, on a deux sous-arbres $\mathcal{V}$ et $\mathcal{W}$. On peut dire que $\mathcal{V}$ est le sous-arbre gauche de la branche en question, tandis que $\mathcal{W}$ en est le sous-arbre droit. L'un comme l'autre peuvent éventuellement être réduit à une feuille, cela ne gêne pas notre exposé.

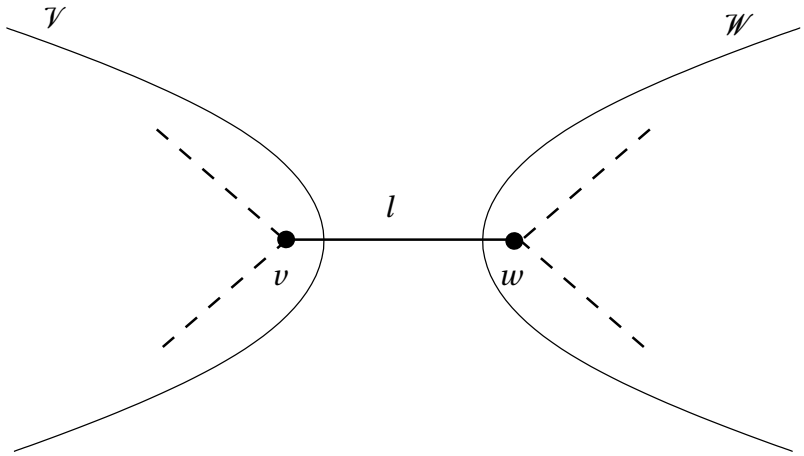


Figure 4.4. *Un arbre phylogénétique $\mathcal{U}$ non raciné, représenté en singularisant l'une de ses branches, qu'on appelle b_0.*

Le calcul de la vraisemblance totale de l'arbre se fait en choisissant implicitement l'un des deux sommets de cette branche (ici nous choisissons v) et en brisant la symétrie des expressions conformément à ce choix :

$$\mathrm{Lk}(Q,\mathcal{U}|D_{\mathcal{V}}\cup D_{\mathcal{W}}) = \sum_{\alpha}\sum_{\beta}\pi(\alpha)\Pr(\alpha \xrightarrow{l} \beta)\,\mathrm{Lk}(Q,\mathcal{V}|D_{\mathcal{V}},v=\alpha)\,\mathrm{Lk}(Q,\mathcal{W}|D_{\mathcal{W}},w=\beta) \quad (4.13)$$

En remarquant que *du point de vue de la branche* b_0, chacun des nœuds situés à ses deux extrémités (à gauche et à droite) apporte un vecteur de vraisemblances partielles à vingt composantes correspondant aux vingt acides aminés, on peut écrire ce qui précède de la façon suivante :

$$\mathrm{Lk}(Q,\mathcal{U}|D_{\mathcal{V}}\cup D_{\mathcal{W}}) = \sum_{\alpha}\sum_{\beta}\mathrm{Lk\_left}(b_0,\alpha)\pi(\alpha)\Pr(\alpha \xrightarrow{l} \beta)\,\mathrm{Lk\_right}(b_0,\beta) \quad (4.14)$$

L'utilisation d'un processus Q réversible rend le calcul de la vraisemblance $\mathrm{Lk}(\mathcal{U}|D_{\mathcal{V}}\cup D_{\mathcal{W}},Q)$ insensible au choix d'une racine implicite, puisqu'on a alors : $\forall l\geq 0,\ \forall(\alpha,\beta)\in\mathcal{A}^2,\ \pi(\alpha)\Pr(\alpha\xrightarrow{l}\beta)=\pi(\beta)\Pr(\beta\xrightarrow{l}\alpha)$.

4.5 Rendre compte de la variabilité des taux d'évolution en fonction des sites : la loi Gamma

Il est courant d'observer dans les jeux de données des vitesses globales d'évolution qui ne sont manifestement pas les mêmes pour tous les sites d'un alignement. Par exemple, étant donné que les substitutions de nucléotides situés en première ou deuxième position de codon induisent bien plus souvent que les mutations en troisième position une modification de l'acide aminé issu de la traduction, on observe une plus grande variabilité des troisièmes positions de codons. D'autre part, certains sites au sein d'un alignement revêtent une importance fonctionnelle particulière, et sont donc globalement moins sujets à mutations (voir par exemple [Fitch et Margoliash, 1967]).

La prise en compte de ces phénomènes nécessite de pouvoir calculer des vraisemblances de modèles phylogénétiques en permettant que le processus de substitution soit plus ou moins « rapide » en fonction des sites de l'alignement. Un site « rapide » connaît statistiquement plus de mutations qu'un site « lent » observé pendant le même laps de temps. De manière équivalente, on peut ou bien appliquer séparément pour chaque site un facteur multiplicatif à la matrice représentant le modèle de substitution, ou bien multiplier de façon indépendante pour chaque site toutes les longueurs de branche de la phylogénie par ce même facteur. Pour un site dont on estimerait à r la vitesse globale d'évolution, les probabilités de substitution sur une branche de longueur l seraient données par la matrice e^{rlQ}. Mais comment déterminer la vitesse globale d'évolution d'un site donné ? Ou plutôt, sans qu'on connaisse précisément la vitesse de tel ou tel site, que peut-on dire de la distribution statistique des taux d'évolution ?

La première tentative aboutie pour répondre à cette question est venue de [Nei et Gojobori, 1986; Jin et Nei, 1990], des publications dans lesquelles les auteurs utilisent notamment la loi Gamma pour modéliser la variabilité des taux d'évolution dans un contexte de phylogénies établies sur des distances entre séquences. Suite à un travail mené en thèse, Ziheng Yang proposa en 1993 une version de cette modélisation dans le cadre de l'établissement de phylogénies par la méthode de maximum de vraisemblance [Yang, 1993]. La loi Gamma est une loi de probabilités acceptant deux paramètres α et β, et définie sur $\mathbb{R}_+^*$ par la densité $g_{\alpha,\beta}$ ci-dessous :

$$g_{\alpha,\beta}(r) = \frac{\beta^\alpha e^{-\beta r}}{\Gamma(\alpha)} r^{\alpha-1} \quad , \tag{4.15}$$

où $\Gamma(x)$ est la fonction spéciale eulérienne représentant la prolongation classique de la factorielle à $\mathbb{C}$. La densité $g_{\alpha,\beta}$ définie ci-dessus donne une distribution dont l'espérance est $\frac{\alpha}{\beta}$ et la variance $\frac{\alpha}{\beta^2}$. α est le paramètre de *forme* de la distribution, tandis que β est appelé paramètre *d'échelle*. Pour ne travailler qu'avec des distributions dont l'espérance mathématique vaut 1, et donc pour conserver un ensemble cohérent entre processus de

substitution et longueurs de branche d'un arbre phylogénétique (définis de telle façon qu'une branche de longueur k corresponde à une espérance de k substitutions par site), on choisit en phylogénie de s'en tenir à des distributions Gamma d'espérance 1, ce qui donne la contrainte $\alpha = \beta$ [Yang, 1993]. Le seul paramètre reste donc le paramètre de forme, un α proche de zéro dénotant une forte variabilité de type exponentiel entre les sites tandis que lorsque $\alpha \to \infty$, on tend vers un même taux d'évolution pour tous les sites, c'est-à-dire l'absence de loi Gamma.

Lorsqu'on adopte une loi Gamma de paramètre α, et si l'on convient que $r\mathscr{U}$ désigne l'arbre $\mathscr{U}$ dont on a multiplié toutes les longueurs de branche par r, la vraisemblance d'un site donné (caractères aux feuilles $D_{\mathscr{U}}$) se calcule en intégrant sur les valeurs (positives) du taux d'évolution r :

$$\mathrm{Lk}(\mathscr{U}|D_{\mathscr{U}},Q) = \int_{r=0}^{\infty} \mathrm{Lk}(r\mathscr{U}|D_{\mathscr{U}},Q)\, g_\alpha(r)\, dr \tag{4.16}$$

Alors que [Yang, 1993] pointait clairement le coût computationnel bien trop élevé engendré par le calcul de l'expression (4.16) dans le cadre de processus d'évolution markoviens (lorsque les phylogénies sont construites à partir de distances entre séquences, on sait résoudre cette intégrale), il proposa en 1994 une version discrétisée de la distribution Gamma [Yang, 1994]. L'idée est d'approcher la distribution continue $g_\alpha(r)$ par une série de K *classes* de vitesse formant autant d'intervalles $]r_k^-, r_k^+]$ partitionnant $\mathbb{R}_+^*$. Le choix d'un représentant r_k par classe (en général la moyenne ou la médiane d'une classe) permet d'effectuer l'approximation classique en théorie de l'intégration des fonctions continues :

$$\int_0^{\infty} f(r) g_\alpha(r) dr \;=\; \sum_{k=1}^{K} \int_{r_k^-}^{r_k^+} f(r) g_\alpha(r) dr \;\simeq\; \sum_{k=1}^{K} \Pr(k) f(r_k) \tag{4.17}$$

où l'on calcule les probabilités de classe à partir de la densité de la loi Gamma :

$$\forall k \in [1,K] \quad \Pr(k) = \int_{r_k^-}^{r_k^+} g_\alpha(r) dr \quad . \tag{4.18}$$

Dans [Yang, 1994] (et partout par la suite), on utilise des intervalles équiprobables ($\forall k \in [1,K]\ \Pr(k) = 1/K$) et l'intégration est faite sur l'espérance au sein de chaque intervalle. D'autres solutions existent mais sont rarement utilisées.

En faisant jouer le rôle de $f(r)$ à la vraisemblance $\mathrm{Lk}(r\mathscr{U}|D_{\mathscr{U}},Q)$, on obtient finalement l'approximation de Yang telle qu'utilisée en phylogénie moléculaire :

$$\mathrm{Lk}(\mathscr{U}|D_{\mathscr{U}},Q) \;\simeq\; \sum_{k=1}^{K} \Pr(k)\, \mathrm{Lk}(r_k\mathscr{U}|D_{\mathscr{U}},Q) \tag{4.19}$$

Il est communément admis [Yang, 1994] que la modélisation par plus de quatre classes de vitesse apporte peu de gains. On adopte donc souvent $K = 4$, là où utiliser par exemple 8

classes de vitesse doublerait la complexité algorithmique et l'espace mémoire nécessaire. C'est ce choix $K = 4$ que nous faisons pour tous les calculs de vraisemblance figurant dans cette thèse.

CHAPITRE

5 Combiner descriptions séquentielle et évolutive : les phylo-HMM

Sommaire

Sous la dénomination commune de « phylo-HMM » se cache en réalité toute une série de modèles apparentés plus ou moins proches les uns des autres, développés par différents auteurs dans la décennie 1990 et dans celle qui a suivi. Ces modèles visent à combiner deux types d'analyse s'appliquant à toute famille de séquences apparentées : d'une part, une analyse tendant à décrire les séquences d'un point de vue longitudinal (comme des assemblages linéaires de positions structurellement et fonctionnellement déterminées) et, d'autre part, une analyse visant à identifier les liens évolutifs reliant les unes aux autres les séquences d'une famille (envisageant celles-ci comme issues d'une seule et même séquence ancestrale).

Cette volonté holistique de combiner information structurelle et information évolutive est légitime et scientifiquement fondée : les séquences que l'on observe actuellement sont à la fois déterminées par une histoire évolutive (modélisée par un arbre phylogénétique) et par des contraintes structurelles ou fonctionnelles (sites actifs, sites de liaison, repliements) sans lesquelles la séquence perdrait toute fonctionnalité. Le couplage difficile à modéliser mais néanmoins fort entre contraintes évolutives et contraintes structurelles est pleinement à l'œuvre dans les séquences protéiques. La dualité entre modèles évolutifs et modèles structurels fait de la recherche de procédés permettant de construire simultanément l'alignement correct et la phylogénie supportant les séquences, le Saint

Graal de la phylogénie moléculaire et de la génomique comparative. À l'heure actuelle, la communauté ne dispose pas d'outils capables de réaliser simultanément alignement et phylogénie de manière cohérente et valide. Les phylo-HMM s'inscrivent dans l'abondante littérature qui aborde ce sujet.

Le paysage composé par les différents travaux relevant de la problématique des phylo-HMM étant très divers, on peut essayer d'en établir une vision synthétique en abordant ceux-ci sous l'angle de leur objectif avoué. Se dégagent alors trois buts principaux qui dessinent les frontières de ce paysage :

1. Dès le début des années 1990, Thorne, Kishino et Felsenstein [Thorne *et al.*, 1991, 1992] ont tenté de définir un modèle d'évolution des séquences sous la forme d'un processus de Markov à temps continu baptisé « links ». Les auteurs modélisaient pour la première fois non seulement les substitutions entre caractères, mais également l'apparition d'insertions ou de délétions. Leur objectif était de fournir une base théorique satisfaisante aux procédés d'alignement de séquences, en utilisant le modèle « links » pour construire des alignements *pairwise*. On peut donc parler pour ce premier objectif d'**alignement guidé par l'évolution**. Ce modèle a ensuite été repris par d'autres auteurs une dizaine d'années plus tard [Holmes et Bruno, 2001; Holmes, 2003]. Parallèlement à cela, Mitchison et Durbin ont introduit en 1995 et 1999 des modèles appelés « tree-HMM » dont le but était également de permettre de préciser progressivement, par un procédé itératif s'appuyant sur la phylogénie reliant les séquences entre elles, l'alignement multiple de ces dernières [Mitchison et Durbin, 1995; Mitchison, 1999].
2. À la suite des travaux de Mitchison et Durbin, Qian et Goldstein ont repris le concept de « tree-HMM » en clarifiant certains aspects (notamment en ce qui concerne le traitement phylogénétique des transitions) mais surtout en changeant d'objectif par rapport à leur prédécesseur : le calcul automatique d'alignements multiples corrects à partir de la phylogénie étant un but difficilement atteignable, Bin Qian et Richard Goldstein ont préféré utiliser ces modèles pour démontrer leur capacité à **rechercher plus efficacement des homologues à une famille de séquences donnée en entrée**.
3. Le vocable « phylo-HMM » lui-même a enfin été introduit par Adam Siepel et David Haussler [Siepel et Haussler, 2004a, 2005], lesquels ont donné la définition théorique d'un HMM dont chacun des nœuds produirait non plus un caractère issu d'un alphabet, mais *tout un site*, c'est-à-dire une colonne d'un alignement multiple. Le modèle de génération de ces sites est décrit par les auteurs à partir d'un arbre phylogénétique (cf. chapitre 4). Les exemples d'application donnés par Siepel et Haussler mettent en jeu des phylo-HMM de taille restreinte (typiquement de deux à six états) dans lesquels à chaque état du HMM correspond une phylogénie avec une vitesse d'évolution associée, censée par exemple modéliser l'évolution de sites correspondant à une certaine position de codon. Cet exemple précis avait déjà été publié par Felsen-

stein et Churchill en 1996 [Felsenstein et Churchill, 1996]. Cette troisième classe de modèles se propose donc de **réaliser l'annotation des séquences grâce à une modélisation fine des sites s'appuyant sur la phylogénie**.

Dans ce chapitre, nous allons examiner les différents modèles signalés ci-dessus, en mettant l'accent sur les travaux de Mitchison et Durbin et sur ceux de Qian et Goldstein leur ayant fait suite, car ce sont ces modèles qui ont largement inspiré notre approche. Nous abondonnons donc l'ordre chronologique pour décrire les modèles selon les buts délimités ci-dessus.

5.1 L'objectif de l'alignement guidé par la phylogénie

5.1.1 Autour du modèle *links* de Thorne, Kishino et Felsenstein

Thorne, Kishino et Felsenstein ont publié en 1991 un modèle [Thorne *et al.*, 1991, 1992] permettant d'aligner deux séquences nucléotidiques l'une contre l'autre en tenant compte de la distance évolutive les séparant. Le modèle en question se compose de deux sous-modèles qui fonctionnent de manière complémentaire mais dissociée. Le premier sous-modèle est un modèle classique d'évolution sur les nucléotides. Il permet de rendre compte des substitutions entre nucléotides en affectant à chacune une certaine probabilité, fonction de la distance t séparant les séquences et d'une matrice de substitution Q (cf chapitre sur les processus de substitution). La nouveauté concerne le second sous-modèle, qui permet de modéliser la façon dont insertions et délétions aparaissent dans l'alignement. Ce deuxième sous-modèle rend donc compte des gaps. Il se présente comme une alternative au modèle classique de coût affine induit par une pénalité d'ouverture de gap et une pénalité d'extension de gap.

Le modèle statistique d'évolution des séquences proposé par Thorne, Kishino et Felsenstein fonctionne comme suit : on considère la séquence ancestrale comme étant formée de « liens », chaque lien symbolisant une position dans la séquence. À tout instant, un lien peut donner naissance à un autre lien qui par convention s'intercalera immédiatement à la droite de son père dans la séquence. De telles naissances se produisent avec un taux λ. Un lien peut également mourir, avec un taux μ. Avec un tel modèle, pour que la longueur d'une séquence ne tende pas vers l'infini lorsque $t \to \infty$, il faut nécessairement avoir $\lambda < \mu$. De plus, pour que la longueur des séquences ne tende pas vers 0 pour une évolution infinie, il faut rajouter un terme dit « d'immigration » ou de « naissance spontanée » : avec le même taux λ, de nouveaux liens peuvent être créés par un lien fictif et immortel placé au début de la séquence.

Ce modèle dit « links » a été repris par Holmes et Bruno [Holmes et Bruno, 2001; Holmes, 2003] qui l'ont étendu et rendu opératoire pour calculer des alignements mul-

tiples à partir d'un arbre phylogénétique reliant des séquences non alignées. L'idée des auteurs est de considérer chaque branche de l'arbre phylogénétique comme le support d'un alignement de deux séquences, les deux séquences se trouvant chacune à une extrémité de la branche. Cette branche définit un temps t qui permet de calculer selon le modèle TKF91 brièvement exposé ci-dessus, l'alignement des deux séquences l'une contre l'autre au maximum de vraisemblance. Holmes et Bruno décrivent leur logiciel HANDEL, qui réalise l'alignement multiple à partir des séquences non alignées et de l'arbre phylogénétique, en inférant les séquences ancestrales et en raffinant les alignements deux-à-deux en plusieurs passes. Mais testé sur la base d'alignements multiples BAliBASE 2.01 [Thompson *et al.*, 1999a], HANDEL fait assez nettement moins bien (de 13% à 19% moins bien) que le programme d'alignement CLUSTALW [Thompson *et al.*, 1994a], dont plusieurs études récentes ont pourtant démontré qu'il est nettement sous-optimal (lire par exemple [Thompson *et al.*, 1999b, 2011]).

HANDEL et son modèle *links* sont des tentatives pour construire des alignements de séquences avec un modèle d'évolution sous-jacent et pas seulement une matrice de substitution et des pénalités. À cet égard, on peut dire qu'il s'agit là d'une construction théorique intéressante, mais qui s'est révélée relativement peu efficace d'un point de vue opératoire sur des données biologiques réelles.

Enfin, il est intéressant de noter que dans [Holmes et Bruno, 2001], les auteurs se penchent sur la notion de résidus « alignés » l'un avec l'autre. La chose paraît pourtant simple : on considère traditionnellement que deux caractères sont alignés l'un avec l'autre dès lors qu'ils figurent sur la même colonne dans un alignement multiple. Or on sait que dans les alignements produits par les outils actuels, bien souvent dans les zones de faible confiance de l'alignement (c'est-à-dire en général les zones fortement gapées) deux résidus peuvent se retrouver placés sur la même colonne de façon plus ou moins arbitraire. Holmes et Bruno font observer, eux, qu'ils ne considèrent deux résidus comme étant alignés l'un avec l'autre que lorsqu'ils se trouvent sur la même colonne *et* qu'aucune des séquences intermédiaires inférées sur les nœuds de l'arbre se trouvant sur le chemin reliant les deux séquences d'intérêt ne porte un gap sur cette même colonne. La raison sous-tendant cette restriction est qu'on ne peut légitimement considérer comme évolutivement liés un caractère ancestral ayant fait l'objet d'une délétion et un caractère contemporain issu d'une insertion ultérieure dans la séquence, fortuitement située à l'endroit de la délétion première. Cette volonté de s'en tenir à des alignements qui respectent les événements phylogénétiques supposés avoir survenu le long des branches de l'arbre se retrouve développée dans les travaux d'Ari Löytynoja et Nick Goldman autour du développement de PRANK [Löytynoja et Goldman, 2005, 2008].

5.1.2 Mitchison & Durbin : HMM et matrices de substitution basées sur des arbres

Dans un article publié en 1995 dans la revue *Journal of Molecular Evolution*, Graeme Mitchison et Richard Durbin [Mitchison et Durbin, 1995] ont exposé une nouvelle classe d'outils pour la modélisation d'alignements de séquences biologiques en tenant compte d'un arbre phylogénétique reliant les séquences entre elles. En réalité, le but avoué de leur démarche était de concevoir des arbres phylogénétiques à partir d'une approche de maximum de vraisemblance (ce qui consiste à maximiser la probabilité des données observées sachant le modèle). Depuis les travaux de Joe Felsenstein en 1981 et la mise au point des matrices PAM et BLOSUM dans la décennie 1980 et au début des années 1990, la communauté disposait déjà au moins théoriquement des outils permettant de concevoir de tels arbres phylogénétiques, mais à la condition de partir de données *sans gaps*. Or il était évident pour tous que dès lors que l'on souhaitait établir des alignements pour des séquences réelles et non plus seulement pour des portions bien alignées (cf. Henikoff et Henikoff et leur base de données BLOCKS [Henikoff et Henikoff, 1992]), il était nécessaire de prendre en compte la possibilité d'aligner certains caractères contre des trous, ou « gaps ». Selon que les caractères alignés au sein des zones de gaps sont en sus ou non de la séquence de référence pour l'alignement, on parle respectivement d'« insertions » et de « délétions ».

La problématique intéressant Mitchison et Durbin était donc de pouvoir concevoir des arbres phylogénétiques *qui modélisent explicitement insertions et délétions*. Jusqu'alors on ne travaillait qu'avec des arbres phylogénétiques de nucléotides ou d'acides aminés, c'est-à-dire des arbres dont les feuilles portent des *caractères* et sur les branches desquels agissent des processus de substitution sur lesdits caractères. Mitchison et Durbin ont introduit la notion d'arbres de transitions, tout en indiquant une méthodologie de résolution pour le problème de l'apprentissage des matrices de substitutions de transitions, constitutives du processus agissant le long des branches de tels arbres.

Processus de substitution affectant les transitions

Mitchison et Durbin partent des HMMs profils dont la structure a été décrite par Krogh et coauteurs en 1994 [Krogh *et al.*, 1994]. Il s'agit d'une enchaînement linéaire de nœuds comportant chacun trois états : un état match, un état d'insertion et un état de délétion. On représente en figure 5.1 un tel HMM profil de longueur 4. Les états match constituent l'ossature du HMM : on s'attend à ce qu'une séquence parfaitement modélisée par le HMM profil n'emprunte que les états match sans en manquer aucun, la séquence dite de consensus pour le HMM étant une telle séquence qui en plus émet sur chaque état match le caractère le plus probable). Les états d'insertion permettent comme leur nom l'indique

d'insérer des caractères entre deux états match sans les aligner contre la séquence consensus, tandis que les états de délétion permettent à une séquence de ne pas exprimer un ou plusieurs des caractères présents dans le consensus. Un tel HMM profil présente 9 transitions sortantes par nœud.

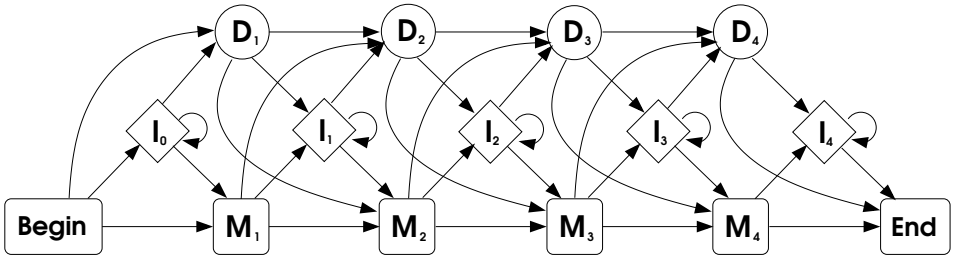


Figure 5.1. *Un HMM profil de structure classique (Krogh et. al, 1994) de longueur 4.*

Mitchison et Durbin font remarquer que le problème de l'alignement d'une séquence contre un modèle de type HMM profil peut se résoudre en considérant le chemin (chemin dans l'automate d'états finis que constitue le HMM, partant de l'état muet initial *Begin* et arrivant à l'état muel final *End*) qui maximise le produit des probabilités de transition et d'émission de caractères rencontrées le long de ce chemin. C'est la stratégie dite « de Viterbi ». Tout modèle de type HMM profil étant lui-même construit à partir d'un alignement multiple, la correspondance entre caractères de l'alignement et états du HMM profil est immédiate pour ce qui concerne les séquences d'apprentissage : on construit un HMM profil à partir d'un alignement en sélectionnant dans ce dernier un certain nombre de colonnes, chacun des états match du HMM profil correspondant à l'une de ces colonnes. Un acide aminé présent sur une colonne « match » correspond à la traversée d'un état match (`M`), un acide aminé présent sur une colonne qui *n'est pas* marquée « match » correspond à la traversée d'un état d'insertion (`I`), tandis qu'un gap présent sur une colonne marquée « match » correspond à la traversée d'un état de délétion (`D`). Si l'on considère l'alignement long de 7 positions présenté en figure 5.2, et si l'on décide d'utiliser comme positions « match » les positions n° 1, 3, 6 et 7, alors on aboutit au HMM présenté en figure 5.3.

On peut décrire le chemin parcouru par les séquences dans le HMM profil pour atteindre leur meilleur score [1] comme une suite de transitions entre états cachés du HMM (`M`, `I` et `D`). C'est ce qui est représenté dans la figure 5.4.

L'*idée fondamentale* des tree-HMMs de Mitchison et Durbin consiste à affirmer qu'il existe un lien de type phylogénétique entre les transitions empruntées par les différentes

1. Du moins suppose-t-on que c'est bien le chemin escompté : ce serait une anomalie du processus d'apprentissage que l'une des séquences d'apprentissage reçoive *in fine* un meilleur score en empruntant dans le HMM profil un autre chemin que celui utilisé pour elle lors du processus même d'apprentissage.

	1	2	3	4	5	6	7
seq1	M	L	P	-	-	R	E
seq2	M	-	P	G	G	R	-
seq3	M	L	P	-	-	K	D
seq4	M	-	-	-	-	-	E

Figure 5.2. *Un exemple d'alignement multiple*

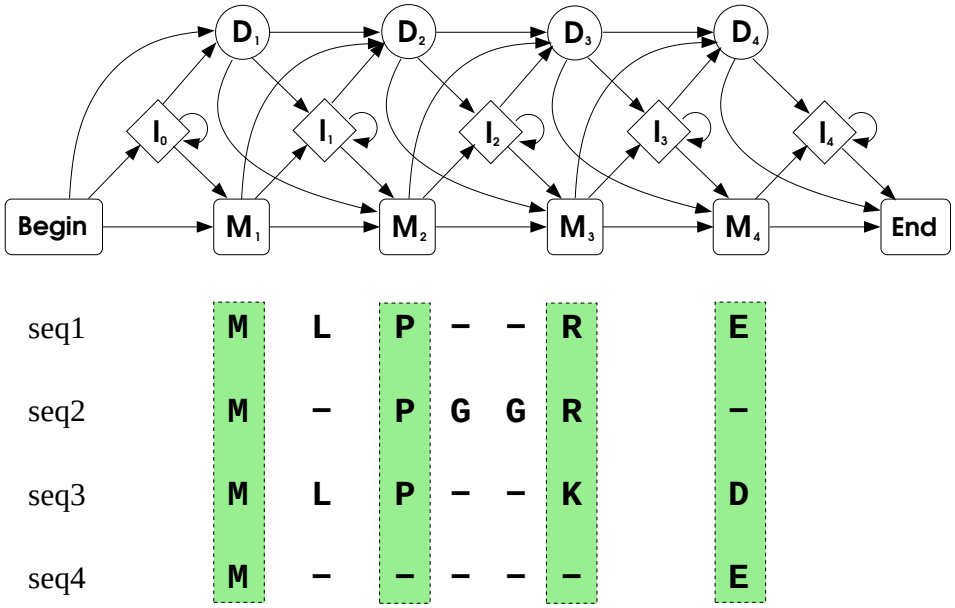


Figure 5.3. *Le HMM profil correspondant à l'alignement de la figure 5.2. Les colonnes ombrées de l'alignement forment l'ossature du HMM (colonnes « match »).*

séquences d'un alignement, tout comme il existe un lien phylogénétique entre les acides aminés émis sur une même colonne de cet alignement. Ce lien est logiquement représenté sous la forme d'un arbre dont les longueurs de branche sont en proportion de la quantité attendue d'événements de substitution entre un nœud de l'arbre et son père.

Deux types d'arbre distincts pour les transitions

D'emblée, Mitchison & Durbin introduisent une idée essentielle et jamais remise en cause par la suite dans la littérature, celle consistant à dissocier deux types de processus évolutifs :

– d'un côté, les transitions quittant un état match (désignées génériquement « M· »)

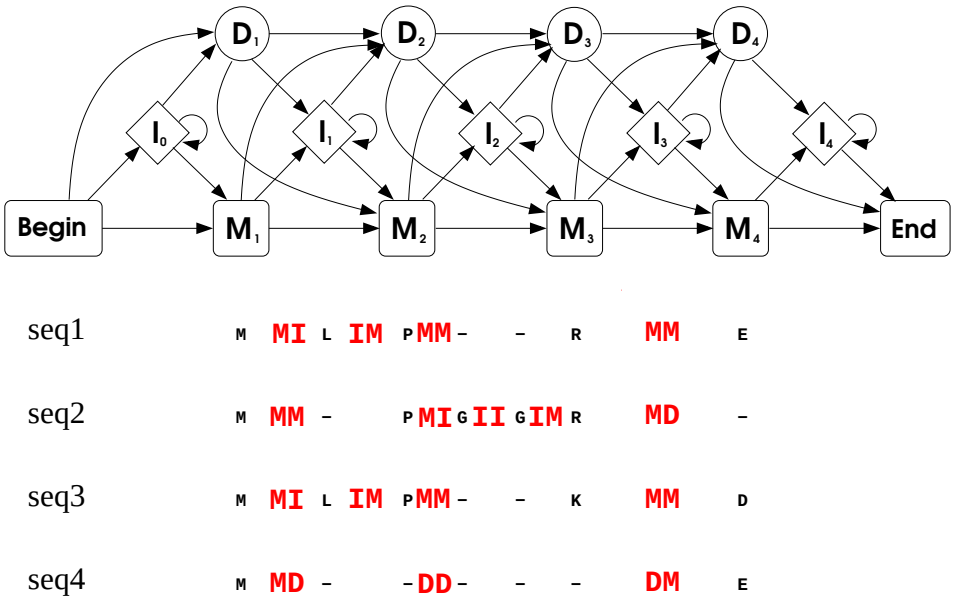


Figure 5.4. *Le HMM profil correspondant à l'alignement de la figure 5.2, avec les chemins empruntés par les séquences d'apprentissage (en rouge).*

sont associées à un processus de Markov propre agissant sur les branches d'un arbre appelé ici M-arbre,
– de l'autre, les transitions quittant un état de délétion (désignées génériquement « D· ») sont associées à un processus de Markov propre agissant sur les branches d'un arbre appelé ici D-arbre.

On a donc *deux* processus de Markov en temps continu pour modéliser les substitutions de transitions (cf. figure 5.5), les caractéristiques des transitions quittant un état d'insertion induisant un traitement particulier pour celles-ci.

Entre deux colonnes match consécutives dans un alignement de séquences, Mitchison & Durbin définissent ainsi deux arbres de transition, l'un pour les transitions M· et l'autre pour les transitions D·. Les feuilles de ces arbres ne sont pas toutes « actives » (c'est-à-dire qu'elles ne portent pas toutes une transition) : pour un site donné, une séquence passe soit par l'état M correspondant, soit par l'état D correspondant. Les deux arbres se « partagent » ainsi les séquences, une feuille active dans l'un étant nécessairement inactive dans l'autre et vice-versa (cf. figure 5.6).

Cette dissociation des processus de substitution affectant les transitions ne va pas nécessairement de soi. [Mitchison et Durbin, 1995] n'argumente pas pour expliquer ce choix, qui vient probablement de l'idée de « substitution élémentaire » qui, dans l'esprit des auteurs, s'accorde mal avec le fait qu'une transition MM, par exemple, évolue en une transition DD. Cette substitution nécessite effectivement à la fois le changement de l'état

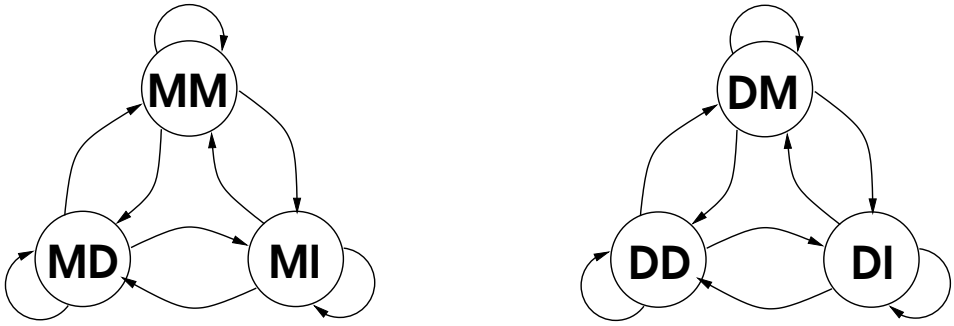


Figure 5.5. *Deux processus de Markov distincts et disjoints pour les substitutions entre transitions*

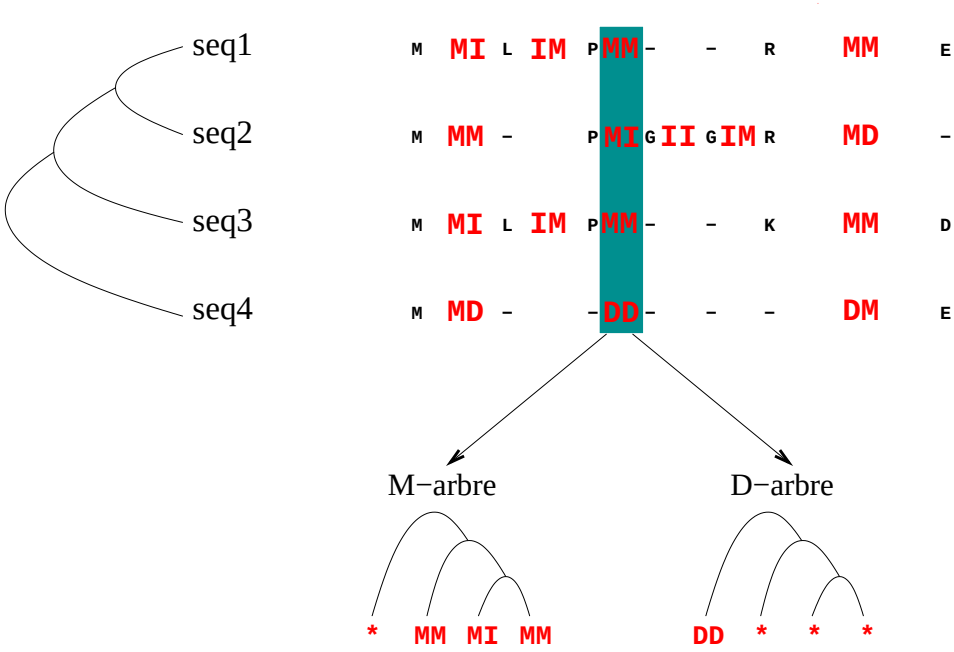


Figure 5.6. *Le couple d'arbres (M-arbre et D-arbre) induit une bipartition sur l'ensemble des feuilles et donc des séquences. Une feuille inactive pour un arbre prend la valeur « * », qui représente l'absence d'information.*

de départ et celui de l'état d'arrivée (M devenant D dans les deux cas). Or, un tel événement de substitution entre transitions peut très bien découler d'un événement évolutif considéré comme élémentaire. Considérons par exemple la situation présentée en figure 5.7. On a là une séquence ancestrale ARKKF qui subit la perte d'un petit bloc de trois acides aminés consécutifs pour donner la séquence contemporaine. Comme on le sait, cette situation est courante en biologie moléculaire, où les insertions et délétions touchent

souvent plusieurs résidus consécutifs à la fois. Si la séquence de référence pour le HMM profil utilisé sur la famille de séquences en question retient comme positions conservées (ou « positions match ») la totalité des positions exprimées par la séquence ancestrale, alors on a *de facto* une transition élémentaire de MM vers DD.

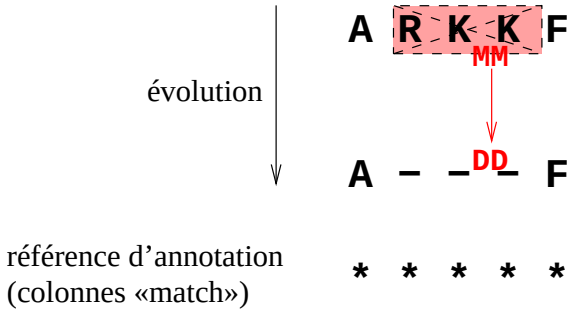


Figure 5.7. *Événement évolutif élémentaire induisant une substitution de MM à DD*

Au delà de cette limitation, on peut également remarquer que le cloisonnement des deux types de transition entraîne une perte d'information. Prenons l'exemple de deux sites consécutifs pour lesquels toutes les séquences d'apprentissage passent d'un état match à l'état match suivant. Les feuilles du M-arbre sont donc toutes peuplées de caractères MM, tandis que les feuilles du D-arbre sont toutes inactives. L'absence d'informations pour le D-arbre va entraîner qu'en tout point de la phylogénie associée la distribution a posteriori des caractères DM, DD et DI sera égale à la distribution a priori. Or, une colonne pleine de transitions MM indique une zone fortement conservée. La « bonne » probabilité a posteriori pour DM est donc probablement *supérieure* à la probabilité a priori : même si une délétion intervient sur le site précédent, statistiquement on aura de bonnes chances de retomber sur un état match pour le site suivant.

Apprentissage des matrices de substitution entre transitions

Pour apprendre leurs matrices de substitution entre M-transitions d'une part et entre D-transitions d'autre part, Mitchison et Durbin sont partis d'un jeu de données de petite taille regroupant des alignements issus de la base HSSP [Sander et Schneider, 1993] : des séquences de globines α et β, ainsi que des domaines variables d'immunoglobulines. Constatant une proportion importante de substitutions MD $\rightarrow$ MM et DD $\rightarrow$ DM, les auteurs ont fait le choix d'interpréter ces substitutions comme des événements d'*insertion* depuis une séquence ancestrale, plutôt que de perte d'une délétion ancestrale. Ils prennent appui sur cette interprétation possible pour contraindre leur processus d'apprentissage en annulant certaines des probabilités de transition dans les processus présentés en fi-

gure 5.5, pour aboutir aux processus tels que représentés en figure 5.8.

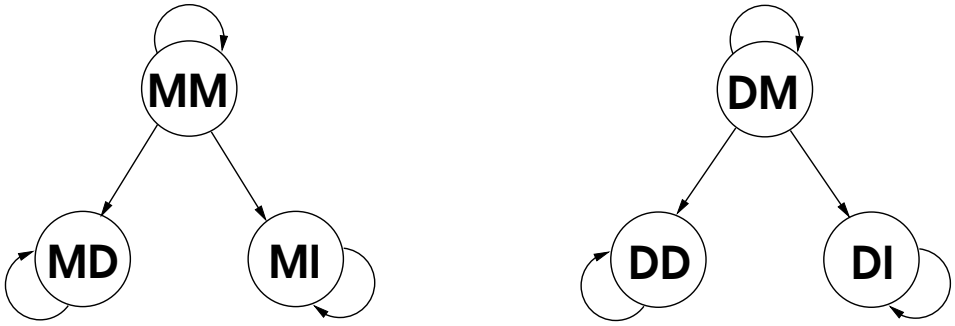


Figure 5.8. *Deux processus de Markov avec contraintes pour les substitutions entre transitions (M&D 1995)*

Ces nouveaux processus transforment les états MD, MI, DD et DI en états *puits* : ces transitions ne peuvent subir de substitution. Lorsqu'on parcourt l'arbre depuis la racine en direction des feuilles, si on rencontre un nœud interne décoré par l'une de ces quatre transitions, alors toutes les feuilles du sous-arbre issu de ce nœud le sont aussi. Les chaînes de Markov associées à ces processus ne sont donc plus ni stationnaires, ni réversibles, mais ce n'est pas le souci de Mitchison et Durbin. Leur approche consiste à concevoir pour chaque processus un jeu de matrices à la PAM [Dayhoff *et al.*, 1978], valable pour un intervalle de temps de divergence donné.

Le chemin de pensée emprunté là par Mitchison et Durbin est un peu surprenant : en constatant tout d'abord l'abondance des substitutions MD → MM et DD → DM dans leur jeu de données, ils justifient quelques lignes plus loin l'abandon pur et simple de telles substitutions dans les modèles qu'ils mettent en place ! Tentons de comprendre en détail ce qui est en jeu dans ces problématiques d'interprétation évolutive des substitutions entre transitions à partir d'un exemple simple. Nous partons d'un alignement constaté entre une séquence ancestrale `X-X` et une séquence fille `XXX`. Nous considérons que ces deux séquences sont reliées par une branche d'un arbre phylogénétique, le sens du processus évolutif étant clairement indiqué (figure 5.9). Nous montrons que l'annotation de cette histoire évolutive en termes de transitions dépend de la position des colonnes « match » du HMM (figures 5.10 et 5.11).

Avec strictement les mêmes données au départ, l'interprétation de la figure 5.10 donne la perte d'une délétion ancestrale, tandis que celle de la figure 5.11 fait l'hypothèse de l'introduction d'une délétion. La différence provient simplement du choix des colonnes fait pour établir l'ossature du HMM profil. Pour la figure 5.10, la séquence qui fait office de référence se trouve être la séquence fille, alors que dans la figure 5.11 c'est plutôt la

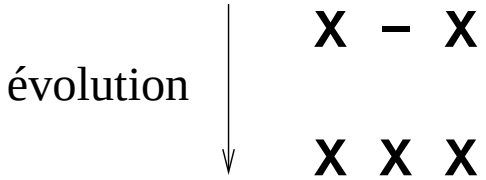


Figure 5.9. *Un arbre phylogénétique minimaliste reliant deux séquences. « X » représente un acide aminé quelconque, « - » un gap.*

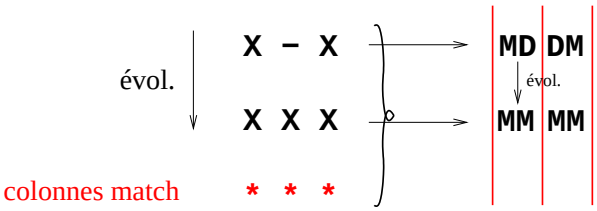


Figure 5.10. *Une interprétation possible de l'évolution des motifs de transition, les trois sites de l'alignement présenté en figure 5.9 étant retenus comme des colonnes « match » par le HMM profil en vigueur. Les colonnes match sont représentées en rouge par des astérisques sous les colonnes de l'alignement de séquences et par des traits verticaux au sein de l'alignement de transitions. Cet exemple induit une substitution MD→MM (perte d'une délétion ancestrale).*

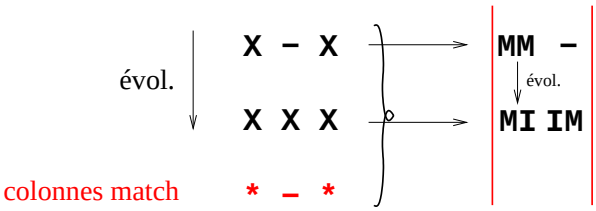


Figure 5.11. *Le même exemple que ci-contre, dans lequel on a modifié l'étiquetage en colonnes « match » pour ne retenir que le premier et le troisième site. Sur les mêmes données que ci-contre, on aboutit à une substitution MM→MI (introduction d'une insertion).*

séquence ancestrale qui fait office de référence. En se plaçant systématiquement dans le deuxième cas de figure, Mitchison et Durbin font l'hypothèse que les états match d'un HMM correspondent toujours de façon bijective aux positions exprimées dans la séquence ancestrale située à la racine de l'arbre. Or c'est loin d'être le cas : on sait que l'architecture des HMMs profils est en générale déterminée par une heuristique simple *à partir des séquences contemporaines*. Par exemple, HMMER3 fait le choix simple de désigner une colonne comme étant une colonne « match » si et seulement si elle possède plus de 50% de symboles (i.e. non gaps). Cette simple constatation devrait rendre plus courants les schémas correspondant à la figure 5.11 par rapport à ceux correspondant à la figure 5.10, battant en brèche la construction réalisée par [Mitchison et Durbin, 1995].

Plus important, on montre que des colonnes entières de l'alignement des transitions apparaissent ou disparaissent selon l'étiquetage de l'alignement de séquences en termes

de colonnes match. Ainsi dans les figures 5.12 et 5.13, la substitution DD→DM n'apparaît que dans une des deux « interprétations » (puisqu'il faut bien considérer, comme nous venons de le faire plus haut, que l'étiquetage d'un alignement de séquences en termes de colonnes match et non match est constitutif d'une interprétation de l'histoire phylogénétique en termes de transitions dans un HMM).

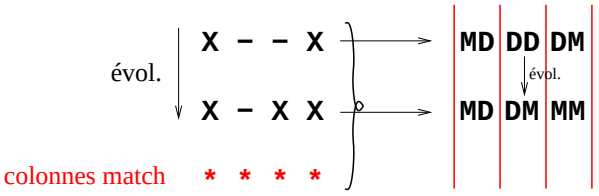


Figure 5.12. *Alignement de séquences avec annotation des colonnes match et motifs de transition correspondants. Comme précédemment, les colonnes match sont représentées en rouge par des astérisques sous l'alignement de séquences et par des traits verticaux au sein de l'alignement de transitions. Cet exemple présente une substitution DD→DM (raccourcissement d'une zone de délétion ancestrale).*

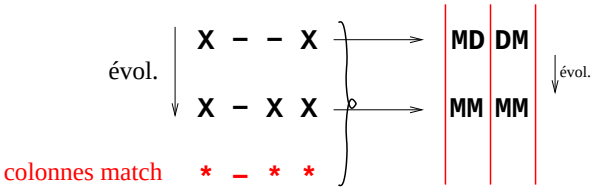


Figure 5.13. *Le même exemple que ci-contre, dans lequel on a modifié l'étiquetage en colonnes « match » pour ne retenir que le premier, le pénultième et le dernier site. La colonne correspondant à la substitution DD→DM a disparu.*

Traitement des insertions

Dans un HMM profil, les états d'insertion possèdent la particularité de boucler sur eux-mêmes (cf. figure 5.1). Ils peuvent donc modéliser plusieurs colonnes consécutives d'un alignement de séquences. De plus, le contenu des insertions est représenté par commodité de manière alignée dans un alignement de séquences (i.e. un acide aminé x d'une séquence se trouve au-dessus d'un autre acide aminé y d'une autre séquence), sans pour autant que l'on puisse discerner un lien évolutif au sein d'une même colonne. Les logiciels d'alignement progressif courants (par exemple ClustalW, Mafft ou Muscle) sont en quelque sorte responsables de ces représentations erronées, parce qu'ils produisent souvent des alignements non pertinents dans leurs régions peu denses (c'est-à-dire à forte proportion de gaps dans la majorité des séquences).

Résultats et conclusions de Mitchison et Durbin en 1995

Nous rappelons que l'objectif des auteurs était de trouver l'alignement multiple optimal sachant un arbre phylogénétique donné, et ensuite de se servir de cet alignement pour déterminer quel est le meilleur arbre (maximum de vraisemblance). Ce processus

itératif est logique étant donné les dépendances cycliques entre arbre phylogénétique et alignement de séquences.

Les auteurs indiquent que s'ils utilisent leur méthode pour raffiner un alignement en partant d'un modèle choisi aléatoirement, alors les alignements obtenus à la fin du processus itératif sont largement sous-optimaux, notamment lorsqu'on les compare à ceux que l'on obtient via un HMM classique. Il semble qu'il s'agisse d'un problème de maximum local, car lorsque l'alignement initial est déjà plus ou moins correct, la méthode itérative donne bien des alignements dont la vraisemblance est nettement accrue.

Mitchison et Durbin pointent le fait que l'alignement structurellement correct n'est pas forcément celui que choisira leur méthode basée sur la phylogénie. Ils mettent en évidence un exemple dans lequel le regroupement sur une même colonne de l'alignement d'insertions ou de délétions formant un clade dans la phylogénie (ce qui est donc justifié si l'on s'intéresse au placement phylogénétique des gaps) peut mener à des aberrations du point de vue structurel ou fonctionnel (si l'on considère en revanche les propriétés physico-chimiques des acides aminés alignés). Il y a dans cette remarque la constatation d'un hiatus parfois irréductible entre préoccupations d'ordre évolutif et préoccupations d'ordre fonctionnel, qui affectera aussi bien les modèles que nous décrivons dans cette thèse.

5.1.3 Mitchison, 1999

Faisant suite au travail publié avec Richard Durbin en 1995 [Mitchison et Durbin, 1995], Graeme Mitchison publie seul un nouveau travail sur les tree-HMM en 1999 [Mitchison, 1999]. L'objectif principal du papier est de présenter une méthode pour calculer simultanément alignement et phylogénie, comme c'était déjà le cas auparavant. Néanmoins ce travail présente des différences notables avec le précédent, lesquelles différences nous décrivons brièvement ci-dessous.

Dans [Mitchison, 1999], G. Mitchison se sépare totalement des états d'insertion existant dans [Mitchison et Durbin, 1995] et dans le travail originel de Krogh et Haussler [Krogh *et al.*, 1994]. Les tree-HMM sur lesquels il travaille désormais ont donc la structure présentée en figure 5.14. La raison de cet abandon des états Insertion réside dans les difficultés théoriques rencontrées avec ces états, déjà discutées ci-avant. Mitchison argue du fait que les insertions sont toujours possibles dans son nouveau modèle : elles existent dès lors qu'une séquence emprunte un état Match là ou sa séquence parente dans l'arbre phylogénétique emprunte un état de Délétion. Mais, sans autre innovation, la longueur d'une séquence à aligner contre le modèle resterait limitée par la longueur de ce dernier. C'est pourquoi Mitchison propose de contourner cette limitation en permettant lors de la phase d'alignement d'une séquence contre le modèle, l'allongement à l'envi de ce dernier

par l'insertion d'une colonne pleine de gaps. Une telle colonne correspond à un nouveau nœud { M, D } dans lequel toutes les séquences de l'alignement en cours empruntent l'état de Délétion. La possibilité d'effectuer ensuite une substitution $DD \rightarrow DM$ ou $MD \rightarrow MM$ juste avant cette nouvelle colonne de gaps permet donc de réaliser une insertion à cet endroit, *entre* deux nœuds qui étaient consécutifs dans le modèle d'origine.

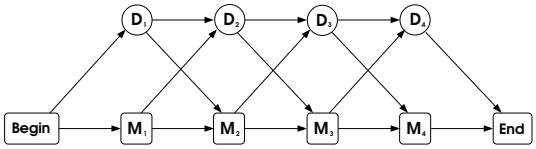


Figure 5.14. *La structure de HMM profil simplifiée retenue par Graeme Mitchison dans [Mitchison, 1999]*

[Mitchison, 1999] conserve l'idée de séparation entre les arbres de transitions quittant un état M d'une part, et les arbres de transitions quittant un état D d'autre part. Chaque nœud du HMM engendre donc trois arbres : celui des émissions de caractères sur l'état Match, celui des M-transitions et enfin celui des D-transitions. L'absence des états Insertion fait que les matrices de taux instantanés correspondant d'une part aux M-transitions et d'autre part aux D-transitions sont des matrices carrées 2×2. Mitchison choisit de leur donner une conformation particulière pour ne les faire dépendre chacune que de deux paramètres. Par exemple pour ce qui concerne les M-transitions :

$$\begin{array}{cc} & \begin{array}{cc} \text{MM} & \text{MD} \end{array} \\ \begin{array}{c} \text{MM} \\ \text{MD} \end{array} & \begin{pmatrix} a+(1-a)e^{-rd} & (1-a)(1-e^{-rd}) \\ a-ae^{-rd} & 1-a+ae^{-rd} \end{pmatrix} \end{array}$$

d représente la distance évolutive pour laquelle la matrice est conçue, $r \geq 0$ est le paramètre de taux global d'évolution, tandis que a détermine la distribution d'équilibre entre MM et MD. En effet, lorsque $d \rightarrow \infty$, $e^{-rd} \rightarrow 0$ et la matrice tend vers $\left(\begin{smallmatrix} a & 1-a \\ a & 1-a \end{smallmatrix}\right)$. La condition de réversibilité impose alors le vecteur d'équilibre $(a, 1-a)$ pour (MM, MD). Mitchison estime séparément les paramètres a et r pour les deux matrices $M\cdot$ et $D\cdot$, à partir de 5000 alignements de quatre séquences tous issus de la base de données Pfam, et trouve les valeurs $(a = 0,983; r = 6,9.10^{-3})$ pour les M-transitions, et $(a = 0,201; r = 3.10^{-3})$ pour les D-transitions.

Le travail de Mitchison est le premier qui montre d'une façon claire et efficace que la prise en compte du chemin pris par les séquences dans le HMM apporte une information supplémentaire permettant d'accroître la vraisemblance du modèle. Ne travaillant que sur des quadruplets de séquences, l'auteur établit le raisonnement suivant :

1. Sur un quadruplet de séquences {A,B,C,D}, et si l'on considère des arbres non racinés, il n'y a que trois topologies possibles, que l'on peut décrire chacune sous la

forme d'une bipartition sur l'ensemble des feuilles : {A,B|C,D}, {A,C|B,D} ou bien {A,D|B,C} (cf. figure 5.15).

2. Pour un modèle phylogénétique donné (incluant ou ignorant les substitutions entre transitions dans le HMM), et pour un quadruplet de séquences, la puissance discriminative du modèle peut se mesurer par l'écart entre la vraisemblance donnée par la meilleure des trois topologies et la vraisemblance donnée par la deuxième meilleure topologie. Si L_0 est la topologie donnant la meilleure vraisemblance aux séquences {A,B,C,D}, et L_1 et L_2 les deux autres, alors on définit $\Delta L = \min(L_0 - L_1, L_0 - L_2)$. ΔL mesure la puissance discriminative du modèle utilisé pour déterminer quelle est la meilleure topologie. On peut aussi le voir comme un indice de confiance dans la topologie L_0.
3. Après avoir estimé les distances deux-à-deux entre les séquences, la vraisemblance peut se calculer soit en fonction des seuls scores d'alignement des caractères (obtenus par une matrice de type PAM), soit en prenant en compte également les scores d'alignement des transitions (obtenus par une matrice telle que décrite ci-dessus). Appelons ΔL^+ l'écart décrit ci-dessus lorsqu'on prend en compte l'alignement des transitions et ΔL^- l'écart lorsque seuls le contenu aligné (nucléotides ou acides aminés) est pris en compte.
4. Si la différence $(\Delta L^+ - \Delta L^-)$ est significativement positive, alors cela signifie que la prise en compte de l'alignement des transitions apporte une quantité d'information qui vient renforcer la confiance que l'on a dans la meilleure topologie. On peut dire alors que le signal issu de l'alignement des transitions est corrélé avec le signal phylogénétique présent dans la topologie L_0.

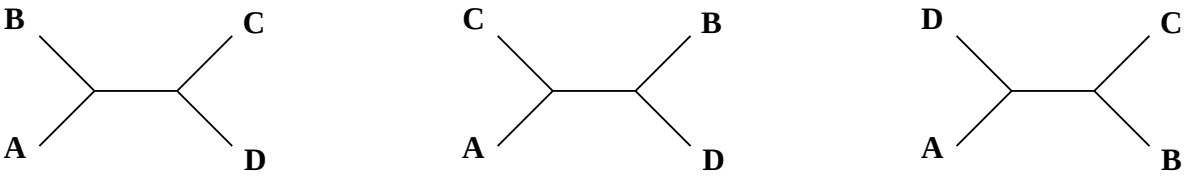


Figure 5.15. *Il n'existe que trois topologies d'arbre binaire sur quatre taxa, définies chacune par une bipartition de l'ensemble des feuilles.*

Mettant ce raisonnement en pratique, Mitchison montre que dans la grande majorité des cas, on a bien une différence $(\Delta L^+ - \Delta L^-)$ qui est positive, et donc la présence d'un signal phylogénétique dans les alignements de transitions qui corrobore le signal contenu dans les alignements correspondants d'acides aminés.

Mitchison remarque que le tree-HMM induit en chacune de ses feuilles k un HMM classique : c'est le HMM dont les probabilités d'émission et de transition sont calculées sachant l'arbre, les modèles de substitution et le contenu porté par toutes les feuilles à

l'exception de la feuille k. Mitchison exploite cette constatation pour réaliser un mécanisme d'échantillonnage de Gibbs : pour échantillonner des alignements, il suffit de faire évoluer un alignement en réalignant à chaque étape l'une des séquences aux feuilles sur le HMM induit sur cette feuille par l'alignement courant des séquences présentes aux autres feuilles. L'échantillonneur de Gibbs itère ainsi en passant successivement sur toutes les feuilles, et la procédure prend fin lorsque l'alignement est stable. Malheureusement cette procédure donne, lorsqu'elle est elle-même répétée plusieurs fois sur le même jeu de données, des scores de séquence très variables. Mitchison explique que le fait d'échantillonner également aux nœuds internes (le calcul des probabilités en un nœud se fait en deux parcours de l'arbre, le premier des feuilles à la racine et le second en sens inverse) permet de se sortir de ce faux pas. Au final il obtient des résultats légèrement moins bons que ceux donnés par CLUSTALW sur des séquences de globines issues de Pfam : en comptant la proportion de paires de résidus correctement alignées, Mitchison obtient un score moyen de 0,615 avec ses tree-HMM contre 0,631 pour CLUSTALW. Il remarque cependant que sur les mêmes données, la méthode de recuit simulé intégrée à HMMER 1.8.4 ne réalise qu'un piètre score de 0,257 (dès la version 2.0, HMMER se défaisait de toute prétention à aligner correctement lui-même des séquences par de telles méthodes. Aujourd'hui, HMMER 3.0 propose une procédure itérative « classique » de recherche de séquences homologues dans les bases pour enrichir un alignement à partir d'une seule séquence requête, *à la* PSI-BLAST).

5.2 Des phylo-HMM pour annoter des alignements

5.2.1 Siepel et Haussler, combinaison de modèles phylogénétiques et HMM pour l'analyse des séquences biologiques

Ce papier correspond à une communication à la conférence RECOMB d'avril 2003 [Siepel et Haussler, 2003] et a fait l'objet d'une republication l'année suivante sous une forme légèrement révisée [Siepel et Haussler, 2004a]. Siepel et Haussler y exposent la théorie de leurs phylo-HMM : l'idée est d'avoir un HMM dont chacun des états est susceptible de modéliser par un arbre phylogénétique avec des paramètres propres (longueurs de branche, taux global d'évolution, mais aussi processus de substitution et distribution stationnaire associée), l'évolution d'un certain type de sites. Les transitions entre les différents états du HMM décrivent les probabilités conditionnelles d'appartenance du site courant à un type de site donné, sachant la catégorie du site précédent dans l'alignement. Siepel et Haussler ne s'intéressent pas dans ce papier au problème de l'apprentissage des paramètres d'un tel phylo-HMM : ils sont appris par comptage sur un alignement annoté, sans même qu'il soit fait usage de pseudocomptes additionnels.

L'intérêt du papier de Siepel et Haussler réside dans la mise en évidence de l'apport de

l'approche phylo-HMM par rapport à une approche classique (c'est-à-dire dans laquelle tous les sites de l'alignement sont générés par une seule et même phylogénie avec des paramètres qui ne dépendent pas du site), ce en terme de log-vraisemblance de l'alignement. Leur propos consiste donc à montrer combien les phylo-HMM permettent d'avoir une représentation plus fidèle de l'histoire évolutive qui a engendré les séquences contemporaines prises dans un alignement multiple.

Les différents états des phylo-HMM considérés dans ce papier modélisent soit des taux d'évolution différents, avec un paramètre d'autocorrélation entre taux d'évolution d'un site et du site suivant dans l'alignement, soit des catégories fonctionnelles différentes (c'est-à-dire première à troisième positions de codon, site intergénique, introns).

Les données utilisées ici par Siepel et Haussler sont issues d'un super-alignement de quelque 2 millions de bases nucléotidiques sur 9 espèces modèles appartenant toutes à la classe des mammifères placentaires (Eutheria). Cet alignement est issu d'une région autour du gène CFTR (pour Cystic Fibrosis Transmembrane conductance Regulator), comprenant à la fois des zones codantes et non codantes. Les auteurs comparent différents modèles de substitution de nucléotides et de dinucléotides avec ou sans classes Gamma ou paramètre d'autocorrélation pour les taux globaux d'évolution, et avec ou sans catégories fonctionnelles encodées dans différents états du phylo-HMM. Ils montrent que la modélisation de différentes catégories fonctionnelles en différents états d'un phylo-HMM permet d'améliorer grandement la log-vraisemblance de l'alignement (écarts allant jusqu'à plusieurs milliers d'unité de log-vraisemblance).

5.2.2 Siepel et Haussler, modèles de Markov cachés phylogénétiques

Dans un livre publié en 2005 chez Springer par Rasmus Nielsen (*Statistical Methods in Molecular Evolution*), un chapitre a été rédigé par Adam Siepel et David Haussler dans lequel ils exposent en détail l'idée et le mode de fonctionnement de leurs modèles phylo-HMM. Une illustration claire valant souvent mieux qu'un long discours, nous republions ici avec l'autorisation des auteurs le schéma introductif apparaissant dans ce chapitre (figure 5.16).

Après avoir donné une définition formelle de leurs phylo-HMM, les auteurs présentent trois exemples d'application.

Le premier exemple donné par Siepel et Haussler consiste en un phylo-HMM à quatre états, dont trois modélisent les trois positions de codon (les objets générés sont des alignements de nucléotides) et le quatrième modélise les sites non codants. La phylogénie supportant les taxa est supposée connue, de même que tous les paramètres du HMM (probabilités de transition entre états, modèles de substitution agissant sur les arbres, arbres

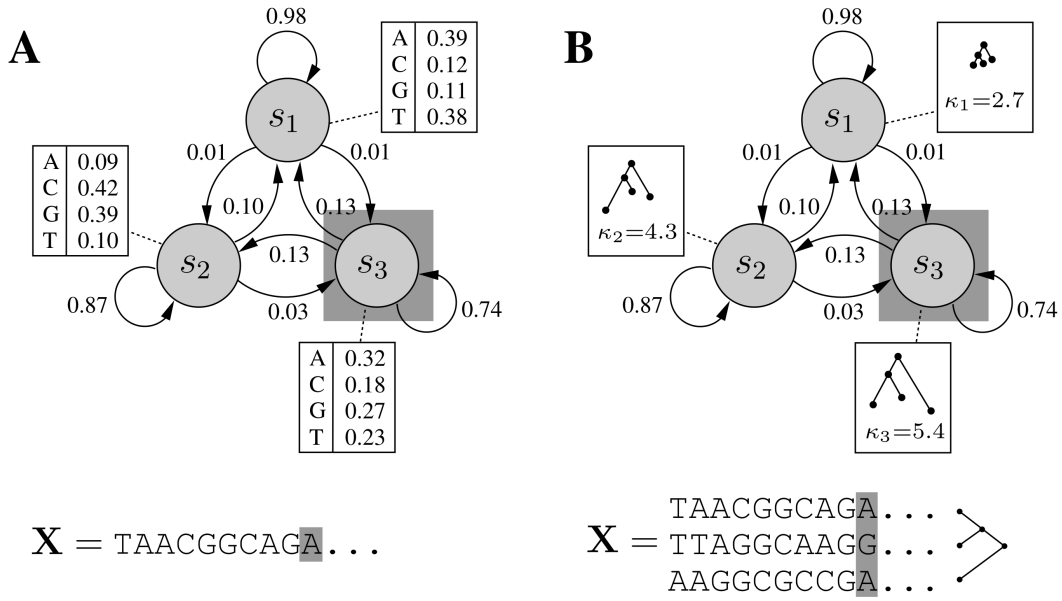


Figure 5.16. *En (**A**), un HMM à trois états générant une séquence de nucléotides. Chacun des états encode une distribution de probabilités sur les quatre bases. Dans un cadre génératif, on tire aléatoirement à chaque étape une des quatre bases, selon la distribution de probabilités de l'état courant. Puis on sélectionne l'état suivant en tirant une transition selon la distribution de probabilités sur les arêtes sortantes de l'état courant. Dans un cadre d'évaluation du score d'une séquence dans le HMM, on effectue simplement le produit des probabilités de transition et d'émission le long du chemin donnant le meilleur score à la séquence (approche de Viterbi). Chacun des trois états du HMM possède une distribution particulière (AT-riche pour s_1, GC-riche pour s_2 et favorisant les purines (A et G) pour s_3).*
*En (**B**), un **phylo-HMM** à trois états également. Dans un tel modèle, chaque état génère non plus un unique caractère, mais une colonne d'un alignement. Les caractères formant cette colonne sont par construction liés entre eux par l'évolution le long des branches d'un arbre phylogénétique, lequel arbre joue le rôle des distributions de probabilité sur les nucléotides vues en (**A**). Ici les trois états reprennent la même topologie d'arbre mais dilatent plus ou moins les branches selon un facteur d'élongation κ_i. Ils modélisent ainsi différents taux globaux d'évolution : s_1 modélise un faible taux global d'évolution, tandis que les colonnes issues de s_2 et a fortiori de s_3 présenteront des caractères plus divergents les uns par rapport aux autres.*

correspondant à chacun des quatre états et paramètres associés). Les auteurs se servent du modèle lui-même pour générer des données tout en maintenant une table d'association entre sites synthétisés et états du phylo-HMM. Ils se servent ensuite du même phylo-HMM pour prédire le chemin emprunté par l'alignement dans le HMM, et testent la pertinence de l'analyse en observant la corrélation entre le chemin utilisé lors de la construction de l'alignement et le chemin prédit par l'algorithme de Viterbi. La comparaison est effectuée entre le modèle phylo-HMM et un modèle de type HMM non phylogénétique qui tire aléatoirement chacun des caractères dans une colonne en fonction de la distribution de fond de l'état courant, sans tenir compte des liens évolutifs entre les taxa. Les auteurs montrent une sensitivité et une spécificité du phylo-HMM qui s'établissent toutes deux à 0,98 là où le HMM non phylogénétique voit sa sensitivité et surtout sa spécificité décroître fortement lorsque le nombre d'espèces dans l'arbre augmente. Ce premier exemple est illustratif et constitue une sorte de preuve du concept des phylo-HMM. Cependant, il constitue de l'aveu même des auteurs une simplification extrême de la réalité, puisque le modèle à l'origine des données est parfaitement connu et correspond exactement au modèle servant pour l'analyse. Une telle situation n'existe pas lorsqu'on se penche sur l'étude de données biologiques réelles.

Le deuxième exemple donné par Siepel et Haussler est celui d'un phylo-HMM à dix états, chacun étant censé modéliser une catégorie de taux d'évolution globale distincte. Un des états s_1 représente une évolution lente, les autres des évolutions plus rapides. L'objectif d'un tel phylo-HMM est de découvrir dans un alignement les zones à évolution lente et donc bien conservées, car on suppose généralement que de telles zones sont fonctionnellement importantes [Thomas *et al.*, 2003a]. Les transitions entre états sont déterminées suivant un paramètre appelé « paramètre d'autocorrélation » $\lambda \in [0,1]$ qui indique à quel point il est probable qu'un site donné évolue dans la même classe de taux d'évolution que le site qui le précède immédiatement dans l'alignement. La phylogénie support ainsi que le paramètre d'autocorrélation sont appris sur les données (un alignement de séquences nucléotidiques de 1,8 million de paires de base sur neuf mammifères euthériens). Pour chaque site X_i de l'alignement, les auteurs évaluent la probabilité (dite « postérieure ») que ce dernier ait été généré par l'état correspondant à l'évolution lente, et en font un score de conservation : cons_score = $\Pr(\phi_i = s_1 | X_i, \theta)$ où ϕ_i est l'état du phylo-HMM correspondant au site numéroté X_i et θ l'ensemble des paramètres caractérisant le phylo-HMM. Siepel et Haussler montrent un profil de conservation des sites ainsi déduit qui correspond exactement aux scores de conservation obtenus sur le même jeu de données par Thomas et coauteurs [Thomas *et al.*, 2003a].

Il faut noter qu'une telle approche (un HMM dont les nœuds représentent chacun un taux d'évolution distinct) avait déjà été publiée auparavant [Felsenstein et Churchill, 1996].

Enfin, le troisième exemple utilisé par Siepel et Haussler dans leur chapitre s'appuie

sur un phylo-HMM à un seul état. Le fait essentiel est donc simplement qu'il s'agit d'un modèle de génération de sites tout entiers et non pas de caractères comme c'est le cas pour les HMM standard. Siepel et Haussler se servent de cet exemple extrait de [Siepel et Haussler, 2004b] pour comparer différents modèles de substitution de nucléotides, de dinucléotides ou de trinucléotides (c'est-à-dire des modèles de Markov d'ordre supérieur à 1). Les auteurs montrent des gains substantiels lorsque l'ordre des modèles de Markov augmente, et également lorsqu'on modélise des doublets ou triplets *chevauchants* (e.g. $N-2$ triplets de sites pour un alignement de longueur N) par rapport à la situation où l'on modélise des doublets ou triplets *consécutifs et non chevauchants* (e.g. $N/3$ triplets de sites pour le même alignement). Bien que dans cet exemple ce ne soit pas directement le caractère phylogénétique du processus de génération des sites qui soit évalué, le fait que les modèles de substitution enrichis apportent des gains de vraisemblance (c'est-à-dire de *goodness of fit*) valide indirectement le concept de « phylogénisation » des états du HMM.

Pour donner une conclusion provisoire à ces travaux d'Adam Siepel et de David Haussler, nous pouvons faire quelques remarques au sujet de « leurs » phylo-HMM :

1. la phylogénie de support des taxa est toujours connue. Les exemples choisis impliquant en général des taxa proches les uns des autres, bien séquencés et bien annotés (mammifères euthériens), avec des taux d'évolution toujours relativement faibles si l'on se place du point de vue de l'ensemble du règne du vivant. Cette caractérisitque de fiabilité des données sur lesquelles travaillent les auteurs pose la question de la robustesse de leurs modèles par rapport à d'éventuelles erreurs d'alignement ou de placement phylogénétique.
2. la question de l'*apprentissage* des paramètres du modèle n'est jamais posée : on a toujours affaire à un modèle soit dont les paramètres ont été imposés *a priori* par une connaissance approximative des données à générer, soit à des paramètres en nombre très restreint (par exemple le paramètre d'autocorrélation λ du deuxième exemple) qui ont pu être rapidement appris sur les séquences en entrée. Ces séquences étaient donc annotées, c'est-à-dire que l'on savait déjà à l'avance quel état du phylo-HMM était associé à chacun des sites.
3. de façon corrélative, les phylo-HMM décrits par Siepel et Haussler ne comportent jamais un grand nombre d'états : il n'est pas question de modèles dont la longueur serait par exemple proportionnelle à la taille de l'alignement modélisé.
4. ces modèles ne servent jamais à augmenter l'ensemble d'apprentissage : il n'est pas question dans ces travaux d'aller rechercher des séquences apparentées dans une base de données. Pour ce faire tout en restant dans le cadre des phylo-HMM tels que décrits ici, il faudrait à la fois aligner les séquences candidates et étendre simultanément la phylogénie aux espèces dont elles seraient issues.

5.3 Des modèles pour rechercher des homologues distants

5.3.1 Continuation de l'idée des Tree-HMM par Qian et Goldstein

En 2003, Bin Qian et Richard Goldstein [Qian et Goldstein, 2003] ont publié un papier reprenant et développant une bonne partie des idées originellement proposées par [Mitchison et Durbin, 1995]. Rappelons que le propos de Mitchison et Durbin était plutôt axé sur l'obtention de vraisemblances accrues pour des modèles phylogénétiques incluant les transitions empruntées par les séquences dans un HMM profil donné. Les auteurs escomptaient pouvoir se servir d'un tel modèle pour éventuellement réaligner les séquences et déterminer la meilleure topologie d'arbre reliant les séquences en question. Le point de vue de Qian et Goldstein est sensiblement différent : ils se servent de modèles évolutifs calculés sur les transitions pour déterminer, à partir d'un seul alignement de séquences, non pas un mais plusieurs HMM profils, échantillonnant l'espace des HMM profils adaptés aux différents points de la phylogénie des taxa impliqués.

Cette approche de Qian et Goldstein est fondamentalement la nôtre : dans cette thèse, nous montrons pourquoi de notre point de vue le travail réalisé par ces deux auteurs est incomplet et mérite certains amendements. Ceci étant dit, il n'en reste pas moins que ce travail est essentiel pour ce qui concerne la modélisation combinée d'aspects évolutifs et longitudinaux en bioinformatique, et nous en donnons ci-dessous quelques points d'explication.

Démarche générale

Les auteurs introduisent leur démarche en faisant quelques remarques quant au processus de construction des HMM profils classiques :

1. il n'existe pas de méthode de pondération des séquences qui ne soit pas une heuristique basée sur des idées plus ou moins « ad hoc ». Nous avons présenté ces méthodes dans la section 3.4,
2. les techniques de pondération en question fonctionnent toutes de manière à maximiser l'entropie du jeu de séquences pondérées. En pratique elles amplifient les « erreurs de casting » en donnant plus de poids aux séquences singulières au sein de l'ensemble d'apprentissage,
3. les séquences dites « homologues » à l'ensemble d'apprentissage n'ont pas de raison de former un groupe homogène. En effet on a toutes les raisons de penser que l'ensemble des homologues est divers et suit une certaine classification dans laquelle on devrait retrouver trace de la phylogénie sous-jacente.

Ayant soulevé ces observations, les auteurs plaident pour une représentation dans laquelle on ne retiendrait pas *un seul* modèle mais bien plutôt un ensemble de *représentants*

stochastiques de la diversité présente dans la famille de séquences d'apprentissage. Qian et Goldstein se proposent de construire de tels jeux de représentants (chacun étant un HMM profil tout à fait standard) en tenant compte *explicitement* de la phylogénie des taxa formant l'ensemble d'apprentissage.

Ainsi, les auteurs se proposent de construire autant de HMM profils que la phylogénie comporte de nœuds, en estimant les paramètres d'émission d'acides aminés et de transition entre nœuds du HMM à partir de la phylogénie reliant les taxa entre eux. Autant que faire se peut (c'est-à-dire, essentiellement, pour ce qui concerne les émissions sur les états Match et les transitions à partir d'états Match ou de Délétion, cf. plus loin), la dérivation des paramètres en question se fait par un mécanisme de *reconstruction ancestrale* : en prenant l'exemple des émissions d'acides aminés A_x, avec un arbre phylogénétique T, un ensemble de paramètres θ décrivant le modèle évolutif sous-jacent (distribution d'équilibre correspondante π), une colonne c quelconque prise dans l'alignement d'apprentissage (acides aminés observés D_c), le caractère inconnu Ω_c porté par le nœud interne Ω de l'arbre T pour cette colonne suit une distribution donnée par la formule suivante :

$$\Pr(\Omega_c = A_x | T, \theta, D_c) = \frac{\Pr(D_c | T, \theta, \Omega_c = A_x)\pi(A_x)}{\Pr(D_c | T, \theta)} \tag{5.1}$$

Cette formule décrit les modalités pratiques de calcul des paramètres du HMM que l'on peut appeler « de reconstruction ancestrale » correspondant au nœud Ω.

Sélection des colonnes « match »

Ce point n'est évoqué que rapidement dans [Qian et Goldstein, 2003], et pourtant il est crucial. Nous avons déjà (section 5.1.2) eu à insister sur le fait que les transitions sur lesquelles nous allons nous pencher par la suite n'ont aucun caractère absolu, elles n'existent que *par rapport* à une architecture bien définie, c'est-à-dire par rapport à un choix de colonnes faisant l'objet d'une modélisation par un état Match. Selon leurs dires, Qian et Goldstein utilisent l'approche historique la plus simple, proposée dès la naissance des HMM profils [Krogh *et al.*, 1994] et consistant à faire un état Match de toute colonne qui affiche plus de 50% de résidus (le reste étant formé de gaps). Cette approche simple perdure et revient même jouer les premiers rôles, puisque HMMER3 l'a choisie comme méthode par défaut alors que ce n'était pas le cas dans HMMER2.

Construction des matrices de substitution de transitions

Pour modéliser les substitutions entre acides aminés et donc l'évolution des émissions depuis les états Match, les auteurs utilisent le processus WAG [Whelan et Goldman, 2001]. En ce qui concerne les transitions, le travail d'estimation des paramètres des processus est indispensable puisqu'il n'existe pas de matrice standard qui soit déjà disponible. Qian

et Goldstein choisissent de partir de la base d'alignements structurels appelée *Combinatorial Extension* [Shindyalov et Bourne, 1998]. Ils en extraient dix alignements composés systématiquement d'une séquence étant censée représenter une structure PDB donnée et de ses 29 homologues les plus distants (en termes de pourcentage d'identité de séquence avec le « représentant »). À partir de chacun de ces 10 alignements de 30 séquences chacun, les auteurs infèrent un arbre phylogénétique puis un modèle de substitution en trois étapes :

1. détermination de topologies candidates à l'aide du logiciel MOLPHY [Adachi *et al.*, 1996],
2. sélection de la meilleure topologie et calcul des longueurs de branche à l'aide de PAML [Yang, 1997] (approche de maximum de vraisemblance),
3. optimisation par la méthode du simplexe des paramètres de la chaîne de Markov correspondant aux processus de substitution entre transitions, de façon à maximiser la vraisemblance de l'arbre construit aux étapes précédentes.

Il faut noter que le processus de construction de l'arbre se fait sur la base de l'alignement de séquences d'acides aminés, non de caractères dénotant des transitions. L'arbre inféré sur des séquences protéiques est ensuite utilisé tel quel pour calculer des vraisemblances sur des alignements de transitions.

Ce processus de construction de processus markoviens aboutit à la mise au jour de *deux processus distincts* : tout comme [Mitchison et Durbin, 1995], les auteurs font le choix de dissocier les histoires évolutives se rapportant d'un côté aux transitions quittant les états Match, et d'un autre côté à celles quittant les états de Délétion. Ce sont donc deux matrices 3×3 qui sont estimées dans [Qian et Goldstein, 2003].

Enfin, un processus de substitution entre transitions I→M et I→D est calculé de même. En effet, à condition de les rejeter systématiquement en fin de zone d'insert, ces transitions sont alignables les unes aux autres (voir plus loin notre présentation du problème de l'alignement des transitions en section 9.3.2). Ce processus à deux états Q_I est utilisé dans ce qui suit.

Traitement des états d'Insertion

Contrairement à Mitchison et Durbin, Qian et Goldstein font bien apparaître des états d'Insertion dans leur modèle, mais ne calculent pas de façon phylogénétique tous les paramètres du modèle de substitution pour les transitions qui en sortent (transitions abréviées IM, ID et II). Dans un premier temps, les auteurs infèrent d'abord le générateur Q_I du processus de substitution entre transitions IM et ID à partir d'une analyse de maximum de vraisemblance (cf. section précédente). Mais ensuite, l'information phylogénétique est purement et simplement ignorée pour déterminer une probabilité de transition

II *indépendante de la position dans l'arbre du nœud* Ω *de la reconstruction ancestrale*. Cette probabilité est calculée simplement à partir du nombre d'observations IM, ID et II dans le HMM profil construit à partir de l'alignement original des séquences d'apprentissage, avec lissage par pseudo-comptes de Laplace. C'est-à-dire que pour la zone d'insertion entre les états Match c et $c+1$ de l'alignement d'apprentissage et des comptages correspondants $N_{IM,c}$, $N_{ID,c}$ et $N_{II,c}$, on a :

$$\Pr_c(II) = \frac{N_{II,c}+1}{N_{IM,c}+N_{ID,c}+N_{II,c}+3}$$

Une étape de normalisation est finalement nécessaire pour calculer les trois probabilités au départ de l'état I du tree-HMM correspondant au nœud Ω de l'arbre, à partir de cette probabilité de bouclage II calculée sans information phylogénétique et des deux probabilités IM et ID calculées à partir de l'évolution du processus Q_I le long des branches de l'arbre.

Résultats

Une fois les phylo-HMM construits (les auteurs ne disent pas de quelle manière ils traitent les émissions sur les états d'Insertion), Qian et Goldstein utilisent ceux-ci pour rechercher des homologues dans une base de données. Pour toute séquence seq, chacun des tree-HMM de reconstruction ancestrale $\mathscr{M}$ donne un score $s_{\mathscr{M}}(\text{seq})$ établi de la manière suivante :

$$s_{\mathscr{M}}(\text{seq}) = \log\left(\frac{\Pr(\text{seq}|\mathscr{M})}{\Pr(\text{seq}|\mathscr{M}^{\text{rev}})}\right)$$

où $\mathscr{M}^{\text{rev}}$ est le modèle $\mathscr{M}$ renversé : le dénominateur de l'expression ci-dessus est la probabilité que la séquence *renversée* ait été engendrée par le modèle $\mathscr{M}$. Sur un jeu de données de taille relativement modeste issu de SCOP (39 superfamilles, 39 tests, 1063 séquences d'apprentissage pour 215 cibles à identifier), les auteurs démontrent des gains nets en terme de performance de détection, par rapport à HMMER ou d'autres méthodes de modélisation (Family-BLAST, PSI-BLAST avec une seule itération, PROSITE).

Un avantage important de la méthode proposée par Qian et Goldstein est qu'elle met en jeu des modèles HMM construits de façon particulière mais ensuite tout à fait utilisables avec les logiciels existants (par exemple ceux de la suite HMMER), qu'il s'agisse de faire des recherches dans les bases de données, d'affecter des scores à des séquences données ou encore d'aligner des séquences. Les bons résultats affichés par les publications de ces auteurs ([Qian et Goldstein, 2004] reprend la méthodologie de [Qian et Goldstein, 2003] en se plaçant dans un cadre de raffinement itératif) nous ont poussé à explorer plus avant les idées de ceux-ci pour élaborer les modèles que nous présentons dans la suite de cette thèse.

Deuxième partie

Méthodologie

CHAPITRE

6 Schéma général

Soit un ensemble de séquences homologues. On souhaite donner de cet ensemble une modélisation aussi puissante que possible, permettant d'aller retrouver dans les grandes bases de données le plus possible d'homologues distants, en évitant bien sûr que les recherches ne soient polluées par de trop nombreux faux positifs. Partir de séquences alignées entre elles est un impératif si l'on veut non pas les considérer comme un ensemble de points isolés, mais faire émerger leurs caractéristiques fonctionnelles ou structurelles communes.

Le travail présenté ici propose de prendre comme point de départ à la fois les HMM profils et les phylogénies au maximum de vraisemblance. En effet, ces deux classes d'outils sont bien adaptées lorsqu'il s'agit de décrire les propriétés structurales caractérisant un ensemble de séquences (modélisation longitudinale des HMM profils) ou les relations évolutives qui relient ces séquences entre elles (modélisation « verticale » ou historique par les phylogénies). À partir de ce double point de départ, nous nous proposons d'établir des modèles capables d'aller rechercher efficacement des protéines homologues dans une base de séquences.

Sommaire

6.1 Problématique et objectifs

Si l'on ne considère que l'alignement multiple des séquences, la solution de modélisation apportée par les HMM profils semble être si ce n'est la meilleure, du moins une excellente solution : la pondération des séquences les unes par rapport aux autres permet de maximiser la quantité d'information présente dans l'alignement des séquences pondérées, et le processus d'apprentissage du HMM profil à partir d'un schéma *Expectation-Maximisation* permet de dériver des paramètres modélisant au mieux l'alignement d'apprentissage.

Le HMM profil ainsi construit à partir d'un alignement donne des performances correctes lorsqu'il s'agit de détecter des homologues relativement proches des séquences d'apprentissage [Krogh *et al.*, 1994]. Mais dès lors que l'on souhaite retrouver des homologues distants, les performances en détection données par les HMM profils se dégradent. On peut avancer plusieurs raisons à cela :

- plus la distance augmente, et plus les séquences sont dissemblables. En particulier, la distance entre la cible et le *consensus* modélisé par le HMM profil (séquence donnée par la suite d'émissions et de transitions de plus forte probabilité) augmente elle aussi. Le modèle le plus à même de détecter un homologue distant donné s'écarte donc du modèle « médian » qu'est le HMM profil standard, obtenu via l'un des schémas de pondération des séquences décrits au chapitre 3.4.
- à mesure que la cible s'éloigne (évolutivement parlant) des séquences d'apprentissage, les liens entre celles-ci et celle-là se font de plus en plus ténus. Un unique modèle de type HMM profil ne suffit plus alors à rendre compte de la grande diversité des séquences homologues existant potentiellement à une distance élevée de chacune des séquences d'apprentissage. En particulier, l'insertion ou la délétion de larges parts des séquences a pu se produire entre temps, donnant lieu à des homologues dont l'un n'est alignable qu'avec une partie de l'autre.

Si des différences notables entre les séquences permettent de donner un partitionnement logique de l'ensemble d'apprentissage, alors on peut envisager de former une telle partition en k sous-ensembles, puis de construire k HMM profils, chacun à partir de l'un des sous-ensembles. Une telle approche a déjà été publiée [Edgar et Sjölander, 2003]. Elle présente le désavantage de construire des modèles à partir d'un ensemble réduit de données, l'étape de pondération globale des séquences renforçant alors la confiance accordée aux connaissances *a priori*, ce qui contribue à noyer dans le bruit le signal correspondant aux observations issues de l'ensemble d'apprentissage.

On peut vouloir faire mieux. Pour cela, il nous faut d'abord constater que l'information phylogénétique apporte en elle-même une structuration sous-jacente de l'ensemble des séquences d'apprentissage dont on aurait tort de se priver lorsqu'elle est connue.

Dans l'exemple présenté en figure 6.1, on montre deux zones (l'une en bleu et l'autre en rouge) de l'arbre phylogénétique correspondant à des clades dont les taxa présentent tous des caractéristiques communes au niveau des séquences (insertions relativement bien conservées par rapport aux autres séquences d'apprentissage). Notre objectif est de fabriquer des HMM dits « de reconstruction ancestrale » par exemple au niveau des nœuds de la phylogénie peints en bleu et en rouge, chacun des deux modélisant le mieux possible les séquences homologues *dans le voisinage phylogénétique* du point considéré. L'exemple montre clairement que les états Match de l'un et de l'autre ne correspondront pas aux mêmes séquences de l'alignement et que les probabilités de transition entre états du HMM n'auront rien à voir entre l'un et l'autre des deux modèles.

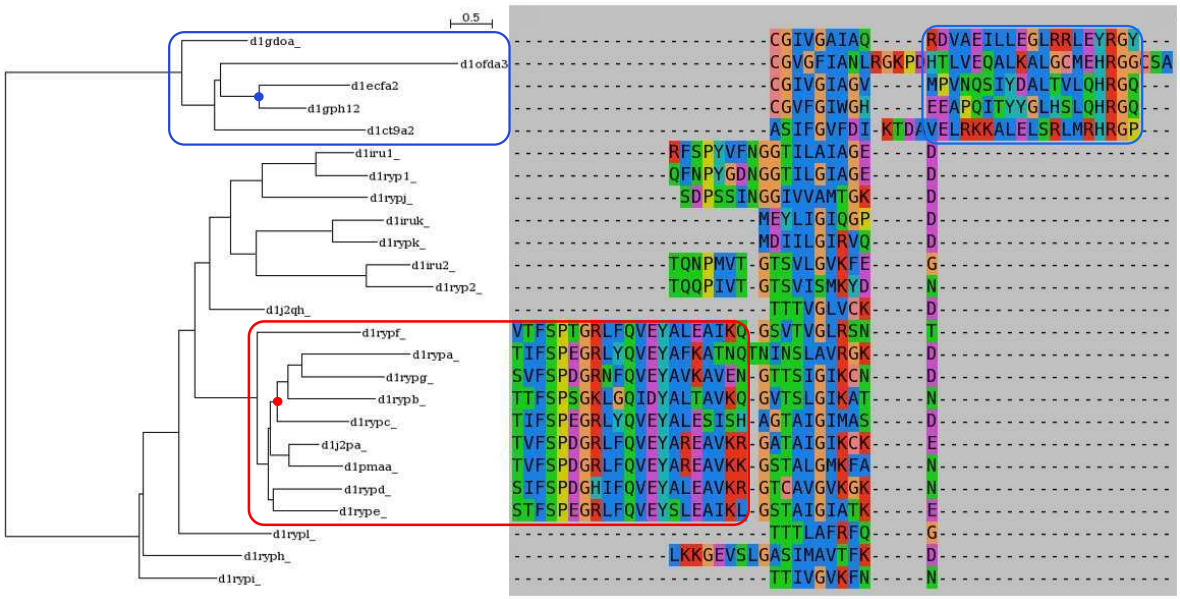


Figure 6.1. *Un exemple d'alignement avec la phylogénie supportant les séquences (glutamine amidotransférases de classe II, groupe SABmark n° 336)*

L'approche classique consistant à modéliser la famille d'apprentissage par un HMM profil puis à effectuer les recherches sur la base de ce modèle, ignore totalement les relations phylogénétiques entre les séquences constituant l'ensemble d'apprentissage. Par le biais de la pondération des séquences en amont de la création d'un HMM, ce dernier tente de maximiser la quantité d'information encodée en donnant systématiquement un poids accru aux séquences singulières au sein de l'ensemble d'apprentissage (cf. section 3.4). En dehors de ce mécanisme d'ajustement, les séquences d'apprentissage entrent toutes de la même façon dans le processus de comptage qui va établir les paramètres du HMM profil. À l'issue de sa conception par l'approche standard de HMMER, ledit HMM profil n'est

qu'une modélisation stochastique d'une sorte de protéine médiane ou « consensus », censée condenser en elle toute l'information structurale présente dans l'alignement d'entrée : ce n'est *qu'un* modèle, maximisant plus ou moins la vraisemblance des données d'apprentissage tout en gardant une souplesse statistique permettant normalement de donner également de forts scores aux séquences *homologues* à l'ensemble d'apprentissage. Cette approche trouve cependant ses limites lorsqu'il s'agit de repérer des homologues dans un contexte de forte divergence évolutive.

Pour la recherche de séquences homologues, une autre approche consisterait à prendre individuellement chacune des séquences d'apprentissage, pour ensuite lancer une recherche de type BLAST à partir de chacune d'elles (cf. section 2.2.3). Restera enfin à rassembler les résultats individuels pour tenter de faire émerger les cibles recherchées.

Les deux approches décrites ci-dessus sont très différentes : l'une synthétise *d'abord* les informations provenant de plusieurs séquences avant de construire un modèle et d'utiliser ce dernier comme outil opératoire de recherche, tandis que l'autre approche construit autant de « modèles » qu'on lui donne de séquences d'apprentissage (ici modèle et séquence ne font qu'un), les utilise comme opérateurs de recherche puis synthétise les résultats obtenus.

L'approche que nous décrivons dans cette partie s'inspire de ces deux-là, tout en donnant une place prépondérante à la phylogénie sous-tendant l'ensemble d'apprentissage. En effet, il va s'agir à la fois de :

- construire *plusieurs* modèles distincts de type HMM profil, chacun de ces modèles étant plus particulièrement adapté à un voisinage phylogénétique donné,
- *inférer à partir de la phylogénie* l'ensemble des paramètres des différents modèles.

À partir d'un ensemble d'apprentissage, on construira donc un ensemble de modèles de description longitudinale en utilisant de manière centrale l'information phylogénétique. Les deux classes de modèles dont nous parlions plus haut se trouveront ainsi unies pour construire les outils opératoires de la recherche d'homologues. Nous escomptons de la diversité des modèles stochastiques ainsi produits un gain en sensibilité sans perte significative en précision.

Nous montrons ici comment, après avoir déterminé la phylogénie reliant les séquences d'apprentissage ainsi qu'un point d'intérêt au sein de cette phylogénie, on peut construire un modèle HMM profil dont *tous* les paramètres (architecture du profil, émissions sur les états Match et Insertion, probabilités de transition entre états) sont déterminés par une analyse phylogénétique faisant agir des processus substitutionnels le long des branches de l'arbre. Les paramètres en question sont détaillés sur la figure 6.2 :

1. Il faut d'abord déterminer l'*architecture* du HMM. Nous nous plaçons dans le cadre des HMM profils tels que produits par SAM ou HMMER, si bien que la structure de l'automate d'états finis est connue à l'avance. Elle correspond à ce qui est montré en figure 6.2. Si cette structure est imposée, elle est néanmoins paramétrique : il faut en effet définir le *nombre* des nœuds qui vont composer le HMM, ainsi que la *correspondance* de chacun de ces nœuds avec une certaine colonne de l'alignement d'apprentissage. La première étape de conception du HMM profil consiste donc à choisir un sous-ensemble des positions au sein de l'alignement d'apprentissage. Chacune des positions retenues fera par la suite l'objet d'une modélisation à l'aide d'un état Match (les positions non sélectionnées seront quant à elles modélisées par des états d'Insertion, cf. section 3.3.4).
2. Chaque état Match est représenté dans le HMM profil par une distribution de probabilités sur les acides aminés que cet état est susceptible d'émettre. Chaque état Match apporte donc au modèle 19 paramètres indépendants (puisque la somme des 20 probabilités d'émission est contrainte à 1) qu'il faut déterminer lors de cette deuxième étape.
3. Chaque état Match vient également avec trois transitions sortantes (vers l'état Match suivant, vers l'état d'Insertion correspondant au nœud courant ou encore vers l'état de Délétion correspondant au nœud suivant). La somme des trois probabilités de transition valant nécessairement 1, ce sont deux paramètres indépendants à déterminer. De même, les transitions sortantes d'un état de Délétion (trois par état dans le cadre défini par SAM, deux seulement dans le cadre HMMER) introduisent un ou deux paramètre(s) indépendant(s) qu'il faut déterminer lors de cette troisième étape.
4. Les probabilités de transitions quittant un état d'Insertion sont également à déterminer. Elles font l'objet d'un traitement différent de ce qui vaut pour l'étape précédente car elles présentent la particularité d'inclure une boucle I→I. Ce sont un ou deux paramètre(s) indépendant(s) à déterminer, là encore selon qu'on se place respectivement dans le cadre HMMER ou bien SAM.
5. Enfin, le cinquième type de paramètre est constitué par les probabilités d'émission sur les états d'Insertion. Le cadre n'est pas tout à fait le même que lors de l'étape (2) car chaque distribution de probabilités est ici déterminée non plus à partir d'une seule colonne de l'alignement d'apprentissage, mais à partir d'un ensemble de colonnes contiguës.

6.2 Méthode générique de « phylogénisation »

Chacun de ces paramètres pose le problème suivant : soit un arbre phylogénétique $\mathscr{T}$ dont les feuilles portent des caractères connus. Soit un point n situé sur l'arbre $\mathscr{T}$ (pour

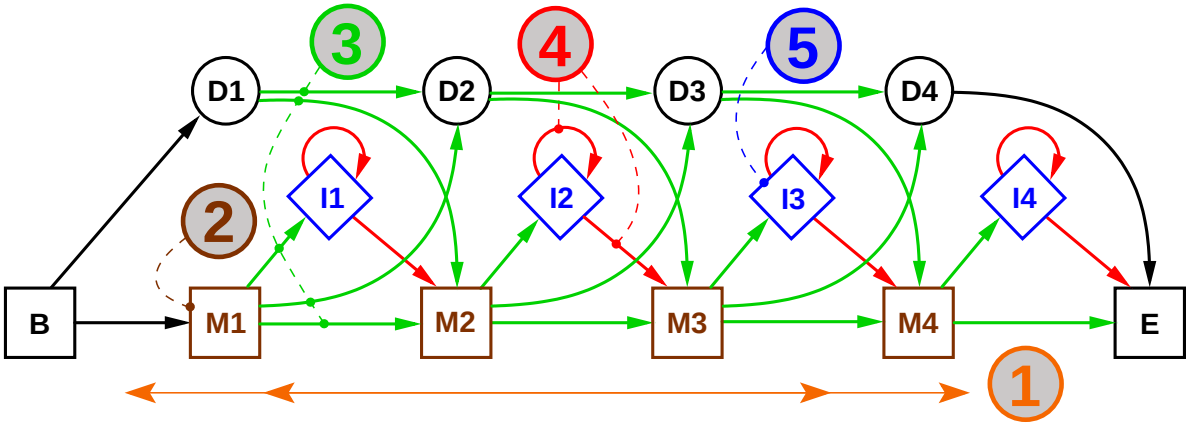


Figure 6.2. *Le cœur d'un HMM profil de type* HMMER *(architecture Plan7). Chacune des cinq étiquettes portant un chiffre correspond à un certain type de paramètres. Notre approche consiste à calculer ces paramètres en nous appuyant sur l'arbre phylogénétique construit sur les séquences d'apprentissage. Remarquons que le cadre de travail* SAM *ajouterait à ce schéma deux transitions supplémentaires par nœud : une transition D→I et une transition I→D.*

simplifier, nous nous bornerons à placer n sur les nœuds de $\mathscr{T}$). Connaissant la phylogénie $\mathscr{T}$ ainsi que l'ensemble des caractères portés par ses feuilles, quelle est la distribution statistique des caractères la plus probable au point n ? L'idée centrale que nous suivons dans notre travail est de calculer les différentes vraisemblances pour l'arbre $\mathscr{T}$ dans lequel on fait varier une contrainte qui est le caractère porté par le nœud n, puis de décréter que pour nous, la probabilité du caractère α au point n sera proportionnelle à la *probabilité postérieure* des données $\mathscr{D}$ aux feuilles de l'arbre $\mathscr{T}$ lorsqu'on fixe à α le caractère présent au nœud n :

$$\Pr(n=\alpha) \propto \Pr(\alpha)\,\mathrm{Lk}(\mathscr{T}|\mathscr{D},Q,n=\alpha) \tag{6.1}$$

On dérivera par ce moyen un ensemble de paramètres le mieux à même de modéliser une hypothétique séquence se trouvant sur le nœud en question ou dans son voisinage phylogénétique, ou encore ayant évolué, avec un degré de divergence raisonnable, depuis ce nœud.

Comment calculer $\mathrm{Lk}(\mathscr{T}|\mathscr{D},Q,n=\alpha)$? Pour cela on reprend l'algorithme d'élagage de Felsenstein exposé dans la section 4.4.2, en l'adaptant au problème qui nous préoccupe. D'emblée nous faisons l'hypothèse que le processus markovien Q est réversible, c'est-à-dire que l'orientation des branches de $\mathscr{T}$ n'a aucune importance. Dans le cas d'un arbre binaire non raciné, si le nœud n n'est pas une feuille, alors il possède trois voisins. Nous représentons cette situation en figure 6.3.

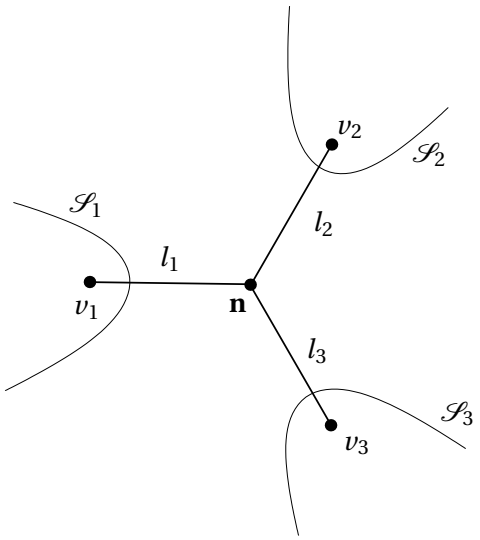


Figure 6.3. *L'arbre phylogénétique $\mathscr{T}$ non raciné, représenté autour du nœud d'intérêt, n.*

Dans une telle situation, on a :

$$\mathrm{Lk}(\mathscr{T}|\mathscr{D}, Q, n = \alpha) = \prod_{i=1}^{3} F_i(\alpha) \tag{6.2}$$

où

$$\forall\, i \in \{1,2,3\} \quad F_i(\alpha) = \sum_{\beta} \Pr(\alpha \xrightarrow{l_i} \beta)\, \mathrm{Lk}(\mathscr{S}_i|\mathscr{D}, Q, v_i = \beta) \tag{6.3}$$

Dans le cas où le nœud n est lui-même une feuille, on se trouve dans la situation représentée en figure 6.4. Soit alors $\mathscr{D} - \{x_n\}$ l'ensemble des données aux feuilles de $\mathscr{T}$, privé de la feuille n. Dans la suite nous remplaçons d'ailleurs toujours l'ensemble des données $\mathscr{D}$ par la différence ensembliste $\mathscr{D} - \{x_n\}$, qui est simplement égale à $\mathscr{D}$ lorsque n n'est pas une feuille.

De façon similaire à ce qu'indiquent les équations (6.2) et (6.3), on a dans ce cas de figure (un peu comme si l'on faisait de n la racine de l'arbre $\mathscr{T}$) :

$$\mathrm{Lk}(\mathscr{T}|\mathscr{D} - \{x_n\}, Q, n = \alpha) = \Pr(\alpha \xrightarrow{l} \beta)\, \mathrm{Lk}(\mathscr{S}|\mathscr{D} - \{x_n\}, Q, v = \beta) \tag{6.4}$$

Maintenant que nous avons établi comment calculer $\mathrm{Lk}(\mathscr{T}|\mathscr{D} - \{x_n\}, Q, n = \alpha)$ selon la position du nœud n et en utilisant l'algorithme d'élagage de Felsenstein, retournons un instant en arrière pour voir d'où nous venons, et regardons de nouveau le schéma général : il s'agit de déterminer les paramètres du HMM profil correspondant au point n de l'arbre

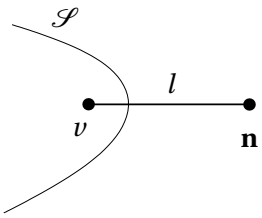


Figure 6.4. *L'arbre phylogénétique $\mathscr{T}$ non raciné dans lequel le nœud d'intérêt n se trouve être une feuille.*

phylogénétique $\mathscr{T}$ qui lie entre elles les séquences d'apprentissage.

Il faut remarquer que chacun de ces paramètres θ (par exemple les émissions sur le troisième état Match du HMM profil ou encore ses transitions sortantes) se présente sous la forme d'une *distribution de probabilités* sur un ensemble de caractères appelé alphabet, que nous noterons $\mathscr{A}$. On a donc $\theta = [\theta_i]_{i\in[1,|\mathscr{A}|]}$ avec $\forall i\ \theta_i \in [0,1]$ et $\sum_i \theta_i = 1$. Dans les deux exemples auxquels il est fait allusion plus haut, $\mathscr{A}$ désigne respectivement l'ensemble des vingt acides aminés et l'ensemble des trois transitions M→M, M→D et M→I. La distribution de probabilités recherchée pour paramétrer le HMM nous sera donnée directement par les probabilités $\Pr(n = \alpha|\mathscr{D} - \{x_n\}, \mathscr{T}, Q)$ lorsque α décrit $\mathscr{A}$.

Comment obtenir ces « probabilités postérieures » $\Pr(n = \alpha|\mathscr{D} - \{x_n\}, \mathscr{T}, Q)$ à partir des vraisemblances $\mathrm{Lk}(\mathscr{T}|\mathscr{D} - \{x_n\}, Q, n = \alpha)$? En appliquant le théorème de Bayes, on écrit [1] :

$$\Pr(n = \alpha|\mathscr{D} - \{x_n\}, \mathscr{T}, Q) = \frac{\Pr(\mathscr{D} - \{x_n\}|\mathscr{T}, Q, n = \alpha)\Pr(\alpha|Q)}{\Pr(\mathscr{D} - \{x_n\}|\mathscr{T}, Q)} \tag{6.5}$$

$$= \frac{\Pr(\mathscr{D} - \{x_n\}|\mathscr{T}, Q, n = \alpha)\Pr(\alpha|Q)}{\sum_\gamma \Pr(\mathscr{D} - \{x_n\}|\mathscr{T}, Q, n = \gamma)\Pr(\gamma|Q)} \tag{6.6}$$

Exprimée en termes de vraisemblances d'arbre et en notant π_Q la distribution d'équilibre propre au processus Q, l'équation (6.5) devient :

$$\Pr(n = \alpha|\mathscr{D} - \{x_n\}, \mathscr{T}, Q) = \frac{\mathrm{Lk}(\mathscr{T}|\mathscr{D} - \{x_n\}, Q, n = \alpha)\pi_Q(\alpha)}{\sum_\gamma \mathrm{Lk}(\mathscr{T}|\mathscr{D} - \{x_n\}, Q, n = \gamma)\pi_Q(\gamma)} \tag{6.7}$$

C'est précisément le résultat recherché, qui exprime les probabilités a posteriori pour les différents caractères α au nœud n, lesquelles probabilités vont directement former la

1. Dans ces deux équations il faut considérer que $\mathscr{T}$ et Q sont toujours en partie droite des probabilités conditionnelles, mais on omet simplement de les mentionner lorsqu'ils n'ont aucune influence sur la partie gauche.

distribution $[\theta_i]_{i\in[1,|\mathscr{A}|]}$: si α_i est le i^{e} caractère de l'alphabet $\mathscr{A}$, alors on définit le paramètre θ_i, qui est une probabilité, de la façon suivante :

$$\theta_i = \Pr(n = \alpha_i | \mathscr{D} - \{x_n\}, \mathscr{T}, Q) = \frac{\mathrm{Lk}(\mathscr{T} | \mathscr{D} - \{x_n\}, Q, n = \alpha_i)\pi_Q(\alpha_i)}{\sum_{j=1}^{|\mathscr{A}|} \mathrm{Lk}(\mathscr{T} | \mathscr{D} - \{x_n\}, Q, n = \alpha_j)\pi_Q(\alpha_j)} \tag{6.8}$$

Ainsi, les paramètres du HMM profil correspondant à un nœud n donné sont le résultat d'un processus de *reconstruction ancestrale* en ce nœud. Dans la suite, nous détaillons ce processus de détermination des paramètres en examinant tour à tour ce qui se passe concrètement pour chacune des classes de paramètres.

Rappelons qu'il s'agit de construire autant de HMM profils qu'il y a de nœuds dans l'arbre phylogénétique reliant les séquences d'apprentissage ($2N-2$ nœuds pour un arbre non raciné sur N taxa). Dans toute la suite de ce chapitre, nous décrivons le processus de construction du HMM profil correspondant à un nœud n quelconque dans cet arbre.

CHAPITRE

7 Architecture du HMM

Dans ce chapitre, nous nous penchons sur la détermination *basée sur la phylogénie* des positions d'intérêt lorsque l'on veut modéliser les séquences ancestrales dans un voisinage donné. La structure des HMM profils étant déterminée à l'avance (voir par exemple en figure 3.3), la détermination de la taille du modèle (c'est-à-dire du nombre de nœuds du HMM profil) ainsi que la mise en correspondance de chacun de ses nœuds avec une colonne de l'alignement d'apprentissage, sont cruciales pour la suite du processus. En effet, seules les colonnes en question font l'objet d'une modélisation fine (par des états Match), les états d'insertion étant formés, eux, à partir de l'agrégation de plusieurs colonnes de l'alignement d'entrée.

Sommaire

7.1 Problématique de sélection des colonnes à modéliser en fonction de la position phylogénétique

Nous l'avons dit, la structure du HMM profil est fixée. Nous nous conformons ici à celle qu'utilise la suite logicielle de création et d'utilisation de HMM profils la plus populaire, à savoir HMMER (cf. section 3.3.4). Nous nous intéressons d'emblée uniquement au cœur des modèles suivant l'architecture Plan7, et non aux éventuels états de bouclage N, C et J qui permettent de repérer plusieurs occurrences du même modèle au sein des séquences cibles.

La tâche qui nous incombe dans le présent chapitre consiste donc uniquement à déterminer quelles seront les colonnes de l'alignement d'apprentissage qui feront par la suite l'objet d'une modélisation par des états Match. Comment effectuer ce choix, sachant que l'objectif est de concevoir un HMM profil qui soit pertinent pour le voisinage phylogénétique du nœud n de l'arbre ? Pour tenter de répondre efficacement à cette question, prenons l'exemple d'alignement présenté en figure 7.1 accompagné de sa phylogénie support. Il s'agit d'un alignement de séquences extraites de la superfamille a.1.1 (globines et apparentées) de la hiérarchie ASTRAL/SCOP. L'alignement a été produit par le logiciel PRANK [Löytynoja et Goldman, 2005, 2008] et l'arbre phylogénétique a ensuite été calculé avec PHYML [Guindon et Gascuel, 2003]. On remarque qu'il existe une corrélation relativement bonne entre la position d'une séquence dans l'arbre phylogénétique et les motifs de lettres et de gaps que présente la séquence. Par exemple, dans la figure 7.1, les cinq séquences médianes présentant des caractères entre les colonnes assez bien conservées d'acides aminés majoritairement aromatiques à gauche (essentiellement F et W) et ATGD à droite, forment un clade. De même pour celles qui ne présentent pas d'acide aminé A ou E juste avant la colonne d'aromatiques, et qui n'expriment pas non plus les deux colonnes consensuelles tout à fait à gauche de l'alignement. Ainsi, si l'on souhaite construire un HMM profil adapté à un endroit précis de la phylogénie, il serait logique de tenir compte de ces corrélations entre position phylogénétique et motif lettres/gaps, pour parvenir *in fine* à modéliser par des états Match les positions de l'alignement qui sont pertinentes *pour le voisinage phylogénétique d'un nœud donné*.

Par exemple, et pour reprendre l'exemple précédent, on pourrait avoir comme objectif de produire pour le nœud n l'architecture correspondant aux astérisques rouges au bas de la figure 7.2, alors que le HMM standard établi sur l'alignement produirait plutôt l'architecture correspondant aux astérisques de couleur noire, toujours sur la même figure 7.2. Nous exposons dans la section qui suit les voies et moyens pour parvenir à cet objectif de modélisation.

Pour rentrer dans le cadre général présenté en section 6, il nous faut (et ce parcours sera structuré de même pour la détermination des paramètres ultérieurs) :

1. établir l'alphabet $\mathscr{A}$ des caractères sur lesquels nous allons travailler,

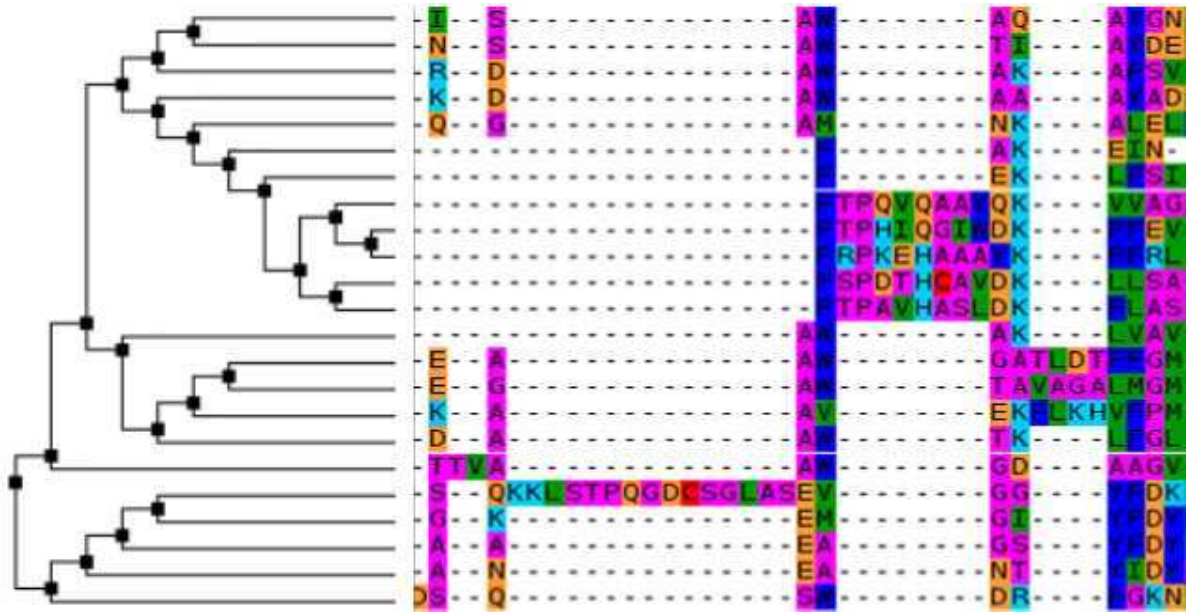


Figure 7.1. *Un alignement de globines et de protéines apparentées (superfamille ASTRAL/-SCOP a.1.1) représenté avec l'arbre phylogénétique calculé au maximum de vraisemblance.*

2. produire à partir de l'alignement des séquences protéiques d'apprentissage l'alignement correspondant de séquences sur l'alphabet $\mathscr{A}$,
3. définir le processus substitutionnel Q qui nous permettra ensuite de calculer des probabilités de substitution entre éléments de $\mathscr{A}$ le long des branches de l'arbre,
4. calculer les vraisemblances d'arbre selon les différentes valeurs de caractère prises par le nœud n pour établir la distribution de probabilités postérieures sur $\mathscr{A}$ au point n (cf. section 6),
5. enfin, dériver la décision de sélection d'une colonne en tant qu'état Match du HMM (décision Oui/Non) à partir de la distribution établie à l'étape précédente.

7.2 Choix d'un alphabet et construction de l'alignement correspondant

Pour traiter les deux premiers points de la liste de tâches ci-dessus, nous sommes partis de l'examen de la façon dont le programme *hmmbuild* de la suite HMMER détermine l'architecture d'un modèle à partir d'un alignement protéique qui lui est fourni en entrée. La méthode par défaut de HMMER 3.0 consiste, après pondération des séquences, à prendre la décision de modéliser une colonne de l'alignement comme un état Match si et

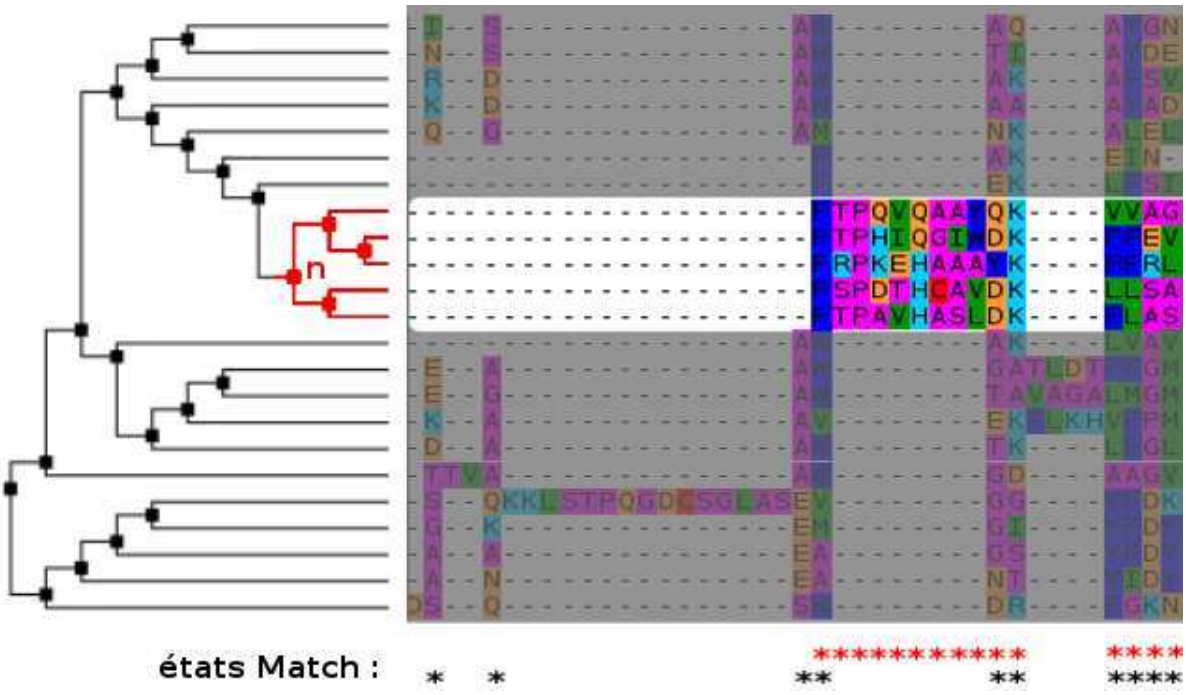


Figure 7.2. *L'alignement de la figure 7.1. On a mis l'accent sur le sous-arbre de racine n, représenté en rouge. Sous l'alignement, les deux lignes d'astérisques donnent d'une part l'architecture de HMM profil souhaitable pour la reconstruction ancestrale en n (en rouge) et d'autre part l'architecture produite par défaut par un logiciel comme* HMMER *pour tout l'alignement (en noir).*

seulement si la proportion de lettres que contient cette colonne (par rapport au nombre de séquences alignées) dépasse un certain seuil, par défaut 0,5 (cf. section 3.3.4). Lors de cette étape comme ailleurs, HMMER ignore l'information phylogénétique contenue dans l'alignement. Puisque nous voulons justement la prendre en compte, alors il faut que la décision prise dans le cadre de la construction d'un HMM profil pour le nœud n s'appuie d'abord sur le voisinage phylogénétique du point n. Au comptage « aveugle » du nombre de lettres dans la colonne (après une pondération faite une fois pour toutes et tendant à renforcer le poids des séquences singulières, voir section 3.4), nous voulons substituer une analyse phylogénétique basée sur la présence ou l'absence de lettres (acides aminés) aux feuilles de l'arbre et pour la colonne considérée.

Nous choisissons donc un alphabet binaire $\mathscr{A} = \{G, N\}$ dans lequel la lettre G signifiera « gap » tandis que N sera son complémentaire « non gap », indiquant la présence d'une lettre. L'alignement sur lequel nous allons calculer les vraisemblances d'arbre se déduit directement de l'alignement des séquences protéiques formant l'ensemble d'apprentissage

des modèles, en remplaçant dans celui-ci tout acide aminé par la lettre N et tout gap par la lettre G. À la fin du processus de détermination de l'architecture, pour toute colonne de l'alignement d'entrée on obtiendra par reconstruction ancestrale au nœud n une distribution de probabilités sur $\mathscr{A}$ à partir de laquelle nous prendrons la décision de modéliser ou non la colonne en question par un état Match dans le HMM correspondant au nœud n de la phylogénie (voir figure 7.3).

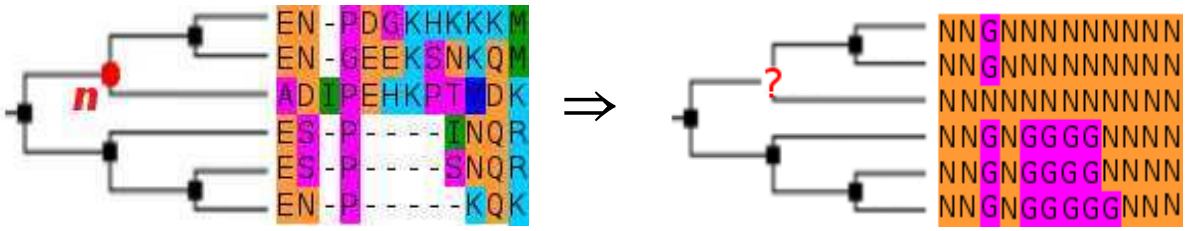


Figure 7.3. *Passage d'un alignement d'acides aminés à un alignement sur un alphabet binaire Gap/Nongap*

7.3 Quel processus substitutionnel pour les motifs Gap/Lettre ?

On fait l'hypothèse que les alignements de caractères binaires {G,N} directement issus des alignements correspondants d'acides aminés, sont une issue statistiquement probable de l'action d'un processus de substitution Q agissant le long des branches de l'arbre phylogénétique (construit par exemple par une approche de maximum de vraisemblance sur les séquences protéiques d'apprentissage). Le processus Q en question doit donc être cohérent avec les couples { arbre, alignement sur $\mathscr{A}$ } que l'on peut rencontrer lorsque l'on aligne des séquences protéiques homologues. D'où l'idée d'*apprendre* Q sur les jeux de données dont nous disposons.

Avant d'aller plus loin, regardons un instant quelle forme va prendre ce processus de substitution, en décrivant son générateur (ou matrice des taux instantanés) Q. Q se présente sous la forme d'une matrice 2×2. Nous convenons de l'ordre dans lequel nous énonçons notre alphabet : d'abord G puis N. On lira donc la matrice Q comme suit :

$$Q = \begin{pmatrix} q_{GG} & q_{GN} \\ q_{NG} & q_{NN} \end{pmatrix}$$

Les sommes sur les lignes devant nécessairement être égales à 0 pour qu'on ait bien une matrice de taux instantanés cohérente, on peut simplifier l'écriture en réduisant le

nombre de paramètres :

$$Q = \begin{pmatrix} -q_{GN} & q_{GN} \\ q_{NG} & -q_{NG} \end{pmatrix}$$

On a ici nécessairement $q_{GN} > 0$ et $q_{NG} > 0$. De plus, si l'on fait l'hypothèse de la stationnarité du processus dont Q est le générateur, alors il existe une distribution d'équilibre sur les lettres G et N, laquelle distribution nous nommons π. On écrit $\pi = (\pi_G, \pi_N)$ et comme il s'agit d'une distribution de probabilités, $\pi = (\pi_G, 1-\pi_G)$. Dans la suite, on pourra également faire appel à la matrice carrée diagonale Π portant les éléments de la distribution π :

$$\Pi = \begin{pmatrix} \pi_G & 0 \\ 0 & 1-\pi_G \end{pmatrix}$$

Nous imposons une contrainte supplémentaire à Q : nous souhaitons travailler avec un processus qui soit temps-réversible, afin que la position de la racine dans une phylogénie n'ait aucune influence sur les vraisemblances calculées (cf. section 4.4). Cette contrainte de réversibilité s'écrit $\Pi Q = 0$ et impose $\pi_G q_{GN} = \pi_N q_{NG}$. En tenant compte de cette contrainte, on peut réécrire Q pour faire disparaître les termes q_{NG} :

$$Q = \begin{pmatrix} -q_{GN} & q_{GN} \\ \frac{\pi_G q_{GN}}{1-\pi_G} & -\frac{\pi_G q_{GN}}{1-\pi_G} \end{pmatrix} = q_{GN} \begin{pmatrix} -1 & 1 \\ \frac{\pi_G}{1-\pi_G} & -\frac{\pi_G}{1-\pi_G} \end{pmatrix} \tag{7.1}$$

Enfin, il est d'usage en phylogénie de travailler avec des processus dont le générateur Q est normalisé, pour que le déroulement du processus le long d'une branche de longueur 1 corresponde toujours à une espérance d'événement de substitution aussi égale à 1. Cette contrainte de normalisation s'exprime de la façon suivante, qui emploie l'opérateur de trace d'une matrice (somme de ses valeurs diagonales) : $\mathrm{tr}(\Pi Q) = -1$. On peut remarquer dans l'équation (7.1) que le paramètre q_{GN} se dégage déjà comme un facteur d'échelle pour Q. Le calcul de la trace de ΠQ donnant $\mathrm{tr}(\Pi Q) = -2\pi_G q_{GN}$, la contrainte de normalisation s'écrit :

$$q_{GN} = \frac{1}{2\pi_G} \tag{7.2}$$

Cette dernière contrainte enlève encore un paramètre libre au générateur Q, que l'on peut finalement écrire en fonction du seul paramètre π_G :

$$Q = \frac{1}{2} \begin{pmatrix} -\pi_G^{-1} & \pi_G^{-1} \\ (1-\pi_G)^{-1} & -(1-\pi_G)^{-1} \end{pmatrix} \tag{7.3}$$

Enfin, rappelons-nous que les processus temps-réversibles peuvent également être décrits en fonction d'une demi-matrice recensant les *échangeabilités* entre caractères. Ici il n'y a qu'un terme d'échangeabilité : $R_{G \leftrightarrow N} = \frac{q_{GN}}{\pi_N} = \frac{q_{NG}}{\pi_G} = \frac{1}{2\pi_G(1-\pi_G)}$.

Pour apprendre ce seul paramètre π_G, nous pouvons simplement compter la proportion de gaps dans l'alignement d'entrée transformé en un alignement sur l'alphabet $\mathscr{A} = \{G, N\}$, puis faire directement de ce ratio un estimateur pour π_G. Dès lors que l'alignement d'entrée n'est pas réduit à une poignée de sites, cet estimateur peut être considéré comme fiable, et le processus de générateur Q est dès lors parfaitement déterminé.

7.4 Questions de temps : quelles longueurs de branches, quelle variabilité des vitesses d'évolution ?

Q étant désormais défini, il est temps d'examiner l'arbre sur lequel nous allons le faire « travailler ». Rappelons que nous supposons un arbre $\mathscr{T}$ donné en entrée de tout le processus de phylogénisation. Il peut être créé en amont par tout moyen jugé adéquat, mais en tout état de cause il l'aura sans doute été à partir de l'alignement des séquences protéiques. Tout du moins, on suppose ici que cet arbre $\mathscr{T}$ est en cohérence avec l'alignement des séquences d'apprentissage. En particulier, les longueurs de branche de cet arbre sont probablement adaptées à un processus de substitution entre acides aminés (WAG, JTT, LG, etc). Pour faire au mieux il faudrait réévaluer ces longueurs de branches pour construire $\mathscr{T}'$ à partir de $\mathscr{T}$, mais le faire serait à la fois coûteux et, si l'estimation est réalisée sur la base d'un seul alignement Gap/NonGap, peu fiable. Nous choisissons donc d'optimiser un unique paramètre d'échelle λ. L'arbre qu'on utilisera par la suite pour calculer les vraisemblances sur les caractères binaires de l'alphabet $\mathscr{A} = \{G, N\}$ sera donc $\mathscr{T}' = \lambda\mathscr{T}$ (même topologie mais longueurs de branches toutes multipliées par le facteur λ). L'optimisation de ce paramètre d'échelle se fait par la méthode de Brent [Press *et al.*, 2007] : on maximise la vraisemblance de l'arbre $\mathscr{T}'$ sur les données Gap/NonGap issues de l'alignement d'entrée en utilisant le processus Q établi à l'étape précédente.

Comme le calcul ultérieur des vraisemblances se fera avec une loi Gamma discrète à quatre classes de vitesse (cf. section 4.5), il nous faut également estimer le paramètre de forme α de cette loi. La boucle d'optimisation présentée ci-dessus inclut donc deux étapes à chaque itération, utilisant toutes deux la méthode de Brent :

1. optimisation du facteur d'échelle λ de l'arbre,
2. optimisation du paramètre de forme α de la loi Gamma à quatre classes de vitesse.

Les itérations prennent fin lorsque d'une itération à l'autre la différence des vraisemblances calculées se trouve en deçà d'un seuil fixé (e.g. 10^{-5} en log-vraisemblance).

7.5 Calcul des vraisemblances avec contrainte au nœud d'intérêt

Pour calculer l'architecture du HMM correspondant au nœud n de la phylogénie, on commence par calculer les vraisemblances sur les données Gap/NonGap en fixant le caractère porté par le nœud n comme exposé dans le chapitre 6. En rappelant qu'ici $\mathscr{A} = \{G, N\}$, que par conséquent les caractères sont $\alpha_1 = G$ et $\alpha_2 = N$, on réécrit l'équation (6.8) de la façon suivante :

$$\forall i \in \{1,2\} \ \Pr(n = \alpha_i | \mathscr{D} - \{x_n\}, \mathscr{T}, Q) = \frac{\mathrm{Lk}(\mathscr{T} | \mathscr{D} - \{x_n\}, Q, n = \alpha_i) \pi_Q(\alpha_i)}{\sum_{j=1}^{2} \mathrm{Lk}(\mathscr{T} | \mathscr{D} - \{x_n\}, Q, n = \alpha_j) \pi_Q(\alpha_j)} \tag{7.4}$$

Ainsi pour chaque nœud n et pour chaque colonne $\mathscr{D}$ de l'alignement d'apprentissage, on obtient une distribution de probabilités à deux termes, $\Pr(n = G | \mathscr{D} - \{x_n\}, \mathscr{T}, Q)$ et $\Pr(n = N | \mathscr{D} - \{x_n\}, \mathscr{T}, Q)$. Cette distribution nous indique combien il est probable ($\Pr(n = N | \ldots)$) que la reconstruction ancestrale au nœud n apporte une séquence dont la position correspondante est remplie par un acide aminé. C'est de cette distribution que nous allons nous servir pour déterminer si la position en question sera modélisée ou non par un état Match dans le HMM profil correspondant au nœud n.

7.6 Décision de sélection des colonnes pour construire l'architecture du HMM profil

La règle de décision peut être la suivante : dès lors que la probabilité a posteriori d'avoir le caractère N sur une certaine colonne est supérieure ou égale à 0,5, on choisit de modéliser la colonne en question par un état Match. Cette première règle de décision a le mérite d'être simple. On peut aussi choisir un seuil qui soit plus adapté à la typicité de l'alignement d'apprentissage. Par exemple, ce seuil peut être donné directement par la proportion d'acides aminés (le restant étant constitué par les gaps) dans l'alignement d'entrée :

$$\text{colonne } \mathscr{D} \text{ sélectionnée} \quad \Leftrightarrow \quad \Pr(n = N | \mathscr{D} - \{x_n\}, \mathscr{T}, Q) \geq \pi_Q(N) \tag{7.5}$$

En effet, on rappelle que le processus Q est par construction tel que sa distribution d'équilibre π_Q est en accord avec la composition de l'alignement d'apprentissage.

7.7 Conclusion concernant les architectures des modèles

Les étapes décrites dans ce qui précède mènent à la construction d'une série de HMM dont l'architecture dépend de la position dans l'arbre sur laquelle ces HMM sont calculés. Ceci constitue une première différence majeure avec [Qian et Goldstein, 2003, 2004], dans laquelle l'architecture de chacun des HMM de reconstruction ancestrale est la même, calculée sans pondération des séquences à partir de la simple heuristique qui consiste à sélectionner une colonne pour modéliser un état Match dès lors que celle-ci contient plus de 50% de caractères. Cette différence majeure doit nous permettre de concevoir des modèles plus divers, plus à même de repérer des séquences homologues dégénérées à partir du moment où les dégénérescences en question (ablation d'un ou de plusieurs domaine(s), introduction de répétitions, etc.) sont cohérentes du point de vue d'une analyse phylogénétique.

CHAPITRE

8 Émissions de caractères sur les états Match du modèle

Ce que nous présentons dans ce chapitre est connu et a déjà été implémenté, notamment dans les *tree-HMM* de Qian et Goldstein [Qian et Goldstein, 2003, 2004]. Nous ne nous étendons donc pas sur le sujet, mais proposons au lecteur de se référer à la section 5.3.1 pour plus d'informations quant à ces travaux.

Néanmoins, nous précisons la démarche que nous avons adoptée. Elle consiste à construire pour chaque nœud de la phylogénie et pour chaque colonne sélectionnée par l'étape de construction de l'architecture pour former un état Match *pour le HMM de reconstruction ancestrale au nœud n*, une distribution d'acides aminés émis calculée en fonction de la phylogénie et des caractères aux feuilles. L'alphabet $\mathscr{A}$ étant celui des vingt acides aminés, nous calculons :

$$\forall i \in [1,20] \ \Pr(n = \alpha_i | \mathscr{D} - \{x_n\}, \mathscr{T}, Q) = \frac{\mathrm{Lk}(\mathscr{T} | \mathscr{D} - \{x_n\}, Q, n = \alpha_i) \pi_Q(\alpha_i)}{\sum_{j=1}^{20} \mathrm{Lk}(\mathscr{T} | \mathscr{D} - \{x_n\}, Q, n = \alpha_j) \pi_Q(\alpha_j)} \tag{8.1}$$

Ces vingt probabilités postérieures donnent directement les vingt paramètres du HMM profil que sont les probabilités d'émission de caractères sur l'état Match considéré (voir figure 8.1).

L'arbre $\mathscr{T}$ est celui qui est donné en entrée par l'utilisateur. Comme on suppose qu'il a été obtenu à partir de l'alignement de séquences protéiques lui-même, on ne prend pas la peine de calculer un facteur multiplicatif sur les longueurs de branche, et on l'utilise tel quel. En revanche, les calculs de vraisemblances mettent en jeu une loi Gamma. Comme précédemment, on choisit une loi à quatre classes de vitesse, et on estime le paramètre de forme de cette loi Gamma à partir de l'alignement des séquences protéiques. Le processus substitutionnel Q est le processus LG décrit par Le et Gascuel [Le et Gascuel, 2008]. Il faut

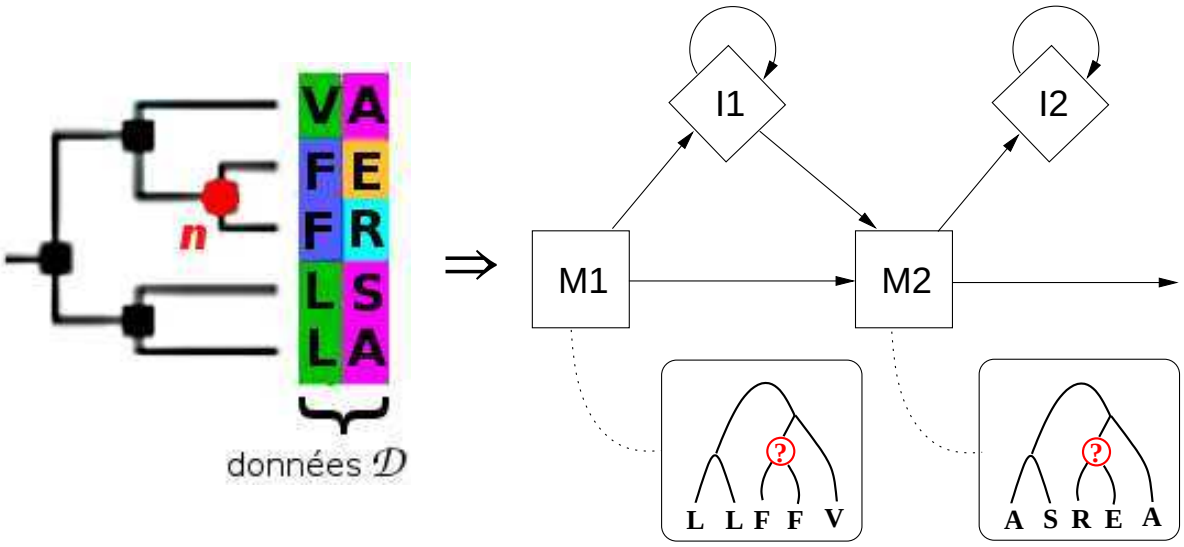


Figure 8.1. *Cadre de la dérivation des probabilités d'émission sur les états Match. Exemple de deux sites consécutifs issus d'un alignement donné, avec la phylogénie correspondante. On suppose que chacun des deux sites est modélisé par un état Match.*

noter là deux différences avec le travail de Qian et Goldstein [Qian et Goldstein, 2003, 2004], puisque ces auteurs utilisaient le processus WAG de Whelan et Goldman [Whelan et Goldman, 2001] et ne parlaient pas de loi Gamma pour modéliser la variabilité des vitesses d'évolution.

Ainsi, on obtient pour chaque nœud de la phylogénie et pour chaque état Match correspondant à ce nœud, une distribution d'émissions d'acides aminés qui tient compte non seulement des observations faites sur la colonne correspondante de l'alignement (comme le ferait un simple HMM profil), mais aussi des *relations phylogénétiques* entre les différents caractères observés ainsi que de la *position phylogénétique de la cible de reconstruction* (ici le nœud n).

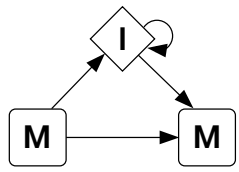
CHAPITRE

9 Transitions quittant les états Match et les états de Délétion

Nous abordons ici l'étape marquée (3) dans la figure 6.2. Il s'agit de déterminer d'un point de vue phylogénétique les différentes probabilités affectées aux transitions entre les états du HMM profil. Le but est d'atteindre une plus grande pertinence dans la mise au point des HMM de reconstruction ancestrale, lorsque les séquences phylogénétiquement proches ont tendance à partager les mêmes motifs d'insertion et de délétion, c'est-à-dire à emprunter les mêmes transitions dans un HMM de référence.

Sommaire

Nous écartons d'emblée les transitions ayant pour origine les états Insertion du HMM, pour nous y intéresser plus tard (cf. chapitre 10). En effet, ainsi que l'ont remarqué les auteurs s'étant penchés sur la question des tree-HMM [Mitchison et Durbin, 1995; Mitchison, 1999; Qian et Goldstein, 2003, 2004], ces transitions présentent un problème incontournable lorsque l'on essaie de les intégrer dans une phylogénie : le fait que les états d'Insertion bouclent sur eux-mêmes implique l'impossibilité d'aligner ces transitions de manière classique, c'est-à-dire en déterminant des ensembles de caractères dérivant du même caractère ancestral. On peut par exemple consulter la figure 9.1 pour s'en rendre compte : à partir d'un alignement de séquences protéiques, on obtient un ensemble de séquences de transitions dont les membres ne partagent pas la même longueur.



	X – – X	M→M
	X – X X	M→I ; I→M
	X X X X	M→I ; I→I ; I→M
colonnes match	* *	

Figure 9.1. *On représente à gauche un ensemble de séquences protéiques où la lettre X dénote un acide aminé tandis que '-' représente un gap. En haut, la portion du HMM profil qui nous intéresse pour modéliser cet ensemble de séquences (architecture Plan7 de* HMMER*). À droite, le matériel à aligner en ce qui concerne les transitions dans le modèle est constitué par les chemins que suivent les différentes séquences, étant donné l'alignement et l'étiquetage en colonnes Match. On doit alors aligner des séquences de transition de longueurs différentes, ce qui illustre la difficulté posée par les transitions provenant de l'état d'Insertion.*

9.1 Quel sens y a-t-il à aligner des transitions ?

Si l'on veut étudier d'un point de vue phylogénétique les transitions empruntées par une famille de séquences dans un HMM profil, on doit avant tout se poser la question de les aligner les unes avec les autres. En effet, ce n'est qu'après avoir déterminé l'alignement de transitions correspondant à un ensemble de séquences qu'on pourra évaluer la vraisemblance d'un arbre phylogénétique reliant ces transitions. Quel sens a ce travail ?

Il faut d'abord avoir clairement à l'esprit qu'il faut plus qu'un alignement de séquences (d'acides aminés ou de nucléotides) pour déterminer l'alignement de transitions corespondant. L'ingrédient supplémentaire est indispensable, et il s'agit de l'étiquetage de l'alignement de séquences en termes de colonnes match. Cette remarque peut sembler évidente mais on a trop vite tendance à l'oublier : l'alignement de transitions est intrinsèquement lié à l'annotation des séquences, et deux annotations différentes du même alignement de séquences produisent deux alignements de transitions différents. Ainsi, on ne saurait être trop prudent en interprétant mécaniquement les états empruntés dans le HMM profil comme étant à mettre en correspondance avec des événements évolutifs : le passage d'une certaine séquence s par un état de Délétion ne signifie pas nécessairement que s est la séquence résultant d'un événement évolutif de délétion d'un ou plusieurs caractères. Plus prosaïquement, le passage de s par cet état de délétion signifie simplement

que la séquence s n'exprime pas d'acide aminé sur cette colonne qu'on a identifié (plus ou moins arbitrairement) comme étant une colonne « conservée ». En fait, la traduction d'une séquence biologique en une séquence d'états du HMM profil se fait automatiquement en examinant le statut de la colonne courante, selon la grille de lecture suivante :

		statut de la colonne courante	
		match	non match
caractère présenté	lettre	état M	état I
	gap	état D	(néant)

Le résultat d'une telle interprétation est illustré en figure 9.2.

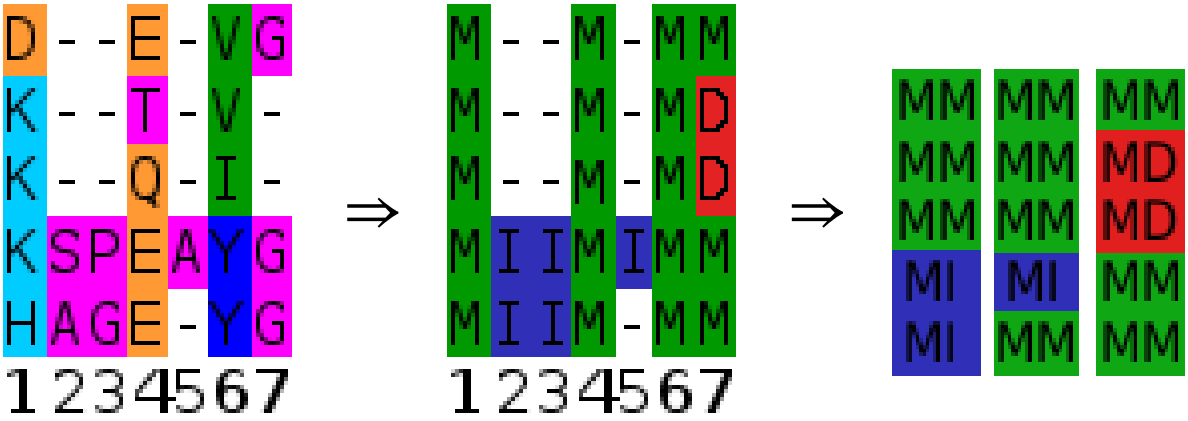


Figure 9.2. *Construction d'un alignement de transitions à partir d'un alignement d'acides aminés et d'un étiquetage de celui-ci en colonnes match et non match. À gauche, l'alignement d'acides aminés. On suppose que les colonnes nº 1, 4, 6 et 7 ont été désignées comme étant des colonnes « match ». Au milieu, l'alignement d'états caché du HMM qui découle de cet étiquetage. À droite, les transitions quittant les colonnes match nº 1, 4 et 6. Il se trouve qu'ici les transitions alignées sont toutes de type M→, mais tel ne serait pas le cas si les colonnes nº 1, 3 et 6 contenaient au moins un gap dans l'alignement de gauche.*

9.2 Quel alphabet ?

En prenant en compte ce qui vient d'être exposé, il nous faut donc dans ce chapitre calculer les transitions ayant pour origine un état Match et celles partant d'un état de Délétion. Les caractères composant l'alphabet $\mathscr{A}$ seront donc les suivants :

- M→M, la transition d'un état Match du HMM à l'état Match suivant, en abrégé MM,
- M→I, la transition d'un état Match du HMM à l'état d'Insertion du même nœud, en abrégé MI,

- M→D, la transition d'un état Match du HMM à l'état de Délétion du nœud suivant, en abrégé MD,
- D→D, la transition d'un état de Délétion du HMM à l'état de Délétion du nœud suivant, en abrégé DD,
- D→M, la transition d'un état de Délétion du HMM à l'état Match du nœud suivant, en abrégé DM,
- D→I, la transition d'un état de Délétion du HMM à l'état d'Insertion du même nœud, en abrégé DI.

On rappelle que la dernière de ces transitions, DI, n'existe que dans les HMM à l'architecture « historique » proposée par Krogh et Haussler et implémentée dans le logiciel SAM. Les HMM conçus et manipulés par HMMER ne font pas apparaître ces transitions. Dans le présent travail, nous adoptons la stratégie consistant à conserver cette transition DI dans nos modèles et dans tous les calculs de vraisemblance, pour finalement ne l'abandonner que lorsqu'il s'agit d'écrire finalement le HMM au format HMMER. Cet abandon de la transition DI et de la probabilité afférente $\Pr(DI)$ se fait alors en normalisant :

$$\Pr_{\text{HMMER}}(DD) = \frac{\Pr(DD)}{\Pr(DD)+\Pr(DM)} \quad \text{et} \quad \Pr_{\text{HMMER}}(DM) = \frac{\Pr(DM)}{\Pr(DD)+\Pr(DM)} .$$

9.3 Quel(s) processus de substitution ?

Les travaux concernant les tree-HMM dont la paternité revient à Graeme Mitchison et Richard Durbin [Mitchison et Durbin, 1995; Mitchison, 1999; Qian et Goldstein, 2003, 2004] ont systématiquement fait le choix de *séparer* les processus substitutionnels affectant les transitions, en mettant d'un côté les transitions MM, MI et MD, et de l'autre les transitions DD, DM et DI (cf. sections 5.1.2 et 5.1.3). Ce choix obéit à une logique qui consiste à concevoir comme événements évolutifs élémentaires les seules substitutions de transitions qui consistent à modifier la destination d'une arête dans l'automate d'états finis que constitue le HMM profil, sans toucher à son origine. Nous allons examiner ci-dessous un certain nombre d'arguments en faveur de la séparation des processus de substitution en deux classes, puis les arguments en faveur de leur fusion en une seule chaîne (puisque les processus sont des chaînes de Markov en temps continu).

9.3.1 Arguments en faveur de la séparation des processus M→ et D→

Tout d'abord, on peut remarquer que les paramètres en jeu forment du point de vue du HMM profil deux distributions séparées. En figure 9.3 on représente en vert les transitions quittant l'état Match et en rouge celles quittant l'état de Délétion. Les distributions de probabilités correspondantes, $\Pr\left(\begin{smallmatrix} M_{i+1} \\ I_i \\ D_{i+1} \end{smallmatrix} \middle| M_i\right)$ et $\Pr\left(\begin{smallmatrix} M_{i+1} \\ I_i \\ D_{i+1} \end{smallmatrix} \middle| D_i\right)$ sont indépendantes l'une de

l'autre lorsqu'on les considère d'un point de vue opérationnel.

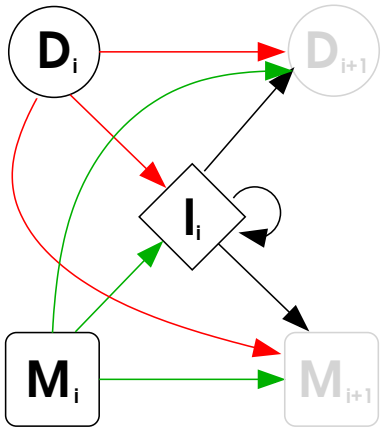


Figure 9.3. *Un nœud i pris au sein d'un HMM profil (les états en grisé appartiennent au nœud suivant). En rouge apparaissent les transitions quittant l'état de Délétion du nœud, en vert celles quittant l'état Match. Les probabilités pondérant ces arêtes forment deux distributions de probabilité disjointes et indépendantes.*

Le deuxième argument en faveur de la séparation des deux processus se situe du côté de l'apprentissage. Si l'on s'attache à décrire les événements évolutifs par des processus réversibles et normalisés (c'est-à-dire dont la trace est égale à -1), alors un processus dont le générateur est de taille 3×3 (par exemple Q_M, le processus concernant les transitions quittant l'état Match) se trouve entièrement décrit par :

- 3 échangeabilités (ici MM↔MI, MM↔MD et MD↔MI),
- 2 paramètres pour la distribution d'équilibre (par exemple ici π_{MM} et π_{MI}), le troisième se déduisant évidemment des deux autres.

À ces cinq paramètres indépendants il faut en retirer un pour prendre en compte la contrainte de normalisation, ce qui ramène à quatre le nombre de paramètres indépendants à déterminer lors de la phase d'apprentissage d'un tel processus.
Si en revanche on décide de fusionner les deux classes M→ et D→, on a un générateur de taille 6×6 et $\sum_{k=1}^{6-1} k + (6-1) - 1 = 19$ paramètres indépendants à apprendre (contre 8 paramètres à déterminer pour deux processus 3×3). La dimension réduite de l'espace des paramètres pour les deux processus envisagés séparément présente donc un avantage computationnel certain.

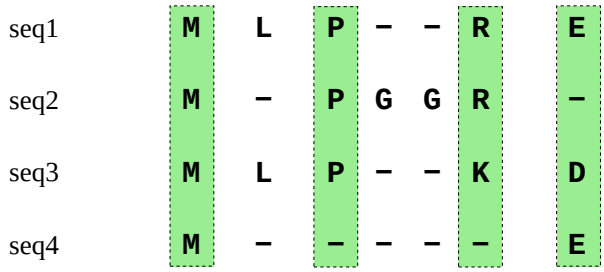


Figure 9.4. *Un exemple d'alignement de séquences protéiques. Les colonnes ombrées de l'alignement forment l'ossature du HMM sous-jacent (colonnes « match »).*

9.3.2 Arguments en faveur de la réunion des processus M→ et D→

Nous avons déjà montré (section 5.1.2) que la conception de deux processus séparés souffrait de nombreux défauts. Reprenons ici l'exemple d'alignement que nous donnions alors, reproduit ici en figure 9.4. Le matériel à aligner en ce qui concerne les transitions empruntées par ces séquences dans le HMM décrit par les colonnes ombrées peut être présenté de la manière suivante, l'astérisque représentant l'absence d'information pertinente :

sortie 1re col. match	1er état ins.	sortie 1er état ins.	sortie 2e col. match	2e état ins.	sortie 2e état ins.	sortie 3e col. match
MI	0 × II	IM	MM	*	*	MM
MM	*	*	MI	1 × II	IM	MD
MI	0 × II	IM	MM	*	*	MM
MD	*	*	DD	*	*	DM

Ainsi, le matériel *potentiellement alignable* en ce qui concerne les transitions peut être partitionné en trois groupes, selon la typologie des colonnes dont proviennent les transitions :

1. transitions au départ d'une colonne match : MM, MI, MD, DM, DI et DD,
2. transitions au départ d'une colonne non match, vers une colonne non match : II,
3. transitions au départ d'une colonne non match, vers une colonne match : IM, ID.

Deux transitions appartenant à un même groupe sont « alignables » entre elles, mais on ne peut avoir deux transitions appartenant à deux groupes différents placées dans la même colonne d'un alignement de transitions. Les guillemets encadrant le terme « alignables » marquent la constatation suivante : les caractères présents dans les zones d'insertion des alignements obtenus grâce aux techniques et aux logiciels d'alignement multiple *ne sont pas* fidèlement alignés les uns aux autres. Ainsi, lorsqu'une séquence donnée présente seulement trois caractères au sein d'une zone d'insertion de longueur

5, la position de ces trois caractères parmi ces cinq positions est décidée arbitrairement. De façon cohérente avec cet arbitraire, le score affecté à ces trois insertions dans le HMM profil correspondant ne changera pas (à condition bien sûr que les cinq positions soient modélisées par un état d'Insertion dans les HMM) : ce score sera égal au produit des probabilités affectées à une transition MI, deux transitions II et une transition IM, également multipliées par les trois probabilités d'émission ad hoc dans l'état d'insertion en question. Ainsi, dire que les transitions II sont alignables les unes aux autres constitue un abus de langage : ces transitions sont à considérer les unes par rapport aux autres, d'une certaine façon que nous exposerons plus loin (section 10).

Scinder artificiellement le premier groupe de transitions en deux groupes distincts présente notamment l'inconvénient d'une perte d'information. Pour illustrer ce fait, prenons l'exemple d'un alignement de trente séquences telles que la colonne de transitions sortant de la première colonne match soit composée de vingt-neuf caractères MM ainsi que d'un caractère DM. Soit f_0 le taxon portant le caractère DM. Une telle colonne de transitions indique manifestement qu'il est très probable, *quel que soit l'endroit où l'on se trouve dans l'arbre*, que le deuxième état Match soit exprimé. En effet, toutes les trente séquences présentent un acide aminé sur la colonne correspondante.

Dans cet exemple, que se passe-t-il pour la reconstruction des probabilités postérieures de transition ? Dans le cas où l'on scinde les deux processus M→ et D→, le M-arbre est construit avec vingt-neuf feuilles présentant une transition MM et la feuille f_0 portant le caractère d'indétermination '*'. Ce caractère induira une probabilité postérieure pour la transition MM qui sera légèrement inférieure à celle reconstruite ailleurs dans l'arbre (toutes les postérieures se trouveront dans l'intervalle $]\pi_{MM}, 1[$). Ce n'est pas très grave, mais la croyance (*statistical belief*) en le fait qu'on exprime le deuxième état Match se trouvera écornée par l'absence totale d'information du point de vue du M-arbre en f_0.

Le phénomène conjugué, en ce qui concerne le D-arbre, est plus grave : cette fois-ci, les feuilles sont toutes indéterminées, à l'exception de f_0 qui porte le caractère DM. La reconstruction phylogénétique autour de f_0 donnera certes une probabilité postérieure importante à DM, mais plus on s'éloignera de la feuille f_0, plus $Pr(DM|\mathscr{T}_D, Q_D)$ tendra vers $\pi_{Q_D}(DM)$, où $\mathscr{T}_D$ est le D-arbre et Q_D le processus concernant les transitions quittant l'état de Délétion.

Ainsi, lorque le premier état Match n'est pas exprimé (état de départ D), la croyance statistique en le fait que l'état Match suivant soit exprimé ignore complètement, dans le cadre de deux processus scindés, l'information importante apportée par toutes les transitions MM observées sur la colonne en question.

Si au contraire on conserve un unique processus pour toutes les transitions quittant les *colonnes* match (et donc les *états* M ou D, cf. tableau en section 9.1), alors logiquement le taux instantané de substitution de MM vers DM sera plus élevé que celui de MM vers DD ou de MM vers DI, et lors de la reconstruction ancestrale la probabilité a posteriori de la transition DM sera rendue sensiblement supérieure à sa fréquence d'équilibre, et ce en tout point de l'arbre grâce aux nombreuses transitions MM aux feuilles. Ainsi, la croyance statistique que le deuxième état Match soit exprimé est accrue du fait de la présence des transitions MM, et ce même si l'on se trouve dans un chemin empruntant le premier état de Délétion. C'est bien ce que l'on souhaitait.

9.3.3 Plaidoyer pour des processus réversibles dans le temps et dans l'espace

Temps-réversibilité

Une vision idéale qui a été un temps celle de Mitchison et Durbin (voir [Mitchison et Durbin, 1995]) consiste à imaginer que les événements de susbtitution entre transitions se produisent le long de l'arbre en partant d'une racine dont la séquence définirait exactement les états Match du HMM. Dans cette vision des choses, tout état Match d'un HMM (évidemment construit sur la base de séquences contemporaines) correspond à un acide aminé qui était exprimé dans la séquence ancestrale, et vice-versa. Dans ce cas, le chemin emprunté par la séquence à la racine dans le HMM n'est fait que de transitions MM. Toujours dans ce cadre, une substitution d'une transition MM par une transition MI correspond bien à un événement phylogénétique d'insertion d'un ou de plusieurs acide(s) aminé(s), tandis qu'une substitution de MM par MD correspond à une délétion. Ce cadre de travail amène [Mitchison et Durbin, 1995] à définir des processus bien particuliers, non réversibles, correspondant à des chaînes de Markov en temps continu non irréductibles (c'est-à-dire qu'il existe des couples d'états (i, j) sans chemin de i vers j) qu'on rappelle en figure 9.5.

Outre le fait que ces processus n'acceptent pas de distribution stationnaire, ils présentent le défaut majeur de ne pas permettre de modéliser l'introduction d'une insertion par rapport à la séquence à la racine, suivie dans l'histoire évolutive d'un certain clade de la suppression de cette insertion pour revenir à la structure en états Match de la séquence originelle : la séquence MM→MI→MM n'y est pas possible.

Au-delà de cette limitation, il nous faut considérer la réalité des HMM profils : la sélection des colonnes conservées se fait en général pour les HMM profils selon une certaine heuristique. Par exemple, HMMER 3.0 sélectionne après pondération des séquences les colonnes pour lesquelles les acides aminés totalisent plus de 50% du poids. Cela revient à sélectionner une colonne à partir du moment où les gaps n'y sont pas majoritaires. Or, *aucune heuristique ne peut nous assurer que l'annotation des colonnes « conservées » dans*

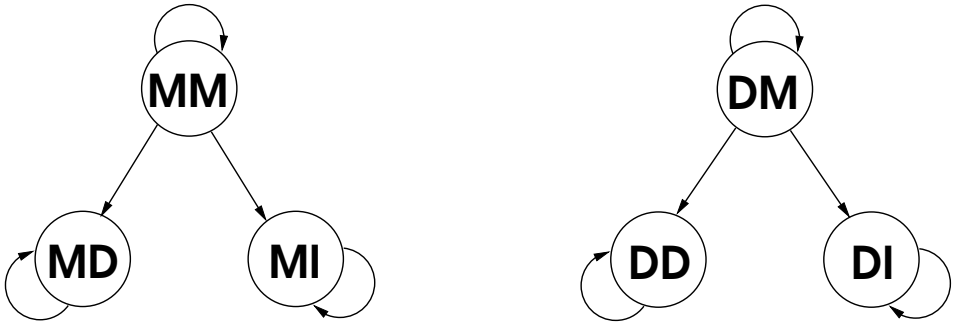


Figure 9.5. *Deux processus de Markov avec contraintes pour les substitutions entre transitions [Mitchison et Durbin, 1995]*

un alignement corresponde aux positions qui existaient dans la séquence à la racine. Ce n'est pas la séquence à la racine (par définition inconnue) mais bien plutôt les séquences aux feuilles qui définissent l'architecture du profil. « Dans la vraie vie », le chemin qu'emprunterait la séquence ancestrale dans le HMM profil, en termes de transitions, n'est pas le chemin singulier fait uniquement de transitions MM, mais *un* chemin comme un autre, dont on pourrait fournir une approximation en reconstruisant la séquence ancestrale.

Ainsi, aucun point de la phylogénie ne se singularise par rapport aux autres lorsqu'il s'agit des transitions : le chemin emprunté par une séquence dans le HMM se définit par rapport à l'architecture choisie, qui elle-même présente un lien avec l'ensemble des séquences aux feuilles. L'histoire évolutive de ces transitions n'est donc *pas orientée*, l'idée d'une racine pour ces processus n'existant tout simplement pas.

Dans ce cadre réaliste, les processus de substitution affectant les transitions ne peuvent qu'être *temps-réversibles*.

Notion nouvelle de réversibilité dans l'espace

L'hypothèse classique faite lorsqu'on calcule des vraisemblances consiste à considérer les différents sites d'un alignement comme évoluant de manière indépendante les uns des autres. Cela permet de calculer la vraisemblance d'un alignement comme un simple produit des vraisemblances de site. Cette indépendance a notamment pour corrolaire que la vraisemblance d'un alignement de séquences écrites en allant de gauche à droite de l'extrémité 5' à l'extrémité 3' (la façon canonique d'écrire des séquences biologiques) est exactement la même que celle correspondant aux séquences écrites toutes dans le sens inverse.

Transposons maintenant ce raisonnement à des alignements de transitions : l'aligne-

ment d'une transition MM contre une transition MD pour les séquences prises dans un premier sens se transforme en un alignement d'une transition MM contre une transition DM lorsqu'on lit les séquences dans l'autre sens. Nous représentons ces deux situations en figure 9.6. En 9.6(a), une transition MD se transforme en une transition MM. En supposant la temps-réversibilité, une écriture possible pour la vraisemblance de cet arbre réduit à une unique branche de longueur l est : $\mathscr{L}_1 = \pi_{MM} \Pr(MM \xrightarrow{l} MD)$. En lisant l'alignement dans l'autre sens, on a affaire à une substitution de DM vers MM (figure 9.6(b)), et l'on écrit : $\mathscr{L}_2 = \pi_{MM} \Pr(MM \xrightarrow{l} DM)$.

(a) sens 5' → 3' (b) sens 3' → 5'

Figure 9.6. *Les mêmes alignements de séquences genèrent des histoires évolutives différentes (au-delà de l'orientation des branches de l'arbre) selon le sens de lecture choisi.*

Si l'on veut respecter la propriété d'égalité entre les vraisemblances calculées dans le sens 5' → 3' et dans le sens inverse, alors il faut que l'égalité $\mathscr{L}_1 = \mathscr{L}_2$ soit respectée quelle que soit la longueur de branche l, en particulier lorsque l tend vers 0. Or

$$e^{Ql} = \sum_{k=0}^{\infty} \frac{(Ql)^k}{k!} \xrightarrow{l \to 0} I + lQ \,,$$

donc :

$$\begin{aligned}
(\forall l > 0 \quad \mathscr{L}_1 = \mathscr{L}_2) \quad &\Rightarrow \quad \left(\forall l > 0 \quad \Pr(MM \xrightarrow{l} MD) = \Pr(MM \xrightarrow{l} DM)\right) \\
&\Rightarrow \quad \left(\forall l > 0 \quad \left[e^{Ql}\right]_{MM,MD} = \left[e^{Ql}\right]_{MM,DM}\right) \\
&\Rightarrow \quad q_{MM,MD} = q_{MM,DM}
\end{aligned}$$

Le même raisonnement que précédemment s'applique lorsqu'on a une transition DD alignée avec une transition DM : dans l'autre sens c'est un alignement DD vs. MD (voir figure 9.7). En procédant de même, on aboutit alors à la condition $q_{DD,MD} = q_{DD,DM}$.

Ces deux hypothèses de « réversibilité spatiale » pour les processus de substitutions affectant les transitions empruntées dans les HMM profils ne sont que des hypothèses. On doit noter qu'elles sont évidemment incompatibles avec l'idée de deux processus (M→) et (D→) séparés, puisque dans ce cas on a des taux instantanés $q_{MM,MD}$ et $q_{DD,DM}$ non nuls alors que par construction $q_{MM,DM}$ et $q_{DD,MD}$ sont alors nuls.

(a) sens 5' → 3' (b) sens 3' → 5'

Figure 9.7. *Un autre cas dans lequel les mêmes alignements de séquences genèrent des histoires évolutives différentes (au-delà de l'orientation des branches de l'arbre) selon le sens de lecture choisi.*

On verra plus loin que lorsqu'on apprend un processus de substitution 6×6 entre transitions en imposant *uniquement* la contrainte de temps-réversibilité, on vérifie expérimentalement sur le processus appris les conditions de réversibilité spatiale ici énoncées.

9.4 Questions de longueurs de branche

Il faut noter d'emblée qu'ici, on a potentiellement pour chacun des nœuds de la phylogénie une architecture de HMM différente (on rappelle que par « architecture » on entend le sous-ensemble des colonnes de l'alignement d'apprentissage retenues pour former les états Match), donc un alignement de transitions différent. Le facteur d'échelle à appliquer à l'arbre ainsi que le paramètre de forme de la loi Gamma de variabilité des vitesses d'évolution sont donc tous deux à optimiser systématiquement pour chacun des HMM de reconstruction ancestrale établis sur chacun des nœuds de l'arbre.

9.5 Calculs de vraisemblance

Une fois déterminés le ou les processus de substitution agissant sur les transitions, on sait calculer les probabilités postérieures correspondant à chacune de ces transitions en tout nœud de l'arbre phylogénétique support, sachant les caractères observés aux feuilles. La figure 9.8 reprend l'exemple de la figure 9.2, en représentant dans le cas où l'on emploie deux processus distincts l'utilisation des probabilités postérieures pour ce qui concerne le processus M→.

9.6 Détermination finale des paramètres du HMM concernant les transitions

Quelle que soit l'approche retenue (un unique processus ou deux processus distincts), les probabilités de transitions forment deux sommes séparées : pour tout nœud i du HMM on a $\Pr(M_{i+1}|M_i) + \Pr(D_{i+1}|M_i) + \Pr(I_i|M_i) = 1$ et $\Pr(M_{i+1}|D_i) + \Pr(D_{i+1}|D_i) + \Pr(I_i|D_i) = 1$.

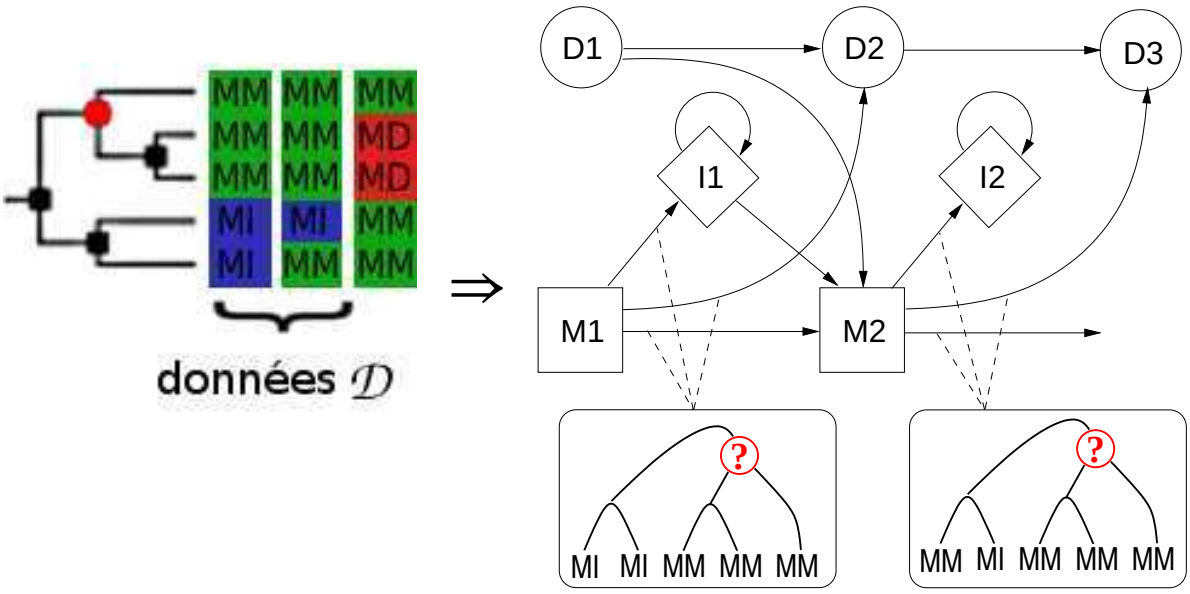


Figure 9.8. *Cadre de la dérivation des probabilités de transition quittant les états Match. Exemple des deux premiers sites de l'alignement des transitions issu de l'exemple représenté en figure 9.2, avec la phylogénie correspondante. Dans le cadre où l'on emploie deux processus distincts, les probabilités de transition sortant des deux états Match condidérés sont directement données par les probabilités postérieures calculées au nœud d'intérêt.*

Dans le cas où les deux processus M→ et D→ sont séparées, les postérieures calculées donnent directement les six probabilités de transition. Dans le cas où les deux processus sont réunis en un seul (cf. plus haut), il faut normaliser les probabilités postérieures ici dénotées p. Pour ne donner que deux exemples, on écrira :

$$\Pr(M_{i+1}|M_i) = \frac{p_{MM}}{p_{MM} + p_{MI} + p_{MD}}$$

et

$$\Pr(M_{i+1}|D_i) = \frac{p_{DM}}{p_{DM} + p_{DI} + p_{DD}} .$$

CHAPITRE

10 Transitions quittant les états d'Insertion

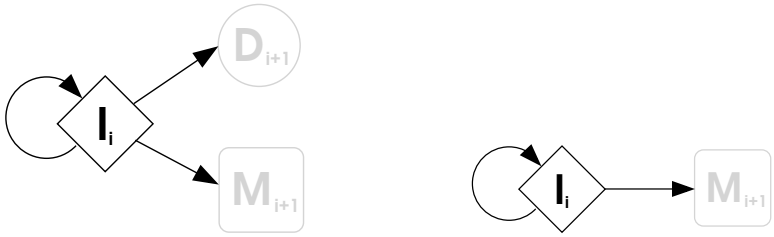
Au chapitre précédent nous avons traité des transitions quittant les états Match et les états de Délétion. Ces transitions ne comportaient pas de boucle. Les états d'Insertion, eux, sont bien particuliers : ils bouclent sur eux-mêmes. La phylogénisation des transitions au départ d'un état d'Insertion ne se fait donc pas comme précédemment, mais via la modélisation de l'évolution des longueurs d'insertion le long des branches de la phylogénie sous-tendant les séquences d'apprentissage. C'est cette modélisation que nous examinons ici, en explorant deux voies : l'une fait appel à des modèles markoviens en espace discret ($\mathbb{N}^*$), l'autre à des modèles sur un espace continu ($\mathbb{R}$).

Sommaire

10.1 Problématique

Ainsi qu'on a pu le voir lorsque nous avons présenté les architectures usuelles de HMM profils en section 3.2, les états d'Insertion ont ceci de particulier qu'ils bouclent sur eux-mêmes. Ces états se présentent donc comme en figure 10.1(a) ou en figure 10.1(b), selon qu'on utilise les HMM profils de SAM ou bien l'architecture « Plan 7 » de HMMER. Cette particularité induit le fait suivant, fondamental pour bien comprendre les enjeux de

la modélisation de ces états d'Insertion.



(a) dans le logiciel SAM (b) pour la suite HMMER

Figure 10.1. *Structure des transitions au départ d'un état d'insertion, selon que l'on se trouve dans le cadre de l'architecture « historique » proposée par [Krogh* et al.*, 1994] ou bien dans celui de l'architecture Plan7 de* HMMER

Considérons une zone d'insertion de longueur N. Ici et dans toute la suite, nous appelons « zone d'insertion » de longueur N tout ensemble de N colonnes consécutives non retenues pour former l'architecture du modèle. Soit une séquence X présentant dans cette zone d'insertion $n \leq N$ caractères insérés. Soit $\mathscr{M}$ le HMM profil considéré et $s_{\mathscr{M}}(X)$ le score de la séquence X dans le HMM $\mathscr{M}$. Soit σ une permutation quelconque des caractères de X n'affectant que la zone d'insertion considérée. Alors pour toute permutation σ respectant ce critère, on a l'égalité des scores de séquence : $s_{\mathscr{M}}(X) = s_{\mathscr{M}}(\sigma(X))$. En effet, si l'on note X_k à X_{k+n-1} les n caractères correspondant à cette zone d'insertion dans la séquence X, $e_I(\alpha)$ la probabilité d'émission du caractère α par l'état d'Insertion en question, p_{MI} la probabilité d'entrée dans la zone d'insertion considérée, p_{II} la probabilité de bouclage et p_{IM} la probabilité de sortie de l'état I (on se place dans le cadre de l'architecture *Plan7* rappelée en figure 10.1(b), si bien que l'entrée et la sortie de l'état d'Insertion ne peuvent se faire respectivement qu'en provenance d'un état Match et à destination de l'état Match suivant), alors le score (incluant les probabilités d'entrée et de sortie de l'état d'Insertion) engendré par la sous-séquence de X correspondant à cette zone d'insertion, quels que soient $n \leq N$ et les emplacements occupés par ces n caractères dans les N colonnes qui composent la zone, est calculé (rapport des logarithmes par rapport à la probabilité de la séquence dans le modèle nul) à partir de :

$$\Pr(\text{insertion}|\mathscr{M}) = p_{MI}\, p_{II}^{n-1}\, p_{IM} \prod_{i=0}^{n-1} e_I(X_{k+i})$$

Ainsi, sans tenir compte des contraintes biologiques qui existent nécessairement, les HMM profils considèrent les zones d'insertion comme des zones à l'intérieur desquelles les séquences *ne sont pas vraiment alignées les unes aux autres*. Dans ce contexte, il devient

illusoire de vouloir aligner le *contenu* de telles zones. Les seules informations pertinentes sont, pour chacune des séquences :

1. longueur de l'insertion (nombres de caractères insérés),
2. composition de l'insertion (comptage des différents caractères insérés).

Nous traiterons le deuxième point au chapitre suivant. Nous nous consacrons pour l'instant à l'apprentissage d'un point de vue phylogénétique des probabilités de transition au départ de l'état d'Insertion, en lien avec les longueurs d'insertion observées dans l'alignement.

10.1.1 Le modèle des zones d'insertion dans les HMM profils

Nous nous plaçons pour plus de simplicité dans le cadre des HMM profils de HMMER, c'est-à-dire avec une structure de nœud qui donne des états d'Insertion correspondant au schéma de la figure 10.1(b). Dans ce cadre, notons p la probabilité de transition depuis I vers l'état M suivant, et donc $(1-p)$ la probabilité de bouclage sur l'état I. La probabilité d'avoir une insertion effective de $n \geq 1$ caractères dans la zone d'insertion considérée est donc égale à $(1-p)^{n-1}p$. C'est-à-dire que dans ce modèle, les longueurs d'insertion sont distribuées selon une *loi géométrique* de paramètre $p \in]0,1[$ (on rappelle qu'on ne considère pas ici le contenu des insertions, seulement leur longueur). L'espérance d'une telle loi est égale à $\frac{1}{p}$ et sa variance à $\frac{1-p}{p^2}$. La variance de la loi est donc très forte pour des probabilités p faibles (variance supérieure à 10 pour des probabilités de sortie de la zone d'insertion inférieures à $0,3$) mais très faible dès lors que la probabilité de bouclage est relativement faible (variance inférieure à 1 pour une probabilité de bouclage inférieure à $0,4$).

Dans la réalité, les zones d'insertion-délétion observées dans les séquences protéiques affichent des longueurs qui ne suivent pas une telle loi géométrique (voir par exemple [Qian et Goldstein, 2001]). Le problème qui nous intéresse n'étant pas ici de rendre compte avec un seul modèle de la totalité des zones d'insertion-délétion rencontrées dans les séquences biologiques, mais d'avoir un modèle spécifique (paramètre q) pour chacune des zones d'insertion issues de l'alignement de séquences d'apprentissage à un HMM profil, une loi géométrique peut être considérée comme une approximation suffisante. De toute façon, ce sont les règles du jeu auxquelles nous devons nous conformer si nous voulons construire des HMM profils qui soient ensuite manipulables par les outils existants (*hmmsearch, hmmscan,* etc.).

10.2 Phylogéniser les longueurs d'insertion : deux approches bien différentes

Il apparaît clairement que si nous savons déterminer le paramètre p adéquat pour chacune des zones d'insertion d'un phylo-HMM donné en fonction de la position phylogénétique à laquelle se réfère ce phylo-HMM, de l'arbre phylogénétique sous-jacent et des longueurs d'insertion observées chez les autres séquences, alors nous aurons atteint le but recherché dans cette section. Pour ce faire, il nous faut déterminer :

1. quels caractères quantitatifs allons-nous considérer comme formant l'alphabet de nos données aux feuilles de l'arbre ?
2. quels modèles de substitution allons-nous utiliser pour ces caractères ?
3. comment allons-nous dériver précisément de la reconstruction ancestrale la valeur recherchée pour le paramètre p ?

Deux solutions s'offrent à présent à nous : ou bien nous considérons des processus évolutifs agissant le long des branches en prenant leurs valeurs dans $\mathbb{N}^*$, et alors ce sont des distributions de probabilité sur $\mathbb{N}^*$ que nous récupérons comme produit de la reconstruction ancestrale ; ou bien nous considérons qu'en tout point de la phylogénie on a une distribution géométrique pour les longueurs d'insertion, dont le paramètre évolue le long des branches selon un processus markovien qui prend ses valeurs dans l'intervalle réel $]0,1]$.

La première solution ne demande aucune interprétation en ce qui concerne les valeurs observées aux feuilles : ce sont directement les longueurs d'insertion observées dans les séquences d'apprentissage. Par contre, toujours dans le cadre de la première solution, une opération est nécessaire pour dériver une valeur de paramètre p à partir de la reconstruction ancestrale (qui donne une distribution de probabilités sur $\mathbb{N}^*$) : nous devons d'une manière ou d'une autre envisager une distribution géométrique à partir d'une *autre* distribution qui n'a aucune raison de l'être. Nous verrons dans la suite comment on peut résoudre cette question simplement, par la méthode des moments.

La deuxième solution envisagée ici peut apparaître comme plus cohérente d'un point de vue mathématique : nous conservons tout le long des branches de l'arbre le modèle sous-jacent des états d'Insertion tels que définis par les HMM profils (i.e. le caractère géométrique des distributions de longueurs d'inserts). Ceci demande de réaliser une opération d'interprétation aux branches, pour dériver une valeur de paramètre p à partir d'une longueur observée. Nous verrons plus loin comment on peut apporter une solution à ce sous-problème en choisissant le paramètre maximisant la vraisemblance de la feuille. Ici, le travail de détermination du paramètre issu de la reconstruction ancestrale au nœud

d'intérêt est plus direct que dans le cadre de la première solution. En effet, il s'agit de déterminer le paramètre $p \in]0,1]$ à partir d'une distribution sur ce même intervalle. On peut alors choisir la moyenne de la distribution, sa valeur maximale ou encore sa valeur médiane.

10.3 Première approche : processus agissant sur les longueurs d'insertion

Nous nous attachons ici à déterminer les paramètres concernant les transitions au départ d'un état d'Insertion, en utilisant la même approche basée sur la phylogénie qu'auparavant et en utilisant les observations que constituent les longueurs d'insertions réalisées par les différentes séquences aux feuilles. Puisqu'on désire utiliser l'information apprise pour modéliser les transitions *au départ d'un état d'Insertion*, il est logique de ne s'appuyer que sur des observations qui concernent les séquences *passant* dans cet état d'Insertion. Ainsi, les observations sont *des entiers positifs stricts*. Ce sont ces observations qui constituent les caractères observés aux feuilles de notre phylogénie.

Considérons par exemple l'alignement d'apprentissage présenté en figure 10.2. Nous supposons que l'étiquetage de l'alignement en colonnes match et non match est tel que la zone d'insertion qui nous intéresse correspond aux cinq colonnes centrales : le premier et le dernier site de la figure 10.2 correspondent à des colonnes « match ». Dans ce cadre, seules les cinq premières séquences empruntent l'état d'Insertion qui nous intéresse. Nous représentons en figure 10.3 la traduction de cette situation en un alignement impliquant des caractères quantitatifs entiers. Le problème qui nous préoccupe alors est représenté en figure 10.4 : il s'agit de pouvoir inférer des valeurs par un processus de reconstruction ancestrale sur un arbre dont les feuilles portent des caractères pris dans $\mathbb{N}^*$.

10.3.1 Processus agissant le long des branches de l'arbre

On cherche à utiliser un processus markovien à valeurs dans $\mathbb{N}^*$, si possible réversible (pour faciliter les calculs de vraisemblance et surtout les rendre insensibles à la position de la racine dans l'arbre). Parmi les processus les plus simples remplissant ces conditions, on trouve les processus dits « de naissance et de mort ». Il s'agit de processus markoviens de saut, définis sur un ensemble d'états correspondant directement à l'ensemble $\mathbb{N}$ (ou à $\mathbb{N}^*$). À partir d'un état i, les seules transitions possibles vont vers son successeur $i+1$ (on parle alors d'une « naissance ») ou bien vers son prédecesseur $i-1$ (une « mort », par analogie avec la modélisation d'une population d'individus). En temps continu, le taux instantané de transition correspondant à une naissance depuis l'état i s'écrit λ_i et le taux instantané

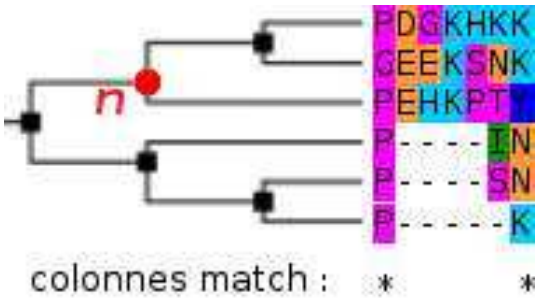


Figure 10.2. *Un exemple de zone d'insertion avec sa phylogénie support*

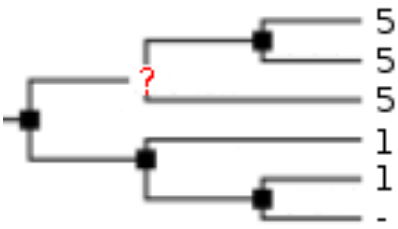


Figure 10.3. *Traduction de la figure ci-contre en un alignement sur des caractères quantitatifs*

correspondant à une mort depuis le même état est μ_i. On représente un tel processus sur $\mathbb{N}^*$ en figure 10.5.

Le problème que pose un tel processus est celui de l'apprentissage des paramètres : considéré sur un ensemble d'entiers non borné, il admet un nombre infini de paramètres λ_i et μ_i ! C'est pourquoi une simplification courante consiste à envisager le processus à taux constants, dans lequel tous les λ_i sont égaux à λ et tous les μ_i à μ.
Si l'on se cantonne à la classe des processus de naissance et de mort *stationnaires*, alors il existe une distribution d'équilibre π sur les états de la chaîne, et cette distribution vérifie les conditions classiques d'équilibre local et global sur une chaîne de Markov (conditions d'équilibre portant sur les entrées et sorties concernant respectivement un état et une bipartition de l'ensemble des états de la chaîne). La condition d'équilibre global s'écrit :

$$\forall i \qquad \lambda\pi_i = \mu\pi_{i+1} \tag{10.1}$$

et la condition d'équilibre local :

$$\forall i \qquad \lambda\pi_{i-1} + \mu\pi_{i+1} = (\lambda+\mu)\pi_i \tag{10.2}$$

À partir de ces deux contraintes ainsi que de l'évidence $\sum_i \pi_i = 1$, on détermine aisément la distribution d'équilibre du processus de naissance et de mort à taux constants sur $\mathbb{N}^*$:

$$\forall i \qquad \pi_i = \left(1 - \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^{i-1} \tag{10.3}$$

On observe au passage une condition *sine qua non* pour que le processus soit stationnaire : on doit avoir $0 < \lambda < \mu$, c'est-à-dire un taux instantané de naissance inférieur au taux instantané de mort. Nous remarquons de plus que la distribution d'équilibre du

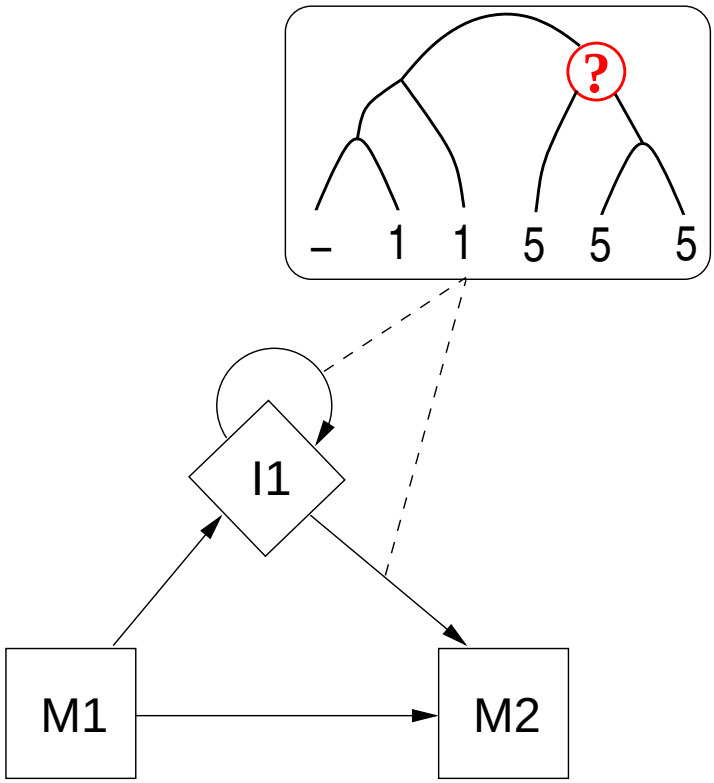


Figure 10.4. *Le problème de l'inférence phylogénétique sur des caractères quantitatifs entiers*

processus, quand elle existe, est exactement la distribution géométrique de paramètre $p = 1 - \frac{\lambda}{\mu}$. Cela signifie qu'avec un tel processus de naissance et de mort, la distribution des longueurs d'insertion observées au bout d'une branche de longueur infinie suit une loi géométrique de paramètre $1 - \frac{\lambda}{\mu}$. Pour déterminer les paramètres λ et μ du processus de naissance et de mort, il convient de se rappeler que si les longueurs d'insert suivent une loi géométrique de paramètre p, alors l'espérance de la longueur d'un insert est égale à $\frac{1}{p}$ (cf. ci-avant en 10.1.1). Plusieurs logiciels (par exemple Prank) utilisant pour les alignements multiples de protéines des pénalités de gap revenant à avoir une distribution géométrique des longueurs de gap avec une espérance de longueur de gap égale à 2, nous avons choisi de suivre ce choix et de fixer $p = 1 - \frac{\lambda}{\mu} = 0,5$, c'est-à-dire $\mu = 2\lambda$. Le choix de la valeur de l'un des deux paramètres est ensuite déterminé par des considérations de vitesse d'évolution [1], et on se décide par exemple à partir du nombre de substitutions attendues sur une

1. En effet, on sait que le processus reste dans un état donné pendant un temps aléatoire distribué selon une loi $\mathrm{Exp}(\lambda + \mu)$, c'est-à-dire en moyenne pendant une durée $\frac{1}{\lambda+\mu}$.

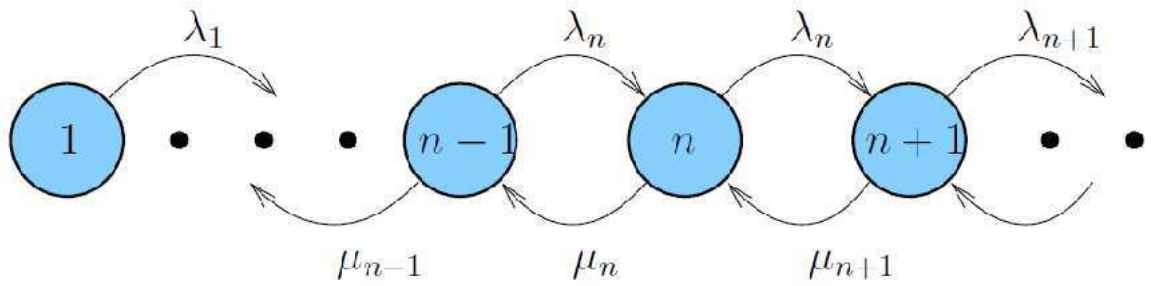


Figure 10.5. *Processus générique linéaire de naissance et de mort sur l'ensemble des entiers naturels non nuls*

branche de longueur 1.

10.3.2 Implémentation concrète des calculs de vraisemblance

L'espace d'états du processus envisagé, $\mathbb{N}^*$, est déraisonnablement grand. Le calcul des probabilités de transition correspondant à une durée t ne peut se faire directement dans ce cadre, et il est important de pouvoir tronquer le processus pour travailler sur $[1, N]$ afin de pouvoir manipuler une matrice Q de dimension finie comme générateur du processus.

La théorie des chaînes de Markov nous apprend que la stabilité d'un processus par troncature n'existe que si ce processus est réversible. Nous rappelons la définition de la réversibilité pour les processus sur un espace fini : $\forall i, j \quad \pi_i q_{ij} = \pi_j q_{ji}$. Dans le processus linéaire de naissance et de mort, les seules transitions qui existent interviennent entre deux états consécutifs de la chaîne, soit par exemple $j = i + 1$. Dans ce cas, la relation de temps-réversibilité est bien vérifiée puisque $\frac{\pi_{i+1}}{\pi_i} = \frac{\lambda}{\mu} = \frac{q_{i,i+1}}{q_{i+1,i}}$.

Cette propriété de réversibilité permet d'utiliser le fait que les Chaînes de Markov à Temps Continu qui sont réversibles sont aussi stables par troncation, c'est-à-dire pour ce qui nous concerne que la chaîne de naissance et de mort tronquée à l'ensemble des états dans $\mathscr{E} = [\![1, N]\!]$ est encore une CMTC réversible dont les probabilités stationnaires s'expriment simplement : $\forall i \in \mathscr{E} \; \pi_{\text{trunc}}(i) = \frac{\pi(i)}{\sum_{k \in \mathscr{E}} \pi(k)}$. On peut par exemple choisir $N = 40$, ce qui semble raisonnable au moins pour toutes les espèces courantes, impliquées dans des alignements où l'on ne s'attend pas à avoir des insertions de plus de 40 résidus par rapport au consensus des autres séquences de la famille.

10.3.3 Dérivation du paramètre d'intérêt à partir de la reconstruction ancestrale

Ainsi, dans tout problème semblable à celui représenté en figure 10.4, en faisant agir le processus de Markov décrit ci-dessus, le produit de la reconstruction ancestrale en un nœud donné de la phylogénie est une distribution de probabilité pour les états du processus. L'espérance m calculée sur cette distribution constitue un estimateur possible pour la longueur attendue de l'insert sur le nœud en question. La probabilité de bouclage p sur l'état d'insertion du nœud correspondant dans le HMM (cf. figure 10.1(b)) s'obtient alors directement : $m = \frac{1}{p} \Rightarrow p = \frac{1}{m}$. La méthode que l'on utilise ce faisant est appelée « méthode des moments », parce qu'elle consiste à faire coïncider le premier moment (c'est-à-dire l'espérance) d'une distribution inférée, avec le moment correspondant d'une distribution théorique.

10.4 Deuxième approche : processus agissant sur le paramètre de la loi géométrique

Nous changeons maintenant notre fusil d'épaule pour adopter une stratégie bien différente. Il ne s'agit plus de travailler sur un processus à valeurs dans $\mathbb{N}$ pour ensuite calculer un simple paramètre p à partir de la reconstruction ancestrale d'une distribution sur $\mathbb{N}$ pour un nœud donné de la phylogénie. Puisque *in fine* c'est la seule valeur p du paramètre qui nous intéresse, et puisque nous nous appuyons sur un modèle (le HMM profil) dans lequel les longueurs d'insertion suivent *toujours* une distribution géométrique, nous pouvons faire l'hypothèse que tout le long de l'arbre le même modèle générateur de longueurs d'insertion est conservé. Seul le paramètre p varierait le long des branches de l'arbre, selon un processus markovien qui reste à établir.

10.4.1 Détermination des valeurs de paramètre aux feuilles

Clairement, notre processus markovien aura pour support l'ensemble des valeurs possibles pour le paramètre p des distributions géométriques valables. Cet ensemble est le semi-ouvert $]0,1]$: si $p = 0$ est impossible car il transforme l'état I en état-puits, $p = 1$ est en revanche licite : c'est la valeur de paramètre imposant des insertions de longueur 1. Aux feuilles, nous observons des entiers naturels positifs stricts correspondant aux longueurs des insertions réalisées dans les séquences d'apprentissage. Comment passer de ces valeurs à des valeurs pour le paramètre p ?

Nous pouvons adopter la stratégie consistant à énoncer que la valeur estimée $\hat{p}_l$ du paramètre à une feuille donnée est celle qui maximise la probabilité de l'observation (lon-

gueur d'insert l) correspondant à cette feuille :

$$\hat{p}_l = \arg\max_p \Pr(l|p) \tag{10.4}$$

Dans le modèle géométrique, on a $\Pr(l|p) = (1-p)^{l-1}p$. Pour maximiser cette quantité à $l \geq 2$ fixé, on dérive suivant p :

$$\frac{\partial \Pr(l|p)}{\partial p} = -(l-1)(1-p)^{l-2}p + (1-p)^{l-1} = (1-p)^{l-2}(1-lp) \tag{10.5}$$

Cette dérivée partielle étant nulle pour $p = \frac{1}{l}$, positive avant ce point et négative après, on a bien un maximum. Si $l = 1$, alors puisque $\Pr(l = 1|p) = p$, on a directement une probabilité maximale lorsque p vaut 1.

Ainsi, on a déterminé dans tous les cas l'estimateur $\hat{p}_l$ au maximum de vraisemblance pour l'observation l :

$$\forall l \in \mathbb{N}^* \qquad \boxed{\hat{p}_l = \frac{1}{l}} \tag{10.6}$$

L'observation d'une longueur d'insert égale à $n \geq 1$ sur une séquence donnera donc le « caractère » $\frac{1}{n} \in]0,1]$ sur la feuille correspondante de l'arbre phylogénétique sur lequel le processus agira.

10.4.2 Choix et construction du processus markovien

Dans ce qui suit, nous appelons E l'intervalle réel $]0,1]$. Nous cherchons à déterminer un processus markovien de support E, stationnaire (la nature et les paramètres du processus ne changent pas au cours du temps, et les probabilités jointes du processus sur un échantillon d'instants sont insensibles par translation dans le temps) et si possible réversible. Pour des raisons de continuité inhérentes au modèle phylogénétique et appropriées à l'idée que l'on se fait de l'évolution du paramètre le long des branches de l'arbre, le processus proscrira les évolutions brusques. De plus, il est souhaitable que ce processus ne puisse pas diverger à l'infini : au bout d'une branche très longue, on souhaite retrouver une certaine information a priori correspondant aux longueurs d'insert attendues dans les séquences biologiques en l'absence d'informations concernant les séquences voisines ou homologues. Les caractéristiques présentées ci-dessus font que le processus recherché appartiendra à ce qu'on appelle la classe des processus de diffusion, qui sont le pendant dans un monde continu des processus de saut dont font partie les processus de naissance et de mort décrits plus haut.

Une courte recherche bibliographique dans le domaine des phylogénies établies sur des caractères quantitatifs [Felsenstein, 1988; Goloboff *et al.*, 2006] convainc le lecteur que

l'usage de caractères quantitatifs le long d'une phylogénie a souvent été cantonné au domaine de la cladistique, et principalement utilisé dans des analyses basées sur la parcimonie. L'approche la plus classique consiste à discrétiser avec des méthodes plus ou moins fines l'espace d'états continu des observations pour manipuler ensuite un espace d'états fini. Les seuls travaux basés sur des caractères continus modélisés comme tels *et* utilisés via une approche de maximum de vraisemblance sont ceux de Joseph Felsenstein [Felsenstein, 1988] concernant une méthode dite « des contrastes » dont le modèle sous-jacent est celui du mouvement brownien. Joe Felsenstein lui-même citait dans son travail le modèle développé par George Eugene Uhlenbeck et Leonard Salomon Ornstein en 1930 [Uhlenbeck et Ornstein, 1930], lesquels ont construit un modèle qui intègre en plus du mouvement brownien une force de rappel élastique vers une valeur moyenne. C'est ce modèle que nous nous proposons d'utiliser ici.

Rappels sur le mouvement brownien

Le mouvement brownien est un processus qui nous vient du champ de la physique des particules et qui est réputé modéliser le mouvement d'une particule dans un milieu uniforme, isotrope, non borné et globalement immobile (fluide ou gazeux) : une telle particule est soumise en permanence à de petits chocs pseudo-aléatoires (collisions avec les autres particules, interactions électroniques, etc) qui déterminent sa trajectoire. La particule se déplace donc. Le processus stochastique modélisant le mouvement brownien est appelé *processus de Wiener*. Sa caractéristique est la suivante : soit une particule W décrivant une trajectoire $W(t)$ qui correspond à une réalisation du processus de Wiener. Si $W(t) = x$, alors au bout d'une durée $\tau > 0$ la position $W(t+\tau)$ de ladite particule suit une loi gaussienne centrée en x et de variance $a\tau$. La constante $a > 0$ est le seul paramètre du processus de Wiener : plus elle est élevée et plus les fluctuations temporelles du processus sont importantes.

Si le processus de Wiener démarre en un point $W(0) = x_0$, sa position évoluera progressivement mais n'est pas bornée : la variance accumulée augmentant au cours du temps, au bout d'un temps infiniement long $W(t)$ suit une loi de probabilité correspondant à une gaussienne centrée en 0 mais de variance infinie, c'est-à-dire une loi qui tend vers une distribution uniforme sur $\mathbb{R}$ (pour un processus de Wiener en dimension 1, ce qu'on suppose dans la suite).

Le processus de Wiener étant défini sur un espace d'états continu, on ne parle plus de probabilités de transition d'un état à un autre en un temps donné, mais de *densités de probabilité de transition* d'un point vers un autre. Les probabilités de transition se calculent en intégrant les densités de probabilité de transition sur des intervalles. Si $q_t(x, y)$ est la densité de probabilité de transition du point x vers le point y en un temps t, la probabilité

de transition depuis le point x vers un intervalle $J = [j_-, j_+]$ est donnée par :

$$\Pr(W(t) \in J | W(0) = x) = \int_{j_-}^{j_+} q_t(x, y) dy \tag{10.7}$$

Et comme $q_t(x, y)$ est directement donné par la densité d'une loi $\mathcal{N}(x, \sigma^2 = at)$, on obtient :

$$\Pr(W(t) \in J | W(0) = x) = \int_{j_-}^{j_+} \frac{1}{\sqrt{2\pi at}} e^{-\frac{(y-x)^2}{2at}} dy \tag{10.8}$$

Le processus de Wiener satisfait l'équation standard de diffusion des physiciens, qui exprime succinctement le lien entre la variabilité spatiale et la variabilité temporelle du processus :

$$\frac{\partial q_t(x, y)}{\partial t} = \frac{a}{2} \frac{\partial^2 q_t(x, y)}{\partial x^2} \tag{10.9}$$

Processus d'Ornstein-Uhlenbeck

Exposé par les auteurs dont il porte le nom dans [Uhlenbeck et Ornstein, 1930], ce processus est défini en toute généralité par l'équation différentielle stochastique [Øksendal, 2003] suivante :

$$dX(t) = X(t + dt) - X(t) = \theta(\mu - X(t))dt + \sigma dW(t) \tag{10.10}$$

Le membre droit de l'équation ci-dessus se décompose en deux parties : on lit tout d'abord l'expression de la *force de rappel élastique* qui est proportionnelle à la distance entre le point courant $X(t)$ et l'espérance du processus, μ. Cette force engendre un déplacement élémentaire $\theta(\mu - X(t))dt$ qui est positif lorsque $\mu > X(t)$ et négatif dans le cas inverse : c'est donc bien une force de rappel *vers* μ. L'autre composante du déplacement élémentaire est proportionnelle à un mouvement brownien : σ est un coefficient multiplicateur (égal à $\sqrt{a}$ si l'on fait l'analogie avec ce qui a été exposé plus haut) qui s'applique à un processus de Wiener normalisé $W(t)$ (la variance de W est égale à 1). Cette seconde composante ajoute donc à la première des fluctuations locales gaussiennes dont l'amplitude est donnée par σ. Au total, le processus $X(t)$ présente des fluctuations aléatoires mais tend à revenir vers la valeur μ, qui est égale à l'espérance du processus.

Un théorème fondamental dû à J.L. Doob et portant son nom [Doob, 1942] stipule que l'équation différentielle stochastique (10.10) décrit un processus qui est à la fois :

1. stationnaire,
2. gaussien,
3. markovien,

4. continu en probabilité.

La *stationnarité* du processus X_t impose que la probabilité jointe du processus sur un échantillon d'instants est la même que la probabilité jointe du processus sur le même échantillon *translaté dans le temps* :

$$\forall \tau > 0,\ t_1 < t_2 < \ldots < t_n \quad \Pr(X_{t_1}, X_{t_2}, \ldots, X_{t_n}) = \Pr(X_{t_1+\tau}, X_{t_2+\tau}, \ldots, X_{t_n+\tau})\ . \tag{10.11}$$

Le caractère *gaussien* de X_t impose que pour tout échantillon d'instants $t_1 < t_2 < \ldots < t_n$, le vecteur $(X_{t_1}, X_{t_2}, \ldots, X_{t_n})$ se compose de n variables aléatoires distribuées chacune selon une loi normale.

La *propriété de Markov*, elle, est bien connue : elle impose l'absence de mémoire, c'est-à-dire que les états passés ne contribuent en rien à prédire l'état futur du processus, seul compte son état présent :

$$\forall\, t_1 < t_2 < \ldots < t_n \quad \Pr(X_{t_n} | X_{t_1}, X_{t_2}, \ldots, X_{t_{n-1}}) = \Pr(X_{t_n} | X_{t_{n-1}})\ . \tag{10.12}$$

Enfin, la notion de *continuité en probabilité* rend infiniment improbables des déplacements discontinus :

$$\forall \epsilon > 0, t > 0 \quad \lim_{\tau \downarrow 0} \Pr\left(|X_{t+\tau} - X_t| > \epsilon\right) = 0 \tag{10.13}$$

L'équation de diffusion pour un processus d'Ornstein-Uhlenbeck d'espérance μ, de variance infinitésimale σ^2 (pour la partie de mouvement brownien) et avec une force de rappel d'intensité θ s'écrit (et l'on peut comparer avec (10.9)) :

$$\frac{\partial q_t(x,y)}{\partial t} = \frac{a}{2}\frac{\partial^2 q_t(x,y)}{\partial x^2} - \theta(x-\mu)\frac{\partial q_t(x,y)}{\partial x} \tag{10.14}$$

10.4.3 Calcul théorique des probabilités de transition dans un processus d'Ornstein-Uhlenbeck (OU)

Nous cherchons ici à déterminer comment évolue $X(t)$ de façon analytique : quelle sont les premiers moments de $X(t)$? Pour ce faire, nous introduisons d'abord la fonction f pour utiliser ensuite ses dérivées partielles. Pour alléger les expressions, nous écrivons systématiquement X_t pour $X(t)$.

Soit $f(X_t, t) = X_t e^{\theta t}$. On exprime la variation infinitésimale de f :

$$\begin{aligned}
df(X_t,t) &= \frac{\partial f(X_t,t)}{\partial t}dt + \frac{\partial f(X_t,t)}{\partial X_t}dX_t && (10.15)\\
&= \theta X_t e^{\theta t}dt + e^{\theta t}dX_t && (10.16)\\
&= \theta X_t e^{\theta t}dt + e^{\theta t}\left(\theta\mu dt - \theta X_t dt + \sigma dW_t\right) && (10.17)\\
&= e^{\theta t}\theta\mu dt + \sigma e^{\theta t}dW_t && (10.18)
\end{aligned}$$

En intégrant de 0 à t, on a :

$$X_t e^{\theta t} - X_0 = \int_0^t df(X_s, s) = \int_0^t e^{\theta s} \theta \mu ds + \int_0^t \sigma e^{\theta s} dW s \tag{10.19}$$

et on trouve finalement :

$$\boxed{X_t = X_0 e^{-\theta t} + \mu(1 - e^{-\theta t}) + \int_0^t \sigma e^{-\theta(t-s)} dW s} \tag{10.20}$$

Le troisième terme de la somme figurant en membre droit de (10.20) étant l'intégration d'un bruit gaussien centré, son espérance est nulle. On a ainsi le résultat fondamental suivant :

$$\mathbb{E}[X_t | X_0] = X_0 e^{-\theta t} + \mu(1 - e^{-\theta t}) \tag{10.21}$$

Le calcul de la variance de X_t est un peu plus complexe, en ceci qu'il fait intervenir l'isométrie de Itō, sans doute inconnue du lecteur peu familier avec le calcul stochastique (on se référera à [Øksendal, 2003] pour une introduction au domaine). Cette variance, apportée par le troisième terme du membre droit de (10.20), s'exprime comme suit :

$$\mathrm{Var}[X_t | X_0] = \frac{\sigma^2}{2\theta}\left(1 - e^{-2\theta t}\right) \tag{10.22}$$

La distribution stationnaire du processus étant celle atteinte après un temps infiniment long, il se déduit aisément de (10.21) et de (10.22) que cette distribution stationnaire correspond à une gaussienne centrée en μ et de variance $\frac{\sigma^2}{2\theta}$ (plus la force de rappel est grande, plus la variance due au mouvement brownien est atténuée).

Le caractère gaussien de X_t étant établi par le théorème de Doob [Doob, 1942], la loi suivie par X_t est entièrement déterminée. On peut dès lors se tourner vers le calcul des densités de probabilités de transition, qui vont être indispensables pour calculer des vraisemblances de modèles arborescents. Soit $t > 0$ un instant quelconque, $\tau > 0$ une durée, $J = [j_-, j_+]$ un intervalle de $\mathbb{R}$. D'après (10.21) et (10.22), la probabilité pour le processus d'Ornstein-Uhlenbeck de passer en une durée τ de la valeur x à l'intervalle J s'écrit :

$$\Pr(x \xrightarrow{\tau} J) = \Pr(X_{t+\tau} \in J | X_t = x) = \int_{j_-}^{j_+} \frac{1}{\sqrt{2\pi \frac{\sigma^2}{2\theta}\left(1 - e^{-2\theta\tau}\right)}} e^{-\frac{1}{2} \frac{\left[\zeta - x e^{-\theta\tau} - \mu\left(1 - e^{-\theta\tau}\right)\right]^2}{\frac{\sigma^2}{2\theta}\left(1 - e^{-2\theta\tau}\right)}} d\zeta \tag{10.23}$$

Enfin, le caractère gaussien et stationnaire du processus d'Ornstein-Uhlenbeck assure sa temps-réversibilité [Weiss, 1975].

10.4.4 Implémentation du calcul des vraisemblances via la discrétisation de l'intervalle]0, 1]

Maintenant que nous savons déterminer (au moins analytiquement) les densités de probabilité de transition d'un point de $\mathbb{R}$ à un intervalle, nous devons nous poser la question de l'implémentation concrète de l'algorithme de pruning de Felsenstein (cf. section 4.4). En effet, nous ne manipulons plus des probabilités dans un espace discret et fini, nous n'avons plus de générateur Q de dimension finie et donc plus de formule de la forme $P_\tau = e^{Q\tau}$ pour calculer les probabilités de transition correspondant à une durée τ. Nous devons nous contenter de l'expression donnée en (10.23), mais surtout nous devons adapter l'algorithme de pruning de Felsenstein à un environnement continu.

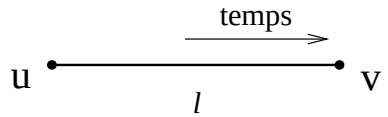


Figure 10.6. *L'arbre $\mathcal{T}$ qui se réduit à une seule branche de longueur l*

Considérons d'abord l'arbre élémentaire $\mathcal{T}$ constitué d'une seule branche et représenté en figure 10.6. Supposons que les caractères portés aux feuilles u et v sont respectivement α et β. Dans l'hypothèse où le processus markovien est à support discret (e.g. le processus WAG d'évolution des acides aminés), on rappelle que la vraisemblance de $\mathcal{T}$ s'écrit : $\mathrm{Lk}_{\mathrm{discret}}(\mathcal{T}) = \pi(\alpha)\Pr(\alpha \xrightarrow{l} \beta)$. Le lecteur aura noté que l'on suppose, comme indiqué en figure 10.6, que le temps s'écoule du nœud u vers le nœud v. Dans le monde des processus à support continu, π fait référence non plus à des probabilités discrètes en nombre fini et sommant à 1, mais à une fonction continue sur $\mathbb{R}$ appelée *densité de probabilité* et dont l'intégrale sur tout $\mathbb{R}$ vaut 1. De même, on ne sait plus calculer des expressions de la forme $\Pr(\alpha \xrightarrow{l} \beta)$ avec $(\alpha, \beta) \in \mathbb{R}^2$, mais seulement des *densités de probabilité de transition* à intégrer comme en (10.23).

Dans l'univers des processus à support continu, la probabilité d'aller d'un point x donné à un autre point y en un temps t est *nulle* quels que soient x, $y \neq x$ et t. Incidemment, la vraisemblance de tout arbre présentant des données aux feuilles qui sont considérées comme autant de *points* de l'ensemble support du processus est tout aussi nulle. Pour pouvoir calculer des vraisemblances, il n'y a pas d'autre choix que de considérer les valeurs aux feuilles non plus comme des points, mais comme des *intervalles* centrés sur la valeur observée. Nous devons commencer par choisir un pas de discrétisation, c'est-à-dire une largeur d'intervalle fixe pour tous nos calculs. Nous appelons cette largeur ϵ. Par commodité, on écrira systématiquement I_x pour l'intervalle $[x - \epsilon/2, x + \epsilon/2]$, intervalle de

largeur ϵ centré en x. On a alors :

$$\mathrm{Lk}_{\mathrm{cont.}}(\mathscr{T}) = \Pr(I_\alpha)\Pr\left(I_\alpha \xrightarrow{l} I_\beta\right) \tag{10.24}$$

Dans la formule ci-dessus, la distribution stationnaire (croyance a priori) est l'intégration de la loi normale évoquée plus haut. Elle s'exprime :

$$\Pr(I_\alpha) = \int_{\alpha-\frac{\epsilon}{2}}^{\alpha+\frac{\epsilon}{2}} \frac{1}{\sigma}\sqrt{\frac{\theta}{\pi}}\, e^{-\frac{\theta(x-\mu)^2}{\sigma^2}}\, dx \tag{10.25}$$

Et d'autre part la probabilité de transition d'un intervalle à l'autre s'écrit :

$$\Pr\left(I_\alpha \xrightarrow{l} I_\beta\right) = \int_{x=\alpha-\frac{\epsilon}{2}}^{\alpha+\frac{\epsilon}{2}} \int_{y=\beta-\frac{\epsilon}{2}}^{\beta+\frac{\epsilon}{2}} q_t(x,y)\,dx\,dy \tag{10.26}$$

La densité de probabilité de transition $q_t(x,y)$ a déjà été utilisée dans l'équation (10.23), elle vaut :

$$q_t(x,y) = \sqrt{\frac{\theta}{\pi}}\,\frac{1}{\sigma\sqrt{1-e^{-2\theta t}}}\, e^{-\frac{\theta\left[y-xe^{-\theta t}-\mu(1-e^{-\theta t})\right]^2}{\sigma^2(1-e^{-2\theta t})}} \tag{10.27}$$

On sait donc calculer la vraisemblance d'un arbre élémentaire (réduit à une branche) sous l'hypothèse d'un processus d'évolution de type OU. Qu'en est-il de l'implémentation de l'algorithme de Felsenstein permettant de calculer des vraisemblances sur des arbres plus complexes (figure 10.7) ?

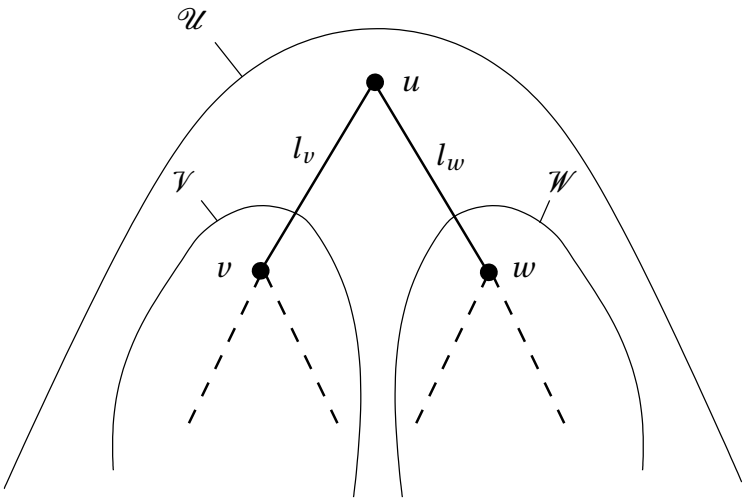


Figure 10.7. *Un arbre phylogénétique de racine u et ses deux sous-arbres*

L'équation de la vraisemblance à la racine, $\mathrm{Lk}(\mathscr{U}) = \sum_\alpha \Pr(\alpha)\,\mathrm{Lk}(\mathscr{U}|u=\alpha)$ devient en milieu continu :

$$\mathrm{Lk}(\mathscr{U}) = \int_{\mathbb{R}} \Pr(x)\,\mathrm{Lk}(\mathscr{U}|u=x)\,dx \tag{10.28}$$

L'équation de récurrence sur les vraisemblances conditionnelles de sous-arbre s'écrivait :

$$\mathrm{Lk}(\mathscr{U}|u=\alpha)=\left[\sum_{\beta}\Pr\left(\alpha\overset{l_v}{\to}\beta\right)\mathrm{Lk}(\mathscr{V}|v=\beta)\right]\cdot\left[\sum_{\gamma}\Pr\left(\alpha\overset{l_w}{\to}\gamma\right)\mathrm{Lk}(\mathscr{W}|w=\gamma)\right] \tag{10.29}$$

Elle devient :

$$\mathrm{Lk}(\mathscr{U}|u=x)=\left[\int_{\mathbb{R}}q_{l_v}(x,y)\,\mathrm{Lk}(\mathscr{V}|v=y)\,dy\right]\cdot\left[\int_{\mathbb{R}}q_{l_w}(x,z)\,\mathrm{Lk}(\mathscr{W}|w=z)\,dz\right] \tag{10.30}$$

Et lorsque l'arbre $\mathscr{U}$ est réduit à une feuille u portant la donnée observée x_{obs}, en cohérence avec la notion d'intervalles présentée plus haut, on a :

$$\mathrm{Lk}(\mathscr{U}|u=x)=\mathbb{1}_{[x_{\mathrm{obs}}-\epsilon/2,x_{\mathrm{obs}}+\epsilon/2]}(x) \tag{10.31}$$

L'implémentation pratique des intégrations sur $\mathbb{R}$ pose deux problèmes :
- le processus OU est non borné, mais les valeurs de paramètre se trouvant en dehors de l'intervalle réel $E=]0,1]$ sont absurdes et à ce titre doivent pouvoir être rejetées ou ignorées,
- en l'absence de formule analytique simple pour les expressions du type $\mathrm{Lk}(\mathscr{W}|w=z)$ considérées comme fonctions de la variable γ ne laisse pas d'autre choix que d'approximer le calcul de l'intégrale par une méthode de type « méthode des rectangles ».

La solution au premier des deux problèmes énoncés ci-dessus fait appel à la nature et aux trajectoires du processus d'Ornstein-Uhlenbeck. Ce dernier se composant d'un terme de rappel vers la valeur moyenne μ de vitesse θ et d'un terme de bruit gaussien d'amplitude donnée par la variance σ^2, on peut légitimement supposer que pour $\mu\in]0,1]$, sous certaines conditions sur θ et σ (θ grand et σ faible, basiquement), et avec toutes les valeurs observées aux feuilles se trouvant dans E, le temps passé à l'extérieur de E par les réalisations du processus stochastique le long des branches de l'arbre est faible. On commet dès lors une erreur raisonnablement peu élevée en intégrant systématiquement sur E plutôt que sur $\mathbb{R}$. On peut se faire une idée plus précise de cette erreur en intégrant la distribution d'équilibre sur E et en calculant donc la probabilité qu'on se trouve à l'équilibre à l'extérieur de E. Pour une espérance $\mu=0,5$, une vitesse de rappel $\theta=4$ et un écart type $\sigma=0,3$, on trouve :

$$\int_{\mathbb{R}\setminus E}\frac{1}{\sigma}\sqrt{\frac{\theta}{\pi}}\,e^{-\frac{\theta(x-\mu)^2}{\sigma^2}}\,dx=2,43\,.\,10^{-6}$$

Avec les mêmes paramètres μ et σ mais avec un rappel moins prononcé $\theta=2$, on a $\Pr(x\notin E)=8,58\,.\,10^{-4}$.

La valeur moyenne de la distribution d'équilibre, μ, peut se déduire d'un grand nombre d'observations (alignements de séquence à des modèles de type HMM profil) : c'est l'inverse de la longueur moyenne des insertions (cf. page 155).

En ce qui concerne l'intégration avec approximation des rectangles, on découpe l'intervalle E, de longueur 1, en $N = \lceil \frac{1}{\epsilon} \rceil$ intervalles élémentaires de largeur ϵ, sur lesquels on considère l'intégrande comme étant constante :

$$\int_E q_{l_v}(x,y)\,\mathrm{Lk}(\mathcal{V}|v=y)\,dy \quad \approx \quad \sum_{i=1}^{N} \epsilon\, q_{l_v}(x,y_i)\,\mathrm{Lk}(\mathcal{V}|v=y_i) \tag{10.32}$$

où chacun des y_i est un représentant de l'intervalle $E_i =]y_i - \epsilon/2, y_i + \epsilon/2]$, avec par construction $\bigcup_{i=1}^{N} E_i =]0,1]$.

10.4.5 Dérivation de la valeur du paramètre à partir de la reconstruction ancestrale

Une fois qu'on a obtenu une distribution a posteriori de densité $d(x)$ pour le paramètre p de la loi géométrique sur le nœud n de l'arbre, plusieurs choix s'offrent à nous :

1. on peut choisir le paramètre qui maximise la fonction de densité : $\hat{p} = \max_{x\in\mathbb{R}} d(x)$,
2. on peut choisir la moyenne de la distribution : $\hat{p} = \int_{\mathbb{R}} x d x$,
3. on peut choisir la valeur médiane de la distribution, m telle que $\int_{-\infty}^{m} dx = 0,5$.

Parmi celles-ci, la deuxième solution est la plus classique : on choisit comme estimateur l'espérance de la variable statistique.

Nous avons présenté ici deux méthodes de détermination phylogénétique des paramètres des HMM de reconstruction ancestrale en ce qui concerne les transitions quittant les états d'Insertion. Pour obtenir des modèles ancestraux dont tous les paramètres sont calculés d'après la phylogénie, il nous reste maintenant à examiner le *contenu* des états d'insertion, c'est-à-dire la question de la détermination des distributions de probabilité pour les émissions sur ces états. C'est l'objet du chapitre suivant.

CHAPITRE

11 Émissions de caractères sur les états d'Insertion du modèle

[Qian et Goldstein, 2003], ainsi que d'autres auteurs avant eux, n'ont pas su intégrer les états d'Insertion dans une démarche de calcul des émissions basée sur la phylogénie (voir section 5.3.1). En effet, ces états ont la particularité, nous l'avons déjà dit, de boucler sur eux-même. Ainsi, un seul état d'Insertion est susceptible de générer une insertion d'acides aminés de longueur quelconque entre deux colonnes Match. De façon concomitante, on doit apprendre les probabilités d'émission sur un tel état à partir, non pas d'une seule colonne d'acides aminés alignés les uns avec les autres, mais de plusieurs colonnes affichant un certain nombre d'acides aminés *non alignés entre eux*. En effet, tout HMM profil modélisant une zone d'insertion par le biais d'*un seul* état bouclant sur lui-même, avec une seule distribution de probabilités d'émission et une seule probabilité de self-transition, la notion d'alignement au sens classique n'existe pas (du point de vue du HMM profil) au sein des états d'Insertion, et les séquences `GMQ---`, `G-M-Q-`, `G---MQ` ou même `M---QG` sont strictement interchangeables du point de vue d'un HMM profil les alignant contre l'un de ses état d'Insertion : elles donnent toutes trois un score égal à $p(G)\,p(M)\,p(Q)\,p_{II}^2\,p_{IM}$, si les p sont les probabilités propres à l'état d'Insertion en question et si le départ de la zone d'insertion se fait à destination de l'état Match du nœud suivant.

Comment apprendre les probabilités d'émission sur ces états d'Insertion en tenant compte de la phylogénie reliant les séquences d'apprentissage ? Les HMM profils calculent classiquement les probabilités d'insertion à partir d'un décompte des acides aminés observés (toutes séquences confondues) dans la zone en question, en mélangeant ces observations à des pseudo-comptes (priors) reflétant la connaissance a priori des acides aminés ayant une propension élevée à se trouver dans des zones structurellement peu conservées et plutôt hydrophiles (voir en section 3.3.4). Mais le mécanisme d'apprentissage est biaisé, comme l'indique Sean Eddy, principal développeur de la suite HMMER :

Les pseudo-comptes correspondant aux émissions sur les états d'insertion ont été artificiellement amplifiés pour atteindre des valeurs α très élevées, ce qui a pour effet de figer les distributions d'émission sur les états d'insertion dans HMMER : les états d'insertion reçoivent tous virtuellement la même distribution d'émission, plutôt que d'apprendre celle-ci individuellement à partir des observations.

Qian et Goldstein [Qian et Goldstein, 2003, 2004] passent totalement sous silence leur traitement des émissions sur les états Match, et l'on peut donc penser que ces auteurs ne procèdent pas différemment de HMMER, c'est-à-dire ignorent l'aspect phylogénétique des choses. Nous proposons en revanche une approche nouvelle, quoique extrêmement simple, permettant de ne pas ignorer la phylogénie.

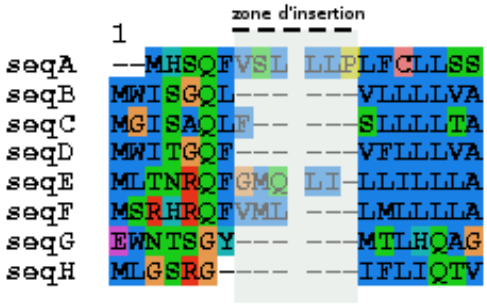


Figure 11.1. *Fragment d'alignement de protéines TXNDC5 chez des drosophiles*

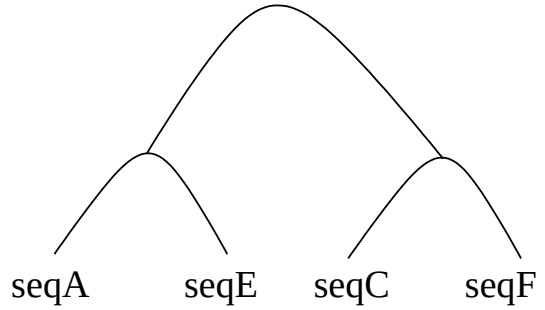


Figure 11.2. *Phylogénie correspondant aux taxa impliqués dans la zone d'insertion*

Considérons conjointement l'alignement de séquences présenté en figure 11.1 et la phylogénie qui se trouve en regard (figure 11.2). Supposons de plus que la zone d'insertion couvre les colonnes surlignées d'un trait pointillé en figure 11.1, c'est-à-dire que la colonne qui précède la première de la zone et celle qui suit la dernière sont toutes deux modélisées par des états Match. Il est important de comprendre que dans l'univers des HMM profils, comme nous l'avons dit plus haut, ni l'ordre des acides insérés, ni leur position sur l'une ou l'autre des 6 colonnes que mesure cette zone d'insertion, n'ont d'importance. Pour déterminer le score d'une sous-séquence dans l'état d'Insertion correspondant d'un HMM profil, seule compte la composition de celle-ci en termes d'acides aminés. Ainsi, aucun processus d'apprentissage des émissions sur les états d'Insertion ne pourra faire mieux que de considérer les acides aminés d'une séquence donnée « alignés » contre cet état d'Insertion, comme étant les éléments d'un *ensemble*, sans autre information pertinente.

Le problème se posant alors au modélisateur qui voudrait utiliser la phylogénie sous-jacente à l'ensemble des séquences d'apprentissage, consiste à travailler avec une phy-

logénie dont les feuilles ne portent non pas chacune un acide aminé, mais un *ensemble* de tels caractères. Cette considération nous mène tout naturellement à représenter la situation décrite par les figures 11.1 et 11.2, en des termes qui sont résumés par la figure 11.3 : on compte pour chaque séquence les différents acides aminés insérés, avec pour chacun un coefficient correspondant à sa fréquence d'apparition dans la zone d'insertion et pour la séquence considérées.

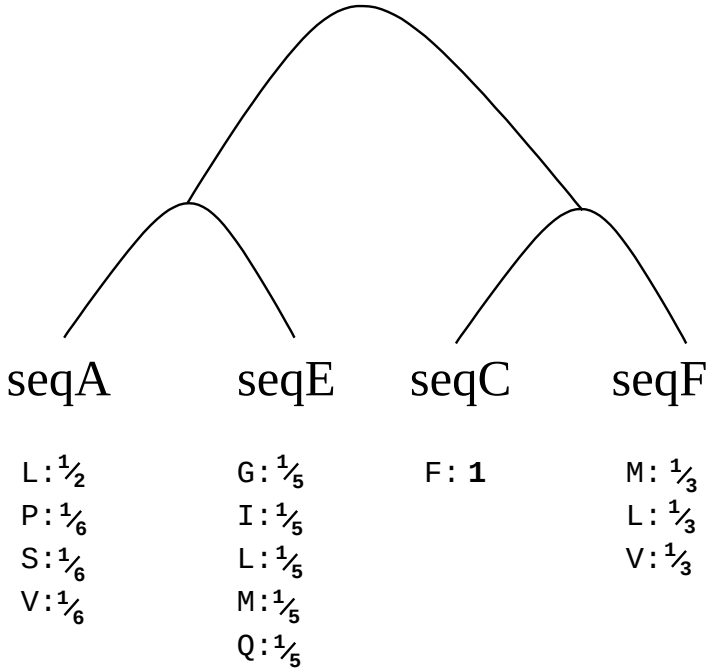


Figure 11.3. *Vraisemblances aux feuilles pour le calcul du profil d'émission d'acides aminés pour l'état d'Insertion correspondant à la zone représentée en figure 11.1*

Rappelons que dans l'algorithme itératif de pruning de Felsenstein (section 4.4), la fin de la récursion correspond au calcul de la vraisemblance d'une feuille. Soit f une telle feuille, c'est-à-dire un arbre élémentaire réduit à un seul nœud. Dans le cas classique d'un site correspondant à une colonne d'un alignement, si a est l'acide aminé effectivement porté par la feuille en question (c'est-à-dire observé dans l'alignement), alors on a simplement :

$$\Pr(f|f=\alpha,Q)=\begin{cases}1 & \text{si } \alpha=a\\ 0 & \text{sinon}\end{cases} \tag{11.1}$$

Nous proposons que dans le cadre des émissions correspondant à un état d'Insertion, cette définition des vraisemblances partielles aux feuilles soit remplacée par :

$$\Pr(f|f=\alpha,Q)=\frac{n_\alpha^f}{\sum_{\beta=1}^{20} n_\beta^f} \quad , \tag{11.2}$$

où n_β^f est le nombre d'observations de l'acide aminé β dans la zone d'insertion à modéliser et dans la séquence d'apprentissage correspondant à la feuille f.

L'intérêt d'une telle modélisation est double :

1. le reste de l'algorithme de pruning de Felsenstein se déroule de façon tout à fait classique, la prise en compte de colonne multiples pour fabriquer un seul site ne présente donc aucun surcoût,
2. la phylogénie est prise en compte car le profil d'émission inféré sur un nœud ancestral dépendra majoritairement des insertions effectuées dans son voisinage phylogénétique.

Ainsi, nous avons présenté dans cette deuxième partie une méthodologie complète de dérivation des paramètres de HMM profils de *reconstruction ancestrale*, c'est-à-dire des modèles de description séquentielle basés sur l'inférence phylogénétique en un point d'intérêt de l'arbre. Nous présentons dans la suite les résultats qu'ont donnés les différents aspects de cette méthode globale de « phylogénisation ».

Troisième partie

Résultats

CHAPITRE

12 Présentation des bancs de test

Nous présentons ici les jeux de données utilisés pour valider nos méthodes de phylogénisation. On définit deux bancs de test avec des distances évolutives moyennes différentes entre l'un et l'autre. Les bancs sont relativement comparables du point de vue de la taille des familles qu'ils comprennent, puisque chaque famille issue de l'un comme de l'autre contient de 3 à 25 séquences. Ceci étant, les familles SABmark sont sensiblement moins fournies (7,8 séquences en moyenne) que les familles du banc de test Treefam (en moyenne 19,4 séquences par famille).

Sommaire

12.1 Distances évolutives modérées : TreeFam

Le premier banc de test est issu de la base de données TreeFam version 7.0 (http://www.treefam.org). Au total, cette base de données comporte 777.321 gènes répartis en 16.141 familles. Parmi toutes ces familles, 1.281 sont dites de classe 'A', pour signaler que les données correspondantes sont passées par une étape d'analyse et de correction via l'expertise humaine. Pour plus de fiabilité, on choisit de travailler uniquement sur les familles de classe 'A' dans leur version « clean ». Il s'agit là de groupes de séquences homologues accompagnés chacun de l'arbre phylogénétique supportant les données. L'alignement et l'arbre phylogénétique ont systématiquement fait l'objet d'une révision via l'expertise humaine. Après un processus itératif d'enrichissement et de validation, les familles « clean » sont issues de la restriction d'un groupe d'homologie aux séquences pro-

venant de 58 espèces, et pour la plupart d'un sous-ensemble de 28 espèces (*Aedes aegypti, Anopheles gambiae, Arabidopsis thaliana, Bos taurus, Brachydanio rerio, Caenorhabditis briggsae, elegans* et *remanei, Canis familiaris, Gallus gallus, Ciona intestinalis* et *savignyi, Dictyostelium discoideum, Drosophila melanogaster* et *pseudoobscura, Gasterosteus aculeatus, Homo sapiens, Macaca mulatta, Monodelphis domestica, Mus musculus, Oryza sativa, Pan troglodytes, Rattus norvegicus, Schistosoma mansoni, Schizosaccharomyces pombe, Tetraodon nigroviridis, Xenopus tropicalis et Saccharomyces cerevisiae*).

Ce jeu de données nous permettra de quantifier les corrélations entre différents types de caractères (position des indels, transitions empruntées dans un HMM, etc) sur des données qui sont évolutivement proches les unes des autres. Pour ce qui concerne les tests en détection, nous prenons comme ensemble de résultats positifs, et pour chaque famille « clean », l'ensemble des séquences de la famille « full » correspondante (cf. [Li *et al.*, 2006]), privé bien entendu des séquences d'apprentissage.

Parmi les 1.281 familles de classe 'A', il s'en trouve 454 possédant entre 3 et 25 taxa, dont 32 posent des problèmes de cohérence des données dans la base en ligne (Jean-Karim Hériché, EMBL Heidelberg, communication personnelle). Sur les 422 familles restantes, 13 familles possèdent une version « clean » égale à la version « full », ce qui se traduit pour nous par une absence de cibles à détecter. Nous ignorons donc ces 13 familles. Restent les 409 familles qui forment notre premier jeu de données. Les 409 ensembles d'apprentissage corespondants (familles « clean ») totalisent 7.950 séquences. Le nombre total de cibles à détecter ($\{\text{full} - \text{clean}\}$) contient 9.610 séquences.

Les recherches se font sur une base de séquences correspondant à l'ensemble des protéines intervenant dans la construction de la totalité des 1.281 familles de classe 'A' de TreeFam 7.0, c'est à dire 91.742 séquences. C'est un nombre respectable, si l'on compare avec les 71.830 entrées PDB (structures protéiques) ou avec les 533.049 séquences que comprend la base de données UniProtKB/Swiss-Prot (données de novembre 2011).

12.2 Grande distance évolutive : SABmark

La base de données SABmark [Van Walle *et al.*, 2005] est une base de données qui s'appuie sur la hiérarchie SCOP [Murzin *et al.*, 1995] de classification des protéines. L'intérêt de SABmark est de proposer des familles de protéines avec un nombre contrôlé de séquences dans chacune d'elles (au plus 25 séquences), ce qui permet d'éviter la sur-représentation des repliements de SCOP les plus fournis en protéines. SABmark tente ainsi de fournir un nombre comparable de représentants pour la totalité de la gamme des repliements répertoriés par SCOP. Chaque famille SABmark est constituée d'une clique de séquences toutes

alignées deux-à-deux en tenant compte le mieux possible des contraintes structurales qui les relient. Ivo Van Walle et coauteurs ont combiné deux approches d'alignement guidé par la structure [Boutonnet *et al.*, 1995; Shindyalov et Bourne, 1998], conjuguées à une méthode développée par les auteurs eux-mêmes [Van Walle *et al.*, 2003] pour calculer ces alignements *pairwise*.

La référence étant donnée par ces alignements *pairwise*, nous avons utilisé le logiciel T-Coffee (cf. chapitre 2) pour dériver un alignement multiple à partir des alignements deux-à-deux.

SABmark se divise en deux jeux de données. D'un côté, on trouve le jeu *Twilight Zone*, dans lequel chaque famille de séquences correspond à un repliement SCOP. Les séquences d'une même famille partagent un faible taux d'identité, soit entre 0% et 25%. Dans ce cadre, l'origine évolutive commune de deux séquences appartenant à la même famille ne peut pas toujours être établie.

De l'autre côté, le jeu de données *Superfamily* recense des familles qui correspondent chacune à une superfamille SCOP. Les distances évolutives au sein d'une même famille sont plus modérées (entre 25% et 50% d'identité). C'est ce jeu de données que nous utilisons ici. Dans SABmark 1.65, il se compose de 425 groupes de séquences. Dans chaque groupe, la référence d'alignement structurel est constituée par des alignements deux-à-deux (chacun des groupes forme une clique dont les sommets sont les séquences et les arêtes les alignements *pairwise*). La construction complète de nos familles d'apprentissage se fait ici en deux étapes :

1. alignement multiple obtenu avec T-Coffee, en donnant comme librairie en entrée l'ensemble des alignements *pairwise*,
2. construction d'un arbre phylogénétique au maximum de vraisemblance avec PhyML 3.0 (loi Gamma à 4 classes de vitesse, modèle substitutionnel LG [Le et Gascuel, 2008]).

Parmi les 425 groupes de séquences, 3 groupes (n^o 169, 353 et 357) sont constitués de séquences (respectivement 3, 3 et 4 séquences) sur lesquelles *aucun* des alignements de référence ne contient deux acides aminés alignés (les résidus d'une séquence sont tous alignés contre des gaps dans l'autre séquence, et vice-versa). La référence étant inexistante, nous choisissons d'ignorer ces trois groupes plutôt que de nous en servir pour construire des alignements *et* des arbres phylogénétiques erronés.

Le test auquel nous allons soumettre ces familles en termes de reconnaissance des séquences homologues va consister à tenter de retrouver à partir des modèles construits sur un groupe de séquences, toutes les *autres* séquences appartenant à la même superfamille

dans la base SCOP correspondante (c'est-à-dire SCOP version 1.65). Sur les 422 familles restantes, deux familles SABmark (n° 401 et 417) incluent déjà la totalité de leur superfamille dans l'ensemble d'apprentissage, donc n'ont pas de positifs identifiés. On ignore donc également ces deux familles, et notre banc de test SABmark contient finalement 420 groupes (ou familles) de séquences homologues.

Les 420 groupes que nous utilisons totalisent 3.263 séquences d'apprentissage, et 39.160 positifs à retrouver dans une base SCOP 1.65 comportant 51.828 séquences. Les 420 superfamilles correspondant à notre banc de test couvrent donc bien une grande partie de la base SCOP.

CHAPITRE

13 Phylogénisation des architectures de HMM

Dans cette partie, nous faisons l'hypothèse que deux séquences proches dans une phylogénie données auront des patterns similaires d'insertion et de délétion par rapport à une référence donnée. Cette hypothèse mérite d'être testée : dans quelle mesure peut-on dire qu'une phylogénie établie en général sur des alignements d'acides aminés, est également explicative pour ce qui concerne les positions de gap dans le même alignement ? De tels tests sont exposés dans la section qui suit, avant de montrer combien la phylogénisation des architectures améliore les performances des HMM profils.

Sommaire

13.1 Tests de corrélation entre patterns d'insertion/délétion et phylogénie

Nous testons dans un premier temps chaque couple formé d'un arbre phylogénétique et de l'alignement de séquences correspondant, afin de déterminer si la topologie et les longueurs de branche de l'arbre sont cohérentes avec le signal phylogénétique induit par les taxa aux feuilles.

Sur des données discrètes telles que les caractères G/N représentant respectivement la présence et l'absence d'un gap en une position donnée dans une séquence donnée, nous menons un test de randomisation. L'idée est que si un signal phylogénétique est clairement présent dans l'arbre, alors toute permutation des données sur les différentes feuilles

de l'arbre doit logiquement mener à une vraisemblance plus faible. Un arbre binaire avec n feuilles induisant $n!$ permutations, on teste au hasard 100 situations différentes (si l'arbre possède 3 ou 4 espèces, on teste exhaustivement toutes les permutations, soit respectivement 5 ou 23 permutations différentes de l'affectation initiale des taxa aux feuilles). On dira que le test de robustesse est positif si et seulement si au moins 95% des permutations ont engendré une vraisemblance inférieure à la vraisemblance de départ. Après chaque permutation et avant de calculer la vraisemblance correspondante, on prend soin de recalculer le paramètre optimal de dilatation globale des branches de l'arbre ainsi que le paramètre optimal de forme de la loi gamma utilisée pour autoriser la variabilité des vitesses d'évolution le long de l'alignement.

Sur les 422 arbres « clean » du jeu de données originel TreeFam, 386 passent le test avec succès (soit 91,5%) et 36 y échouent (8,5%). Cela signifie donc qu'à ce niveau d'éloignement phylogénétique, les patterns Gap/Nogap respectent globalement la phylogénie support des séquences.

Sur les 420 familles du banc de test SABmark, seules 263 passent le test avec succès (soit 62,6%) alors que 157 y échouent (37,4%) : l'arbre phylogénétique construit soutient moins fortement l'idée de la présence d'un signal phylogénétique dans les patterns Gap/Nogap. Le succès ou l'échec au test de randomisation est corrélé au nombre de séquences composant la famille. En effet, le résumé statistique de la distribution du nombre de séquences parmi les familles sur lesquelles le test échoue est le suivant :

minimum	1er quartile	médiane	moyenne	3e quartile	maximum
3	3	4	4,72	5	19

alors que la même distribution sur les familles qui réussissent le test de randomisation donne :

minimum	1er quartile	médiane	moyenne	3e quartile	maximum
3	4	7	9,59	12	25

Ceci implique que l'espérance du nombre de séquences pour les familles ayant échoué au test est significativement inférieure à celle correspondant aux familles ayant passé le test avec succès (p-valeur égale à $2,2.10^{-16}$ dans le test de Student d'égalité des moyennes).

Ce phénomène s'explique en partie au moins par le fait que les algorithmes de construction d'arbre à partir de séquences alignées ne prennent pas en compte de façon correcte la présence des gaps. Soit ils suppriment à l'avance les colonnes comportant des gaps, soit ils traitent ceux-ci comme des données manquantes et les remplacent par le caractère 'X', un joker reconnaissant n'importe quel acide aminé [Yang, 1997; Guindon et Gascuel, 2003]. Cette caractéristique induit un comportement qui peut sembler surprenant. Considérons par exemple un site `A--` issu d'un alignement. Considérons un arbre

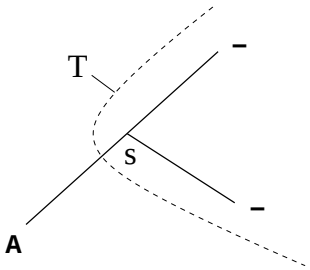


Figure 13.1. *Un arbre phylogénétique élémentaire construit sur un site comportant deux gaps*

binaire non raciné construit sur ce site, par exemple l'arbre représenté en figure 13.1. Dans cet arbre, l'algorithme de pruning de Felsenstein donne une vraisemblance partielle égale à 1 pour le sous-arbre T, et ce quels que soient le caractère inféré au nœud s et les longueurs de branche. En effet, les vraisemblances partielles sont toutes égales à 1 pour les sous-arbres réduits à une feuille qui porte un gap. On calcule alors aisément que dans ce cadre, la vraisemblance est égale à $\pi(A)$, c'est-à-dire à la probabilité de trouver l'acide aminé A dans la distribution de fond. Le site en question présentant la même vraisemblance quel que soit l'arbre, il ne participe aucunement au processus de détermination de la meilleure topologie et des longueurs de branche au maximum de vraisemblance. Ainsi, si nous considérons l'alignement représenté en figure 13.2, l'arbre construit sera semblable à celui représenté à droite de l'alignement. La topologie de cet arbre doit tout à la proximité de la tyrosine et du tryptophane, et rien à la longue insertion qui précède. Or, si l'alignement est fiable, on peut légitimement penser que les séquences 1 et 2 sont plus proches l'une de l'autre que de la séquence 3.

seq1	-	-	-	-	-	-	W
seq2	-	-	-	-	-	-	A
seq3	A	A	A	A	A	A	Y

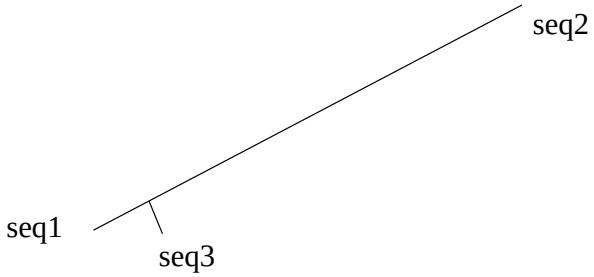


Figure 13.2. *Ci-dessus à droite, l'arbre phylogénétique construit au maximum de vraisemblance sur les séquences de gauche. Les longueurs de branche sont calculées uniquement en fonction du dernier site de l'alignement, ce qui rapproche seq1 de seq3 nonobstant le long motif de gaps partagé entre seq1 et seq2.*

On observe à plusieurs reprises ce type de situations dans le jeu de données SABmark,

et le phénomène est d'autant plus visible que les séquences sont peu nombreuses, la méthode de construction de l'alignement multiple avec l'objectif de cohérence par rapport aux alignements deux-à-deux donnant par exemple relativement souvent des colonnes composées de deux gaps dans un alignement de trois séquences. La phylogénie étant construite en ignorant de telles colonnes, il n'est pas surprenant de constater que la corrélation entre l'arbre obtenu et les motifs Gap/Nogap n'est pas très élevée.

13.2 Phylogénisation des architectures seulement

13.2.1 Sur le banc de test SABmark

Sur chacun des 420 groupes « utiles » du banc de test SABmark, on effectue l'inférence phylogénétique de l'architecture du HMM en chacun des nœuds de la phylogénie support. Pour chaque architecture décidée, le reste du processus de construction du HMM est entièrement géré par HMMER3.0, c'est-à-dire que les autres paramètres des différents HMM sont tous calculés sans connaissance de la phylogénie, comme le fait habituellement HMMER : en mêlant observations et connaissances a priori, à partir de l'alignement seul.

Chaque groupe SABmark donne donc lieu à la construction d'un certain nombre de HMM profils, un par nœud de la phylogénie support (on considère aussi bien les nœuds internes que les feuilles, ce qui peut poser question. Voir à ce sujet la discussion en fin de chapitre). Chacun de ces HMM profils est ensuite utilisé pour rechercher des protéines homologues dans la base dont sont extraites les séquences de SABmark 1.63, c'est à dire SCOP 1.63. On dira ensuite que *les résultats donnés par le phylo-HMM correspondant à ce groupe* sont constitués par la liste ordonnée des résultats de séquence, elle-même construite en ne retenant pour chaque séquence que *le score maximal renvoyé par la famille de HMM profils pour cette séquence.*

Parallèlement, chaque groupe SABmark donne un HMM standard construit par HMMER3.0 sur la base de l'alignement correspondant (rappelons que pour la base SABmark, ledit alignement est formé par T-Coffee à partir des alignements deux-à-deux de référence).

À partir de là, plusieurs points sont à évaluer pour comparer les performances des HMM profils à l'architecture calculée phylogénétiquement, par rapport à l'approche consistant à utiliser le HMM standard construit par HMMER3.0 :

– on peut évaluer les scores donnés à chacune des cibles à prédire par les HMM correspondants (phylogénisés ou non). On rappelle que chaque score est le logarithme du rapport de la vraisemblance donnée par le HMM par rapport à celle issue du mo-

dèle nul, ce dernier étant un simple état bouclant sur lui-même comme exposé en section 3.3.4, page 51. On évalue alors la vraisemblance des modèles par rapport aux vrais positifs,
– on peut également évaluer la capacité des modèles construits à bien discriminer résultats positifs et négatifs : le modèle idéal donnant une liste de résultats (naturellement triés par ordre de scores décroissants) présentant *d'abord* tous les résultats positifs (c'est-à-dire les séquences effectivement homologues aux séquences sur lesquelles le modèle est construit) *puis* tous les résultats négatifs, on donnera une idée de la puissance du modèle en présentant des courbes ROC (*Receiver Operating Characteristics*, voir plus loin).

Résultats en termes de vraisemblance des cibles

En représentant en ordonnée la différence de score entre HMM phylogénisé et HMM standard (positive si le HMM phylogénisé donne un meilleur score que le HMM standard et négative dans le cas inverse) contre le score du HMM standard en abscisse, on obtient le graphique présenté en figure 13.3. Sur les 39.160 cibles que compte le banc de test, 32.598 ont une vraisemblance plus forte dans le HMM phylogénisé que dans le HMM standard correspondant à leur groupe (soit 83,2%) contre 5.336 (soit 13,6%) pour lesquelles le HMM standard donne une vraisemblance plus élevée que le HMM phylogénisé. Les 3% restants ont une vraisemblance qui ne varie pas sensiblement.

Résultats en termes de détection

Sur le banc de test SABmark, lorsqu'on fusionne puis ordonne tous les listes de score des cibles issues des différentes familles, on obtient les courbes ROC présentées en figures 13.4 et 13.5.

13.2.2 Sur le banc de test TreeFam

Résultats en termes de vraisemblance des cibles

Les résultats sont présentés en figure 13.6 : en abscisse figure le score obtenu par la cible dans le HMM standard, et en ordonnée la différence entre le score obtenu par l'approche des phylo-HMM et celui obtenu dans le HMM standard. On voit que dans la grande majorité des cas, les modèles HMM phylogénisés donnent une plus grande vraisemblance aux cibles.

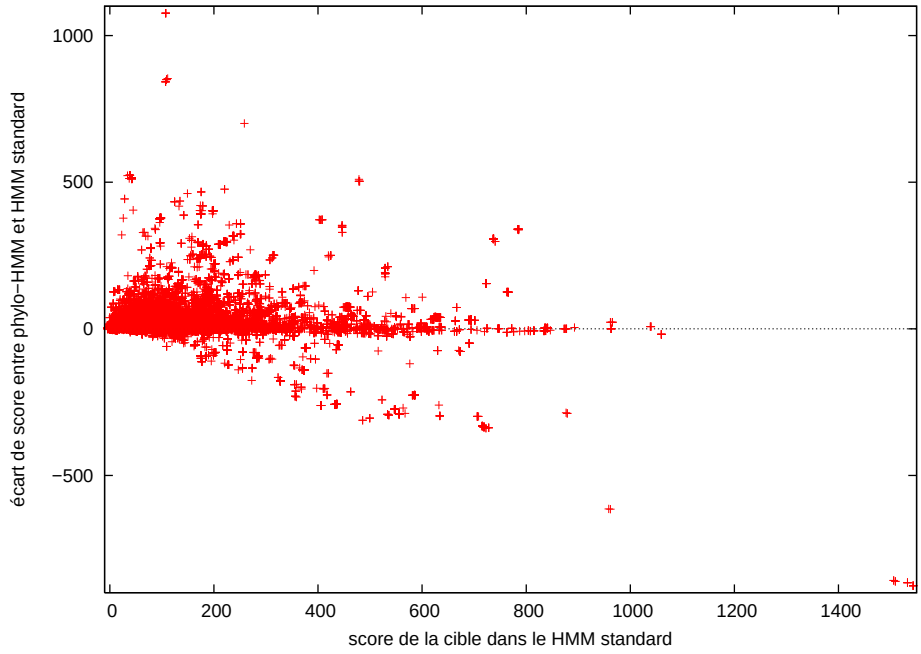


Figure 13.3. *Différence de score entre HMM standard et HMM phylogénisé (phylogénisation de l'architecture seulement), pour chacune des 39.160 cibles correspondant au banc de test SABmark*

Résultats en termes de détection

On présente en figure 13.7 les résultats obtenus par la phylogénisation de l'architecture des HMM profils sur le banc de test TreeFam. On y constate que les résultats obtenus par les HMM issus du processus de phylogénisation ne se distinguent pas beaucoup de ceux des HMM standard. Alors même que les gains en vraisemblance pour les cibles à prédire sont avérés (figure 13.6), les HMM phylogénisés ajoutent des résultats négatifs (scores élevés donnés à des séquences non homologues) en plus des cibles correctement prédites. On verra plus loin comment on peut contourner cette difficulté de façon simple.

13.2.3 Discussion

L'effet de la phylogénisation des architectures de HMM semble positif, du moins si l'on en juge sur le banc de test SABmark. Cependant, on relève une faiblesse amenée par la méthode de phylogénisation lorsque le nœud d'intérêt se trouve être une feuille. Le processus de détermination de l'architecture ignore alors le contenu de la feuille en question elle-même (cf. section 6.2, page 121). Cela se traduit en une perte d'information préjudiciable par rapport au HMM standard. Il serait sans doute plus pertinent de ne pas phylogéniser

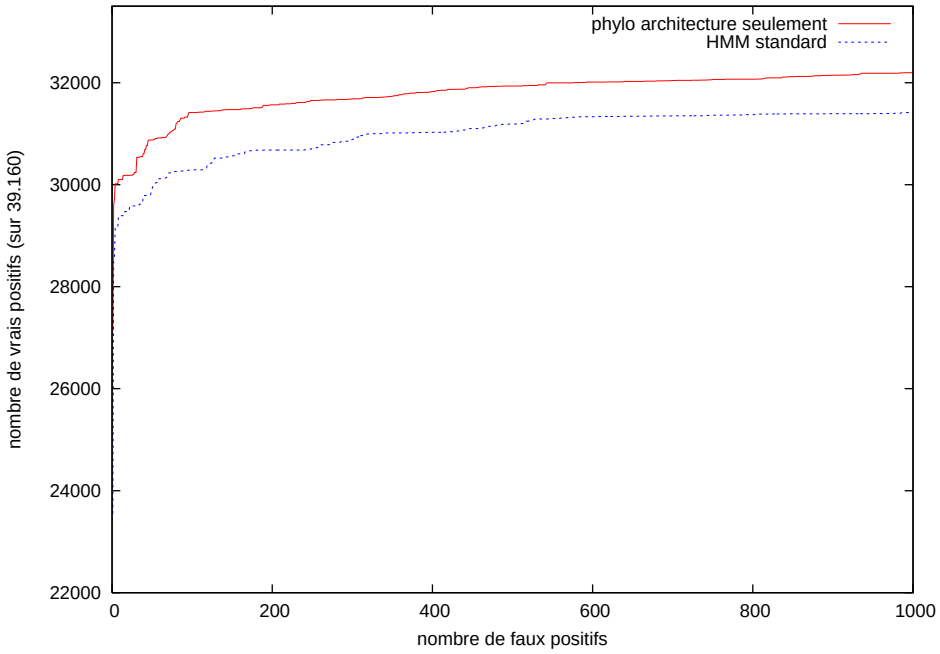


Figure 13.4. *Courbe ROC pour le banc de test SABmark, toutes familles confondues, phylogénisation de l'architecture seulement*

systématiquement et aveuglément sur tous les nœuds, mais par exemple de choisir un point médian sur la branche reliant une feuille à son père, plutôt que la feuille elle-même. Ce même raisonnement peut d'ailleurs se généraliser pour éviter de phylogéniser sur des nœuds distants de tous leurs voisins, car la phylogénisation sur de tels nœuds nous fait immanquablement tendre vers des modèles fabriqués « au hasard », qui doivent plus aux fréquences de fond des processus de substitution qu'aux observations aux feuilles.

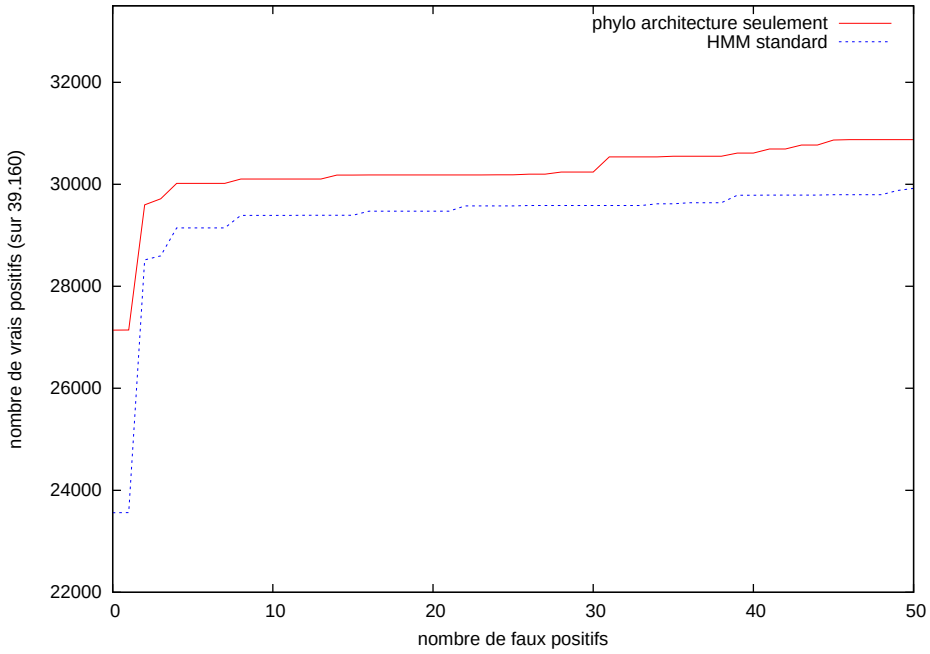


Figure 13.5. *Courbe ROC pour le banc de test SABmark, toutes familles confondues, phylogénisation de l'architecture seulement (zoom)*

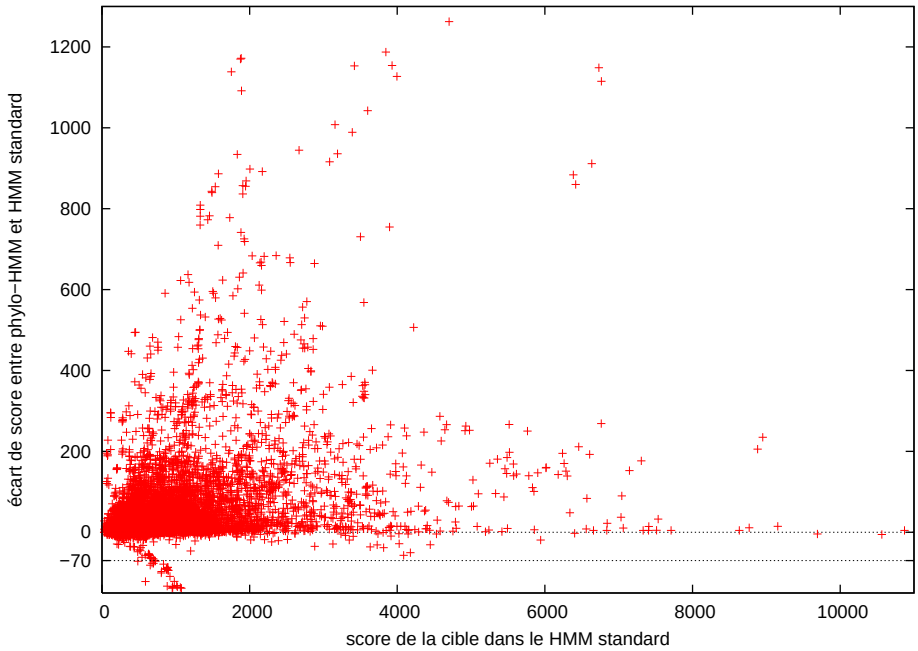


Figure 13.6. *Accroissement de la log-vraisemblance des cibles à détecter pour le banc de test TreeFam, toutes familles confondues, phylogénisation de l'architecture seulement. Les 28 cibles situées en deçà de la droite d'équation $y = -70$, en bas à gauche du graphique, appartiennent pour 26 d'entre elles à une même famille, la famille TF105249. Au sein de cette famille, ces séquences cibles sont exactement identiques à l'une au moins des séquences d'apprentissage (nommément aux séquences* Dynactin subunit 4 *de numéros d'accession UniProt Q9UJW0, Q9QUR2 et Q5R7U7). La méthode de phylogénisation utilisée présente la caractéristique de donner généralement des scores plus faibles aux séquences d'apprentissage elles-mêmes, par rapport à ce que fait le HMM standard. En effet, le modèle phylo-HMM construit sur une feuille de l'arbre ne* contient pas *l'information liée à la séquence portée par cette feuille, mais seulement une* estimation *de cette information basée sur la phylogénie. Ce n'est pas à proprement parler l'objectif de nos phylo-HMM que de donner de meilleurs scores aux séquences d'apprentissage elles-mêmes. On retrouve la même configuration pour les deux cibles restantes, issues de la famille TF105911, chacune des deux séquences cibles se retrouvant également dans l'ensemble d'apprentissage correspondant.*

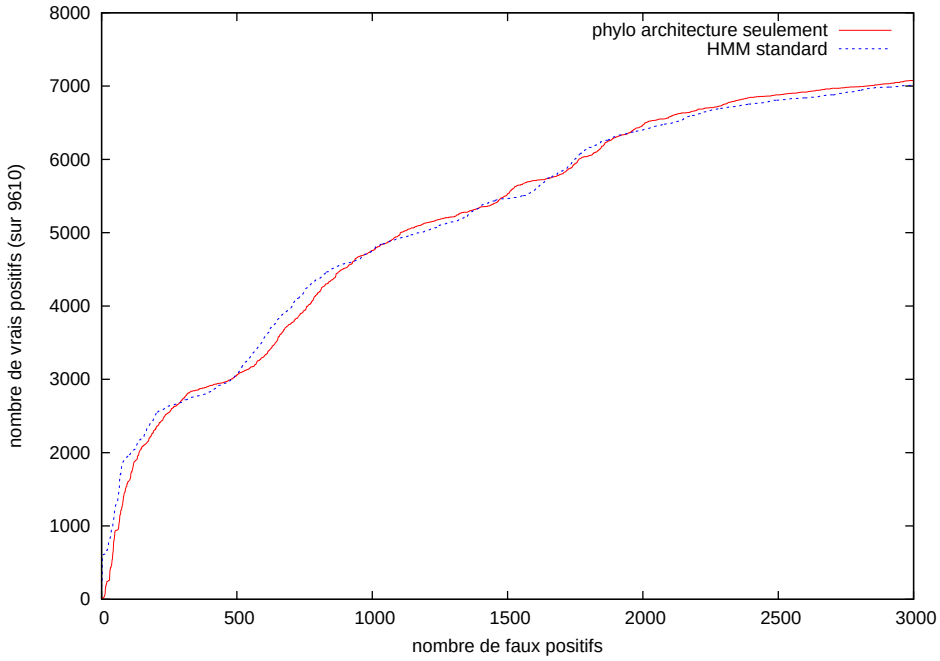


Figure 13.7. *Courbe ROC pour le banc de test TreeFam, toutes familles confondues, phylogénisation de l'architecture seulement*

CHAPITRE

14 Phylogénisation de l'architecture et des états Match des HMM

Dans cette section, nous détaillons les résultats obtenus via l'application de la méthode de dérivation phylogénétique des paramètres du HMM à deux types de paramètres. En effet, on procède en utilisant à la fois la méthode exposée au chapitre 7 et celle présentée au chapitre 8 : en chaque nœud de la phylogénie support, les colonnes à modéliser par un état Match sont déterminées par un processus d'inférence phylogénétique, de même que les distributions correspondant aux émissions d'acides aminés sur ces états Match. Le reste des paramètres des HMM est calculé par HMMER, sans prise en compte des corrélations phylogénétiques. Comme précédemment, la phylogénisation aboutit à la création de $(2n-2)$ HMM pour n séquences d'apprentissage. Le score d'une séquence quelconque dans le « phylo-HMM » correspondant à une famille donnée est le meilleur des scores obtenus dans les différents HMM construits sur les séquences d'apprentissage de ladite famille.

Sommaire

14.1 Résultats sur le banc de test SABmark

14.1.1 Vraisemblance des cibles positives

Nous rappelons que nous avons pour ce jeu de données, 420 familles couvrant une grande partie de l'étendue de représentation des protéines couverte par SCOP. En tout, 39.160 cibles positives sont à identifier par les différents HMM. On représente sur le

graphique en figure 14.1 l'accroissement du score de chacune des cibles que la phylogénisation apporte par rapport au modèle HMM standard. On a donc un nuage de 39.160 points, ceux situés en dessous de l'axe des abscisses correspondant à des cibles pour lesquelles le HMM phylogénisé donne une vraisemblance *plus faible* que le HMM standard.

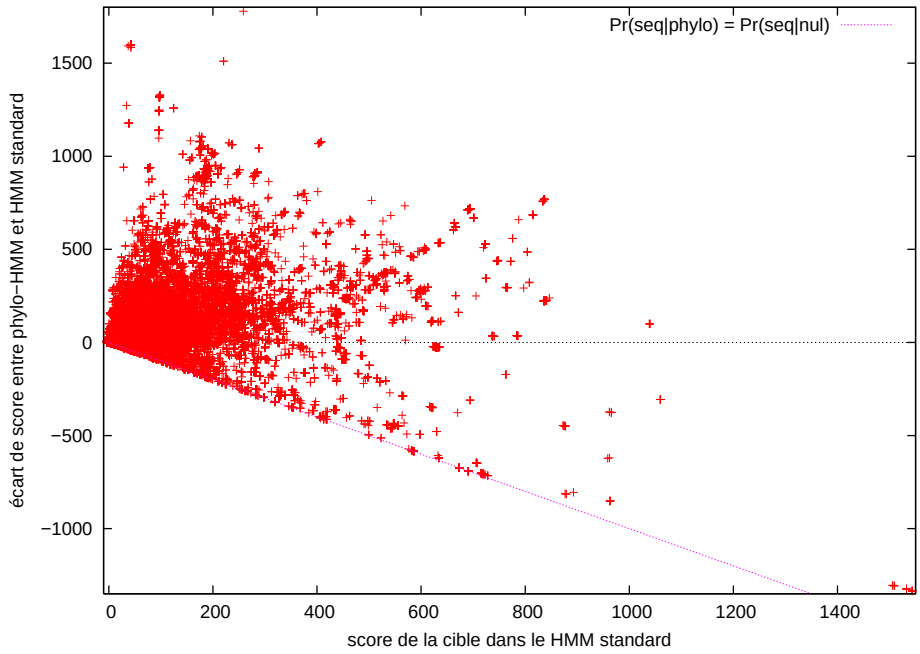


Figure 14.1. *Résultats en termes d'accroissement de la log-vraisemblance des cibles sur le banc de test SABmark, phylogénisation de l'architecture et des émissions sur états Match. En pointillés violet, la droite d'équation $y = -x$ correspondant à des HMM phylogénisés donnant des vraisemblances aussi médiocres que le modèle nul.*

La première constatation est que le centre de gravité du nuage de points se trouve bien au-dessus de l'axe des abscisses : 31.908 séquences voient leur vraisemblance augmenter via la phylogénisation des modèles, contre 7.194 pour lesquelles la vraisemblance diminue. On remarque ensuite une linéarité surprenante autour de la droite d'équation $y = (-x)$. L'ordonnée du point correspondant à la séquence s étant égale à :

$$\log\left(\frac{\Pr(s|\text{phylo})}{\Pr(s|\text{nul})}\right) - \log\left(\frac{\Pr(s|\text{standard})}{\Pr(s|\text{nul})}\right) = \log\left(\frac{\Pr(s|\text{phylo})}{\Pr(s|\text{standard})}\right) \quad , \tag{14.1}$$

cette droite correspond aux séquences s vérifiant :

$$\log\left(\frac{\Pr(s|\text{phylo})}{\Pr(s|\text{standard})}\right) = -\log\left(\frac{\Pr(s|\text{standard})}{\Pr(s|\text{nul})}\right) \quad , \tag{14.2}$$

c'est-à-dire finalement : Pr(s|phylo) = Pr(s|nul).

Pourquoi obtient-on de si mauvais résultats sur certaines cibles ? En analysant les familles d'apprentissage dont proviennent ces résultats, on trouve à plusieurs reprises la situation suivante : une famille d'apprentissage très hétérogène, dans laquelle on trouve au moins une branche très longue et une feuille au bout de la branche en question. Une telle configuration est représentée par exemple en figure 14.2. La feuille dessinée en rouge correspond à la séquence très dissemblable par rapport aux deux autres. Le HMM reconstitué par le processus de phylogénisation sur cette feuille en bout de branche longue est inféré à partir de la connaissance des séquences situées au bout des deux autres feuilles, mais le signal de celles-ci est effacé par la très grande distance qui l'en sépare. Sur ce nœud, le HMM issu de la phylogénisation correspond donc à un HMM construit essentiellement sur la base des fréquences de fond des acides aminés. Ceci explique que les scores qu'il renvoie ne se distinguent que peu des scores donnés par le HMM nul, qui est un simple état bouclant sur lui-même en émettant des acides aminés selon les fréquences de fond du HMM, proches de celles du processus LG utilisé pour l'inférence phylogénétique (cf. page 51).

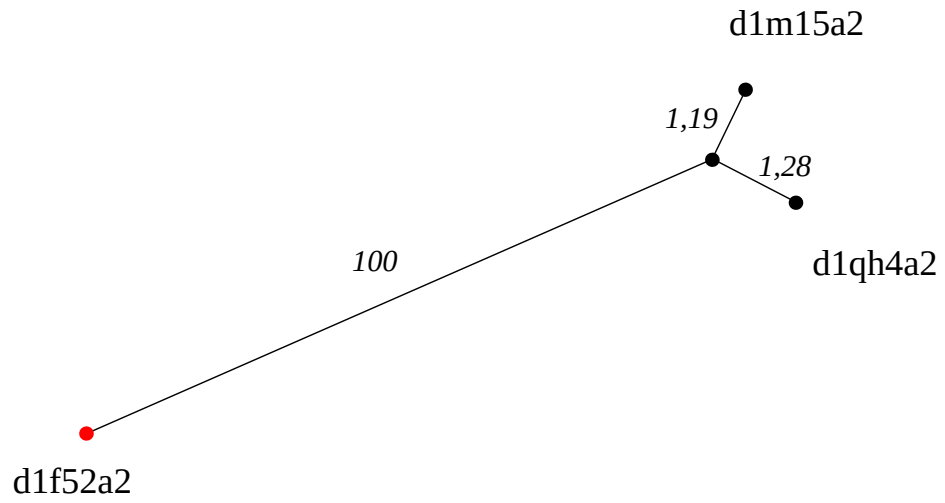


Figure 14.2. *Arbre construit par PhyML sur la base des trois séquences du groupe n° 323 du banc de test SABmark : l'une des trois séquences étant très dissemblable des deux autres, elle est rejetée à l'infini dans l'arbre (100 est une valeur plafond pour les longueurs de branche lors du processus d'optimisation d'arbre par PhyML).*

Le problème que pose une telle situation est celui de la perte pure et simple de l'information liée à la ou les séquence(s) singulière(s), nommément la séquence *d1f52a2* pour ce qui concerne la famille représentée en figure 14.2. Alors que le HMM standard parvient

à retenir cette information, le HMM phylogénisé l'ignore. Les cibles se trouvant dans le voisinage phylogénétique des séquences singulières ne sont donc pas bien modélisées par les HMM phylogénisés, et recoivent donc un score bien meilleur par le HMM standard.

On peut donc se dire que le problème sera absent ou moins fréquent dans les arbres de faible taille (la taille d'un arbre étant la somme des longueurs des branches qui le composent). On constate en réalité (figure 14.3) que les « mauvais résultats » apparaissent même au sein d'arbres de taille modérée, alors que de bons résultats continuent d'être observés dans des arbres de grande taille : tous les HMM phylogénisés issus d'une famille donnée ne sont pas à discréditer simultanément. Par exemple, dans la situation représentée en figure 14.2, les trois nœuds de droite donnent de bons résultats de vraisemblance sur une partie des cibles positives de cette famille. Le fait d'abandonner purement et simplement les HMM phylogénisés issus de nœuds isolés par rapport au reste de l'arbre semble une bonne idée mais cela ne résoud pas le problème : les séquences singulières reçevant alors leur meilleur score d'un HMM phylogénisé sur un nœud situé en un endroit diamétralement opposé de la phylogénie, ce score ne saurait être élevé.

La solution pourrait être apportée par l'utilisation d'une nouvelle stratégie de détermination des scores : pour déterminer le score d'une séquence dans le modèle phylogénisé, on calcule le score renvoyé pour cette séquence par chacun des HMM de nœud issus de la phylogénisation, mais également le score renvoyé par le HMM standard. On retient ensuite comme score de la séquence dans le modèle phylogénisé le meilleur des différents scores obtenus, en incluant donc le score donné par le HMM standard. On ne retient donc les scores fournis par les HMM phylogénisés que lorsqu'il se trouve au moins l'un d'entre eux pour fournir un score supérieur à celui du HMM standard. Bien sûr, dans le cas où l'un des HMM de nœud donne un score plus élevé que celui donné par le HMM standard à une séquence qui n'est pas homologue, alors ce score plus élevé est conservé : le jeu n'est pas pipé, et les HMM phylogénisés sont à égalité avec le HMM standard, apportant potentiellement aussi bien de vrais et de faux positifs.

Ainsi, on peut espérer que les cibles proches de séquences isolées au sein de l'ensemble d'apprentissage recevront un score correct par le HMM standard, tandis que les séquences moins isolées seront détectées convenablement par un HMM phylogénisé. Dans la suite, on précisera systématiquement la stratégie de *scoring* utilisée. La seconde stratégie (incluant les résultats du HMM standard) s'impose sur le banc de test SABmark, mais elle n'est pas nécessaire pour obtenir de bons résultats sur le banc de test TreeFam.

14.1.2 Résultats en termes de détection

On superpose les courbes ROC de détection des cibles positives dans les deux cas, selon que l'on intègre ou non dans le processus de traitement des résultats des phylo-HMM ceux des HMM standard. Cf. figures 14.4 et 14.5. On constate que le processus de phylogénisation seul ne parvient pas à détecter plus de cibles positives que les HMM standard lorsqu'on ignore les résultats donnés par ceux-ci lors du processus de détermination du meilleur score d'une protéine. En revanche, si l'on retient les résultats des HMM standard et qu'on enrichit ceux-ci avec ceux en provenance des HMM phylogénisés, le gain en détection apparaît clairement (environ 1000 positifs supplémentaires détectés pour le même nombre de résultats négatifs).

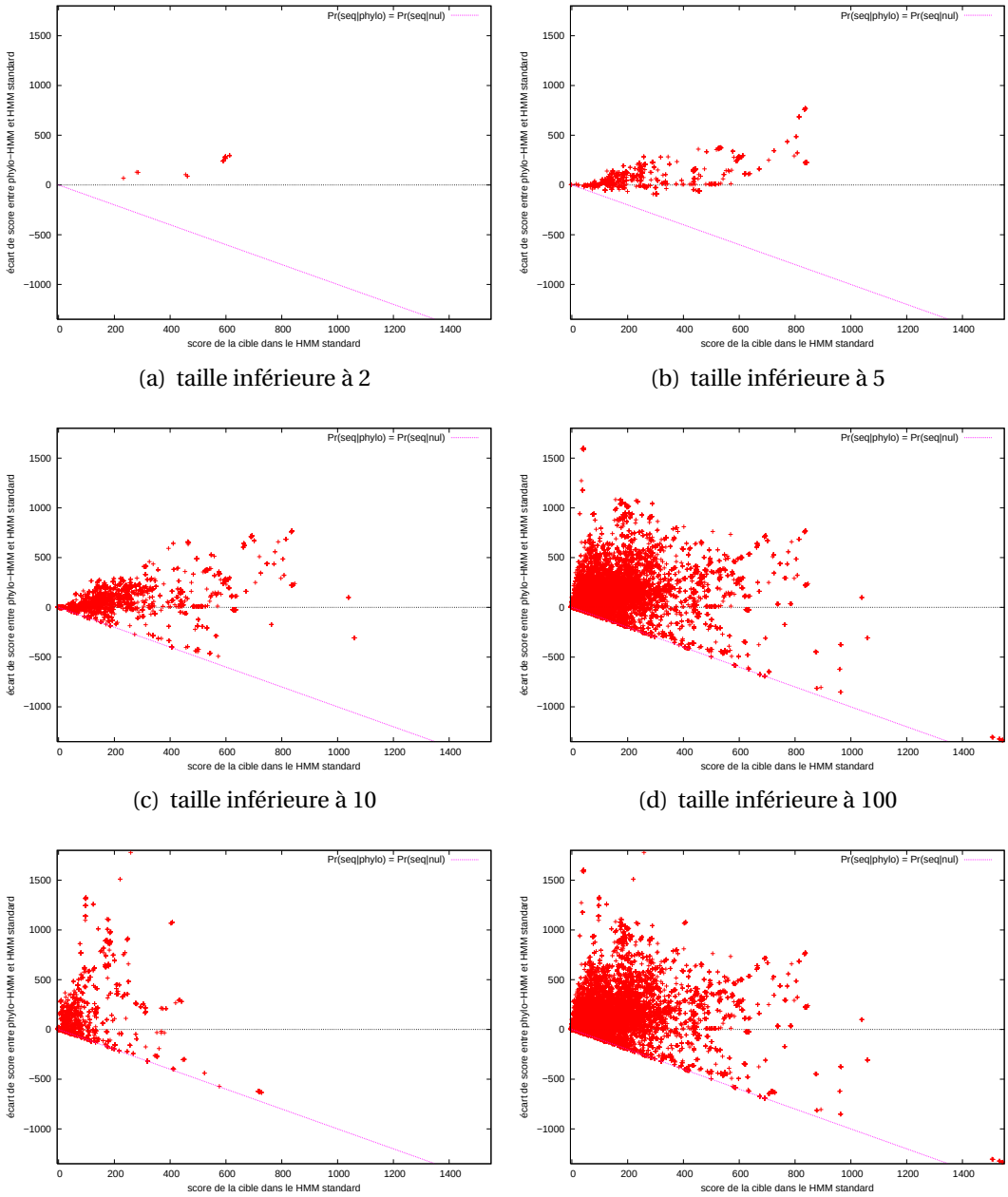
14.2 Sur le banc de test TreeFam

14.2.1 Vraisemblance des cibles

Les résultats obtenus sont montrés en figure 14.6 : par rapport à la phylogénisation de l'architecture seulement, on obtient une belle augmentation de la vraisemblance des cibles, avec peu de cibles pour lesquelles la vraisemblance diminue dans le HMM phylogénisé par rapport au HMM standard. On observe également une certaine linéarité : l'accroissement du score d'une séquence semble grossièrement proportionnel au score obtenu par celle-ci dans le HMM standard.

14.2.2 Résultats en termes de détection

Les résultats obtenus en termes de détection des cibles positives sont montrés en figure 14.7. On observe une très forte amélioration des résultats par rapport à la situation dans laquelle on ne phylogénise que l'architecture des HMM, ceci sans même intégrer les résultats du HMM standard dans le processus de calcul des scores des séquences, ce qui est remarquable.



(a) taille inférieure à 2

(b) taille inférieure à 5

(c) taille inférieure à 10

(d) taille inférieure à 100

(e) taille *supérieure* à 100

(f) toutes tailles d'arbre confondues

Figure 14.3. *Déclinaison des résultats d'accroissement de la vraisemblance des cibles pour le banc de test SABmark, en fonction de la taille de l'arbre construit sur la famille correspondant à chacune des cibles. La phylogénisation porte sur les architectures des HMM et sur les émissions d'acides aminés issues des états Match.*

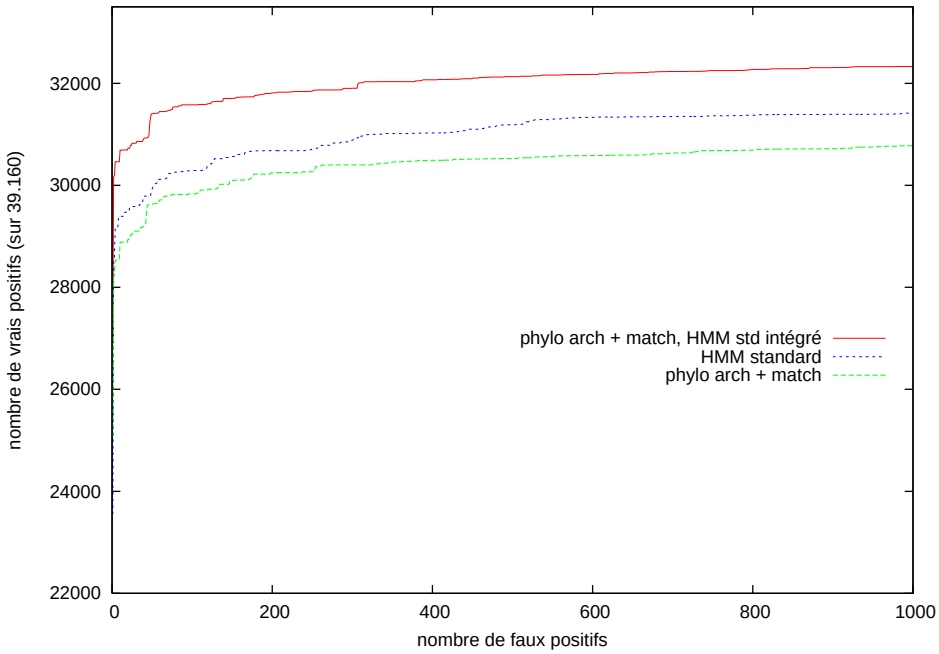


Figure 14.4. *Courbes ROC sur le banc de test SABmark, phylogénisation de l'architecture et des émissions sur les états Match. Sans l'inclusion des résultats du HMM standard, la phylogénisation des émissions sur les états Match en plus de celle des architectures* dégrade *les performances en détection, ceci alors même que la phylogénisation des architectures apportait à elle seule un gain en détection. Les phylo-HMM, avec des distributions d'émission a priori plus informatives, ratent donc parfois leur cible en amplifiant éventuellement du bruit (par exemple lorsqu'un état Match est mal choisi, suivi d'une détermination phylogénétique de la distribution des émissions sur cet état calculée sur trop peu de caractères, etc). On peut constater cependant en figure 14.5 que la phylogénisation donne des performances supérieures à celles du HMM standard dans la zone correspondant aux très faibles taux d'erreur.*

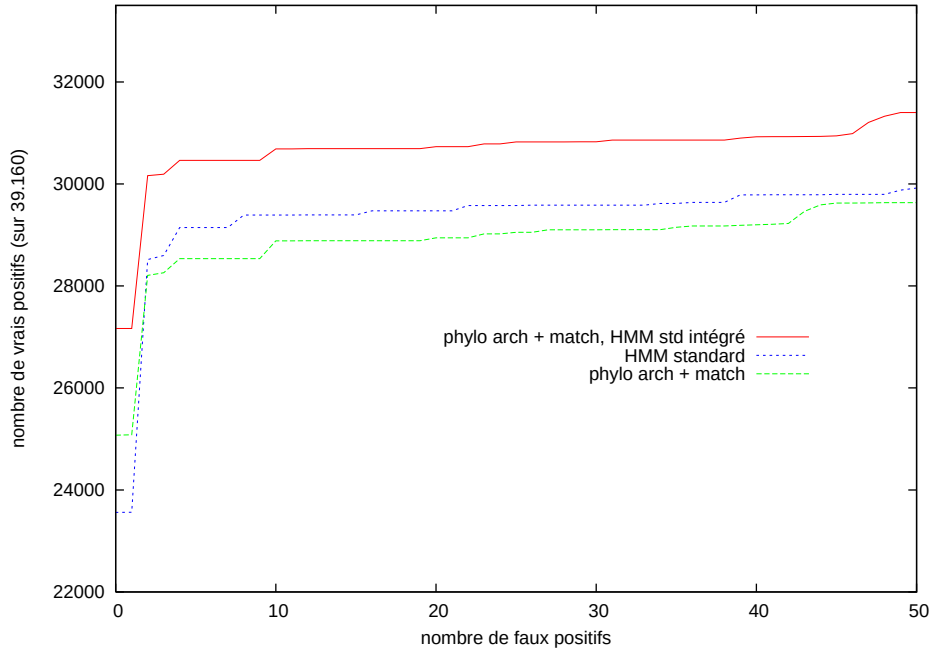


Figure 14.5. *Courbes ROC sur le banc de test SABmark, phylogénisation de l'architecture et des émissions sur les états Match (zoom). On constate que les phylo-HMM donnent par rapport aux HMM standard un nombre significativement plus grand de cibles avec un score supérieur à celui du premier faux positif (+1513 vrais positifs supplémentaires). Les résultats s'inversent cependant très vite au profit du HMM standard, ce qui illustre la nécessité de ne retenir le score donné à une cible par un HMM phylogénisé que lorsqu'il est* supérieur *à celui donné par le HMM standard (stratégie correspondant à la courbe dessinée ici en rouge).*

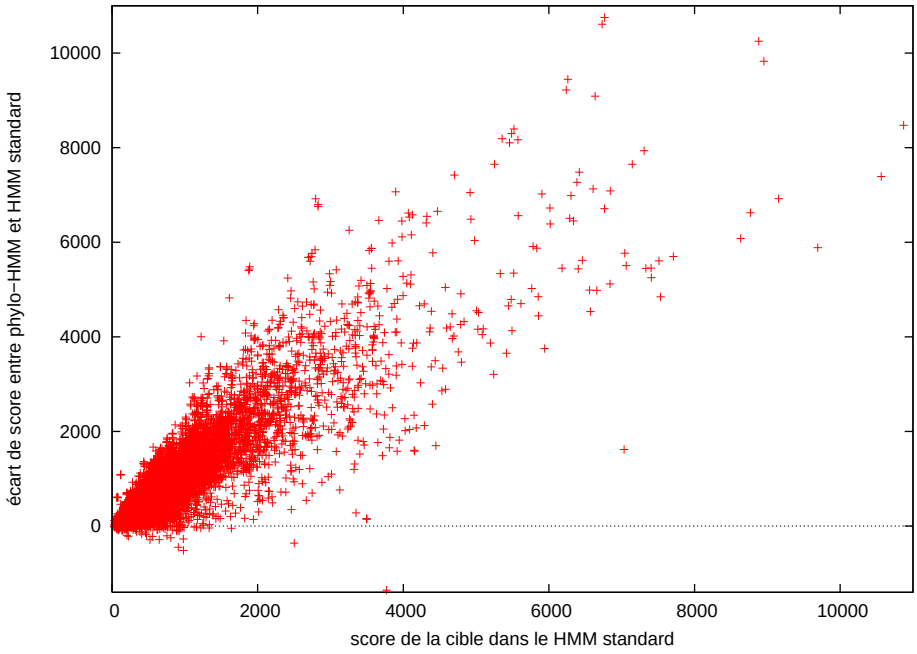


Figure 14.6. *Accroissement de la log-vraisemblance des cibles positives sur le banc de test TreeFam, phylogénisation de l'architecture et des émissions sur les états Match*

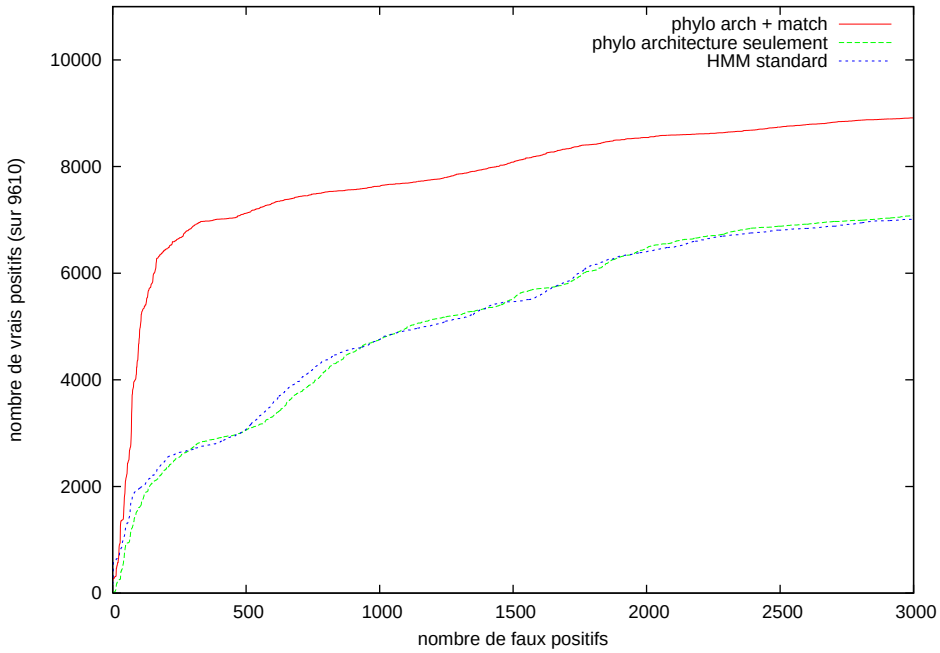


Figure 14.7. *Courbes ROC sur le banc de test TreeFam, phylogénisation de l'architecture et des émissions sur les états Match. Les scores donnés par le HMM standard ne sont pas pris en compte lors de la détermination du score dans le modèle phylogénisé.*

CHAPITRE

15 Apprentissage des processus de substitution entre transitions

Sommaire

15.1 Rappel des processus utilisés par Qian et Goldstein

Dans [Qian et Goldstein, 2003], Bin Qian et Richard Goldstein avaient développé l'idée des tree-HMM de Mitchison et Durbin [Mitchison et Durbin, 1995; Mitchison, 1999] en s'affranchissant des contraintes sur les processus que Mitchison et Durbin avaient énoncées à partir d'une interprétation à notre avis erronée du caractère phylogénétiquement explicatif des substitutions entre transitions (cf. notre critique en section 9.3). Néanmoins, Qian et Goldstein avaient fait le choix de conserver *deux processus distincts*, l'un pour les transitions quittant les états Match et l'autre pour les transitions quittant les états de Délétion.

Qian et Goldstein avaient appris leurs deux processus de substitution à partir d'alignements de séquences protéiques issus d'une base de données d'alignements structurels, *Combinatorial Extension* (http://cl.sdsc.edu/ce.html, [Shindyalov et Bourne, 1998]). Qian et Goldstein avaient extrait de cette base 10 alignements de 30 séquences chacun, chaque alignement étant composé d'une séquence dite « représentative » et des vingt-neuf séquences les plus distantes tout en étant structurellement voisines de la séquence représentative. À partir de chacun des 10 alignements, les auteurs inféraient des topologies

d'arbre à l'aide de MOLPHY [Adachi *et al.*, 1996] puis choisissent la meilleure topologie et déterminent les longueurs de branche en utilisant PAML [Yang, 1997]. Après quoi les matrices de substitution entre transitions sont calculées de manière à optimiser les vraisemblances, par une méthode du simplexe.

Figure 15.1. *Les processus de substitution entre transitions calculés et utilisés par [Qian et Goldstein, 2003]. On représente ici les taux instantanés de substitution q_{ij}. Les arcs non représentés correspondent à des taux nuls. Les boucles sur chacun des états sont implicites (processus de Markov en temps continu).*

Qian et Goldstein obtiennent les processus de substitution représentés en figure 15.1 : il n'y a plus d'états puits comme c'était le cas dans [Mitchison et Durbin, 1995], mais les échangeabilités entre MD et MI ainsi qu'entre DM et DI sont nulles. Ceci semble signifier que le jeu de données utilisé n'a jamais produit d'alignements MD↔MI ni d'alignements DM↔DI, ce qui est surprenant mais illustre peut-être aussi la pauvreté du jeu de données en question (300 séquences réparties en 10 alignements dont les auteurs ne précisent pas la longueur).

15.2 Jeu de données utilisé

Pour l'apprentissage des substitutions entre transitions, nous choisissons d'utiliser un jeu de données de plus grande taille que les jeux de test SABmark et Treefam présentés plus haut. Le jeu retenu devait contenir des alignements se composant de séquences assez distantes les unes par rapport aux autres, et couvrir autant que possible tout le spectre des structures protéiques connues. C'est vers la base SCOP [Murzin *et al.*, 1995] que nous nous sommes donc tournés.

À l'automne 2011, la version courante de la base de données SCOP est la version 1.75. C'est celle que nous avons utilisée (http://scop.mrc-lmb.cam.ac.uk/scop/). SCOP est

l'acronyme de *Structural Classification of Proteins*, « classification structurelle des protéines ». Comme son nom l'indique, la base SCOP s'appuie sur des données en provenance de la base PDB ([Berman *et al.*, 2000], http://www.pdb.org/), laquelle répertorie les structures de protéines connues à ce jour (on dit aussi « élucidées ») par des techniques telles que la cristallographie par rayons X, la résonance magnétique (RMN) ou encore la microscopie électronique. Alors que la *Protein DataBank* a vocation à répertorier toutes les structures tridimensionnelles connues, SCOP est conçue pour cataloguer de façon hiérarchique les protéines partageant entre elles tout ou partie de ces structures. La hiérarchie employée dans SCOP est la suivante, donnée ci-dessous dans l'ordre allant de la catégorie la plus spécialisée à la catégorie la plus générique :

Domaines Un domaine est une sous-unité fonctionnelle au sein d'une séquence protéique. La plupart des « petites » protéines n'ont qu'un seul domaine, mais certaines protéines en possèdent plusieurs. Il faut noter que même si on emploie très souvent ici le terme *protéine*, ce sont en réalité des *domaines protéiques* que l'on manipule et qui constituent les objets (séquences d'apprentissage et cibles de détection) de notre étude. 110.800 domaines sont répertoriés dans SCOP 1.75.

Familles Les protéines sont regroupées en familles de telle façon qu'au sein d'une même famille toutes partagent un fort taux d'identité de séquence ($\geq 30\%$) et/ou une similarité fonctionnelle claire, avec toujours un certain seuil d'identité de séquence. Nous renvoyons à [Murzin *et al.*, 1995] pour ce qui concerne les détails, mais disons que l'appartenance à une même famille est un gage certain d'une origine évolutive commune. On dénombre 3902 familles dans SCOP 1.75.

Superfamilles Les familles sont regroupées en superfamilles lorsque les séquences les composant présentent des caractéristiques fonctionnelles proches avec un taux d'identité de séquence faible mais significatif. L'appartenance à une même superfamille (on en compte 1962 dans SCOP 1.75) est le signe d'une parenté phylogénétique probable à très probable.

Repliements Un repliement, ou *fold* en anglais, regroupe des superfamilles au sein desquelles les séquences présentent plus ou moins strictement le même arrangement de structures secondaires (hélices, feuillets et boucles). SCOP 1.75 se décompose en 1195 repliements.

Classes Enfin, les classes se trouvent au sommet de la hiérarchie SCOP : elles ne sont qu'au nombre de sept et constituent un moyen de ranger les repliements en fonction du nombre de domaines, de la caractérisation des structures secondaires (e.g. les protéines sont-elles constituées seulement par des hélices alpha, seulement par des feuillets beta ou bien encore par un mélange d'hélices et de feuillets ?) ou encore des caractéristiques fonctionnelles des protéines (protéines membranaires).

Construite à partir de SCOP, la base de données ASTRAL ([Brenner *et al.*, 2000; Chandonia *et al.*, 2004], http://astral.berkeley.edu) fournit les séquences associées aux domaines

issus de SCOP, mais surtout propose des jeux de données *filtrés* par rapport à ceux de la base SCOP. Il s'agit de fournir à la communauté des jeux de données dans lesquels les séquences ne partagent pas plus d'un certain seuil d'identité de séquence. Par exemple, le jeu de données « ASTRAL/SCOP PDB40 » contient des domaines protéiques tous issus de SCOP et formant un ensemble dans lequel pour toute paire (s_1, s_2) de séquences, le taux d'identité entre résidus alignés est inférieur ou égal à 40%. Cette caractéristique est précieuse pour qui souhaite avoir des jeux de données d'apprentissage relativement « difficiles », composés de séquences assez distantes les unes des autres.

C'est le jeu de données ASTAL/SCOP 1.75 PDB40 [1] que nous avons utilisé pour l'apprentissage des modèles de substitution pour les transitions dans les HMM. Il se compose de 10.569 séquences réparties dans une hiérarchie de 1961 superfamilles et 3898 familles, et couvrant tout le spectre des sept classes de SCOP.

Dans une approche similaire à celle employée par [Qian et Goldstein, 2003], nous construisons chacune des familles d'apprentissage en réunissant les séquences appartenant à $(N-1)$ des N familles d'une superfamille SCOP, N variant bien sûr d'une superfamille à l'autre. Nous ignorons les superfamilles composées d'une seule famille et les ensembles d'apprentissage constitués de moins de 10 séquences. Cette stratégie nous laisse avec 1462 ensembles de séquences d'apprentissage, tirés de 180 superfamilles différentes. Les alignements protéiques correspondants sont calculés par le logiciel MAFFT dans sa version 6.857 [Katoh *et al.*, 2002].

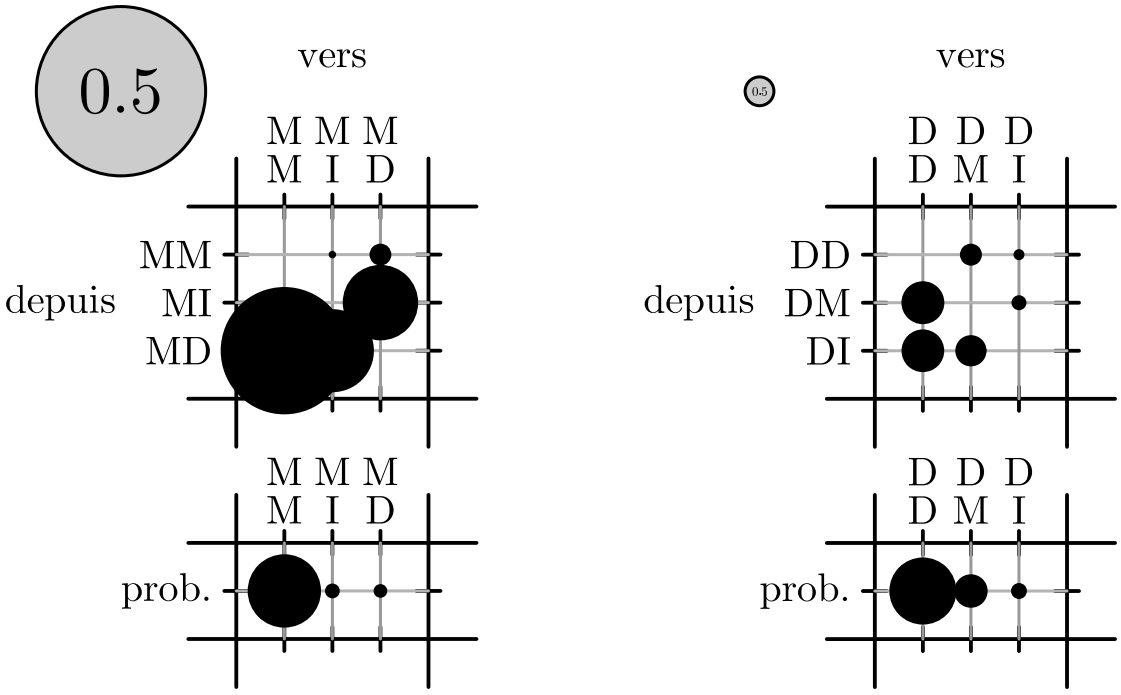
15.3 Processus de substitution appris sur notre jeu de données

Nous choisissons d'apprendre les processus de substitution affectant les transitions à partir du jeu de données d'alignements distants issu de ASTRAL/SCOP PDB40 et exposé ci-dessus. Nous avons donc 1462 couples constitués chacun d'un alignement multiple et d'un arbre estimé par maximum de vraisemblance sur les séquences protéiques (PhyML, loi Gamma à 4 classes de vitesse). Une étape importante pour la suite de l'apprentissage étant le choix des colonnes à sélectionner comme colonnes « match », nous choisissons l'approche la plus standard qui est celle que *semblent* avoir utilisé [Qian et Goldstein, 2003] (les auteurs ne le précisent pas mais évoquent cette seule heuristique dans une autre partie de leur papier) : sans pondération des séquences, sont retenues les colonnes couvertes par 50% des séquences au moins (par « couverture » nous entendons la présence d'un acide aminé à l'intersection colonne × séquence). L'apprentissage des processus se fait

1. http://astral.berkeley.edu/seq.cgi?get=scopdom-seqres-gd-sel-gs-bib;ver=1.75;item=seqs;cut=40

par un algorithme EM (*Expectation-Maximisation*) visant à maximiser les vraisemblances des arbres phylogénétiques, implémenté dans le logiciel XRATE de Klosterman, Holmes et collègues [Klosterman *et al.*, 2006]. La condition de réversibilité temporelle des processus est imposée et maintenue tout au long de la phase d'apprentissage.

Le processus d'apprentissage de deux CTMC (chaînes de Markov en temps continu) distinctes pour les transitions quittant les états Match et celles quittant les états de Délétion aboutit, à partir de notre jeu de données, aux générateurs Q_M et Q_D présentés [2] en figure 15.2. Les représentations graphiques des processus correspondants obtenus par Qian et Goldstein sont données pour comparaison en figure 15.3.

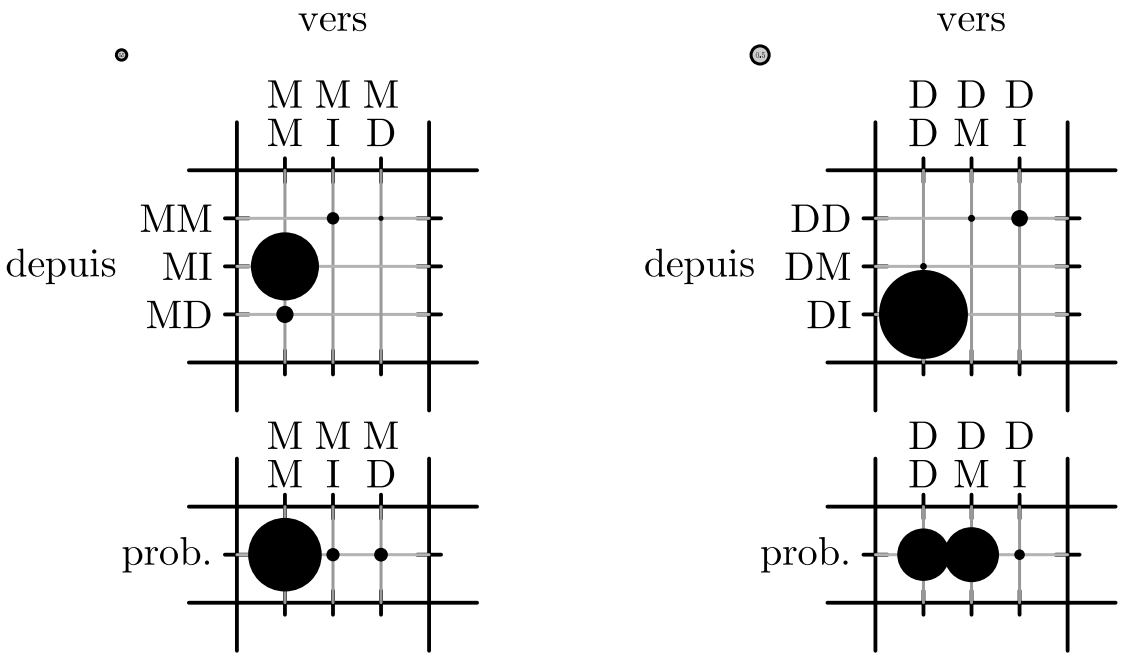


(a) Transitions quittant les états Match

(b) Transitions quittant les états de Délétion

Figure 15.2. *Processus appris sur le jeu de données ASTRAL/SCOP. On représente les taux instantanés de substitution ainsi que les probabilités d'équilibre. Le cercle en haut à gauche de chaque figure donne l'échelle, il représente un taux instantané égal à 0,5.*

2. Les descriptions graphiques de processus de substitution apparaissant dans ce qui suit sont obtenues par l'intermédiaire du script Perl `visualizeRates.pl` modifié par nos soins. Le programme fait partie de la bibliothèque logicielle DART maintenue par Ian Holmes (http://biowiki.org/DART). On en doit la paternité à Nick Goldman et Robert Bradley, ainsi qu'à Ian Holmes lui-même.



(a) Transitions quittant les états Match

(b) Transitions quittant les états de Délétion

Figure 15.3. *Processus issus de [Qian et Goldstein, 2003]. On représente les taux instantanés de substitution ainsi que les probabilités d'équilibre. Le cercle en haut à gauche de chaque figure donne l'échelle, il représente un taux instantané égal à 0,5.*

Une comparaison rapide des processus appris sur notre jeu de données avec ceux publiés par Qian et Goldstein fait resortir les points suivants :

1. en ce qui concerne les fréquences d'équilibre, elles sont similaires pour les transitions quittant les états Match. En revanche, pour [Qian et Goldstein, 2003] les transitions DD et DM sont quasi à égalité, dénotant des délétions courtes par rapport à l'architecture consensuelle (colonnes match). Par contraste, dans notre jeu de données ASTRAL/SCOP on a $\pi_{DD} \gg \pi_{DM}$, signe de délétions plus longues,
2. en ce qui concerne les substitutions, on observe que les substitutions MD↔MI et DM↔DI, absentes chez [Qian et Goldstein, 2003], sont loin d'être négligeables sur la base de notre jeu de données,
3. Qian et Goldstein ont $q_{MM,MD} < q_{MM,MI}$ alors que c'est l'inverse dans les processus que nous avons appris. Le faible taux $q_{MM,MD}$ chez Qian et Goldstein dénote des colonnes match bien conservées dans des alignements où les variations sont introduites presque exclusivement par des insertions par rapport à l'architecture consensus. Dans notre jeu de données en revanche, les variations par rapport au consensus

sont en plus grand nombre des délétions que des insertions, signe d'une architecture consensus qui n'est pas si consensuelle que cela.

Lorsqu'on apprend un unique processus 6×6 sur les transitions quittant les états Match *et* les états de Délétion, à partir de notre jeu de données ASTRAL/SCOP on obtient le processus représenté en figure 15.4.

On peut clairement voir que les substitutions par construction impossibles chez [Qian et Goldstein, 2003] sont loin d'être négligeables dans le processus 6×6 que nous avons appris. Nous classons dans le tableau ci-dessous les 15 échangeabilités obtenues (on rappelle $\text{ech}_{i\to j} = \frac{q_{i,j}}{\pi_j}$) :

éch. majeures	éch.importantes	éch. moyennes	éch. très faible
DI↔MD (81,48)	MD↔DM (43,34)	MM↔DI (8,66)	MM↔DD (0,10)
DI↔DM (79,96)	MI↔MD (38,33)	DD↔DI (4,79)	
DI↔MI (70,55)	MI↔DM (37,63)	MM↔MD (4,43)	
		MM↔DM (4,33)	
		MM↔MI (3,75)	
		DD↔MD (2,78)	
		DD↔DM (2,77)	
		MI↔DD (2,45)	

La première colonne du tableau nous indique qu'au-delà de ce qui est attendu en raison de sa faible probabilité d'équilibre, la transition DI est extrêmement instable. La dernière colonne nous renseigne sur le fait que les délétions par rapport au consensus sont en général plutôt très courtes (un acide aminé). Enfin, en troisième colonne, on constate que l'échangeabilité entre MM et MD est proche de celle constatée entre MM et DM, et de même entre d'un côté DD↔MD et de l'autre DD↔DM. La condition de réversibilité spatiale énoncée en 9.3.3 exprimait deux égalités sur les taux instantanés de transition : $q_{MM,MD} = q_{MM,DM}$ et $q_{DD,MD} = q_{DD,DM}$. On observe dans le processus appris les taux suivants :

$q_{MM,MD}$	$q_{MM,DM}$	$q_{DD,MD}$	$q_{DD,DM}$
0,0490	0,0499	0,0308	0,0319

On peut donc dire que les conditions de réversibilité spatiale énoncées plus haut sont relativement bien vérifiées par le processus appris, sans qu'on ait fixé de telles contraintes lors de la phase d'apprentissage.

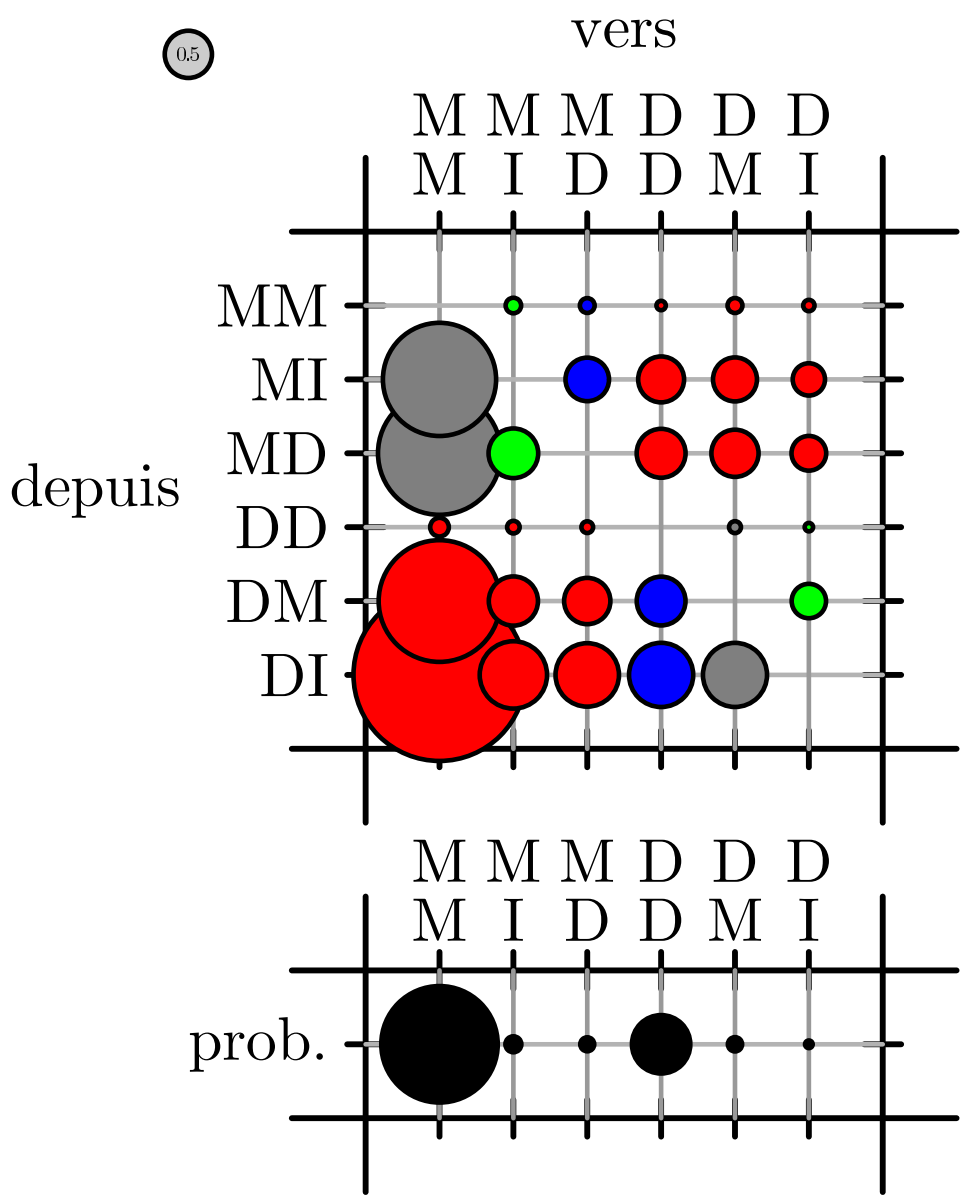


Figure 15.4. *Processus de substitution entre transitions (M→) et (D→), appris sur le jeu de données ASTRAL/SCOP. On représente en rouge les substitutions faisant évoluer l'état de départ de la transition, c'est-à-dire précisément celles qu'ignorent les processus construits séparément sur les états Match et sur les états de Délétion.*

CHAPITRE

16 Phylogénisation de l'architecture, des émissions sur les états Match et des transitions quittant les états Match et Délétion

Nous introduisons en plus de la phylogénisation des architectures de HMM et des émissions sur les états Match, la phylogénisation des transitions quittant les états Match et Délétion. Cette phylogénisation se fait conformément à la méthode exposée au chapitre 9, en utilisant les processus appris au chapitre précédent. Avant de procéder à la phylogénisation des transitions, nous vérifions la présence d'un signal phylogénétique dans les alignements de transitions.

Sommaire

16.1 Vérification de la présence d'un signal phylogénétique dans les alignements de transitions

Nous désirons tester la présence d'un signal phylogénétique, comme nous l'avions fait en section 13.1 pour ce qui concernait les alignements de caractères binaires Gap/Nongap. Ici nous voulons faire le moins possible d'hypothèses quant au processus de substitution à l'œuvre dans les arbres dont les feuilles portent des caractères représentant les transitions. Le type de test nous permettant de nous affranchir de nombre d'hypothèses (et en particulier de ne pas supposer connu le processus de substitution) est un test de randomisation avec mesures de parcimonie : une phylogénie avec des caractères aux feuilles

donne un score de parcimonie qui ne dépend ni d'un modèle de substitution, ni même des longueurs des branches : c'est le nombre minimal d'événements mutationnels permettant d'expliquer l'arbre observé (cf. section 4.2).

Si les caractères aux feuilles sont posés aléatoirement, sans tenir aucun compte des relations de proximité induites par la phylogénie, alors il sera aisé de trouver une permutation des caractères aux feuilles donnant un score de parcimonie *strictement moins élevé* que le score originel. Nous nous servons de cette propriété pour définir notre test de randomisation : si sur 100 permutations aléatoires des caractères aux feuilles, 5 ou plus donnent un score de parcimonie moins élevé que le score correspondant à la configuration observée des caractères aux feuilles, alors on pourra dire que la phylogénie n'explique pas de manière satisfaisante les caractères aux feuilles, et on dira que le couple { arbre, alignement } échoue au test de randomisation.

Comme en section 13.1, on conserve un seuil de 5% pour séparer l'échec du succès au test, même lorsque la famille testée comporte trop peu de feuilles pour permettre 100 permutations. Un succès correspond alors à une situation dans laquelle le nombre constaté de scores de parcimonie inférieurs au score originel, rapporté au nombre total de permutations possible, est strictement inférieur à 5%.

Il faut remarquer ici que les alignements de transitions *dépendent* de l'étiquetage de l'alignement d'apprentissage en colonnes « match » et « non match ». Pour une famille donnée, il y a donc autant d'alignements de transitions que la phylogénie support comprend de nœuds. Pour chaque famille, on teste la totalité des alignements de transitions issus de la phylogénisation des architectures sur ces différents nœuds.

Enfin, on se place dans l'hypothèse conservatrice (par rapport à la littérature existante) d'alignements distincts, d'une part pour les transitions au départ d'un état Match, d'autre part pour celles au départ d'un état de Délétion.

Sur les 409 familles du banc de test TreeFam, on détermine ainsi 15.082 architectures en tout. Parmi les alignements de transitions M→ issus de ce jeu de données, 14.954 passent le test de randomisation (soit 99,1%). Ce chiffre descend à 13.170 (87,3%) pour ce qui est des alignements de transitions D→.

Sur les 420 familles du banc de test SABmark, on détermine un total de 5686 architectures, sur lesquelles 4448 arbres de transitions M→ passent le test (soit 78,2%), tandis que ce chiffre est de 4499 (79,1%) pour les arbres de transitions D→.

Il faut prendre ces tests pour ce qu'ils sont : des tests simples et rapides à implémenter ne prétendant pas à l'irréfutabilité. Le seuil choisi (5%) est relativement peu contraignant,

on aurait par exemple pu choisir de n'autoriser aucune inversion de score de parcimonie pour valider un arbre de transitions. Néanmoins, ce test est suffisant pour démontrer une corrélation non nulle entre la topologie et les transitions observées aux feuilles, ce qui est une hypothèse réellement *a minima* pour qui veut avancer et procéder à de l'inférence phylogénétique sur de tels caractères.

16.2 Résultats sur le banc de test SABmark

16.2.1 Vraisemblance des cibles positives

Comme précédemment, les HMM phylogénisés augmentent largement, en moyenne, la vraisemblance des protéines cibles. La différence de gain de vraisemblance par rapport aux expériences précédentes (phylogénisation de l'architecture seulement ou bien de l'architecture et des émissions sur les états Match) n'est pas suffisamment grande pour qu'on puisse l'observer aisément à partir d'une représentation en nuages de points comme précédemment. On représente en revanche les points de log-vraisemblance gagnés par la phylogénisation des transitions par rapport à la situation sans phylogénisation des transitions (mais avec une architecture et des émissions sur les états Match calculés via le processus de reconstruction ancestrale que l'on a exposé). On représente les résultats obtenus en figure 16.1 pour le cas d'une phylogénisation via un seul processus markovien unifié, et en figure 16.2 pour le cas avec deux processus markoviens distincts.

Dans les deux cas, on observe un nuage de points dont le centre de gravité se trouve au-dessus de l'axe des abscisses, c'est-à-dire que globalement les résultats de la phylogénisation sont positifs, et les cibles recoivent des vraisemblances accrues grâce à la phylogénisation des transitions. Ce comportement est notablement plus marqué pour la phylogénisation passant par deux processus distincts (figure 16.2). En effet, si l'on met en regard les résumés statistiques des deux distributions d'écarts de score, on constate :

	minimum	1er quartile	médiane	moyenne	3^{e} quartile	maximum
1 processus	-137,7	-0,6	1,4	3,8	7,1	100,9
2 processus	-57,6	-0,6	1,6	4,8	8,5	112,3

La modélisation de l'évolution des transitions par deux processus distincts semble donc mieux convenir à ce jeu de données, en donnant aux cibles à prédire des scores plus élevés.

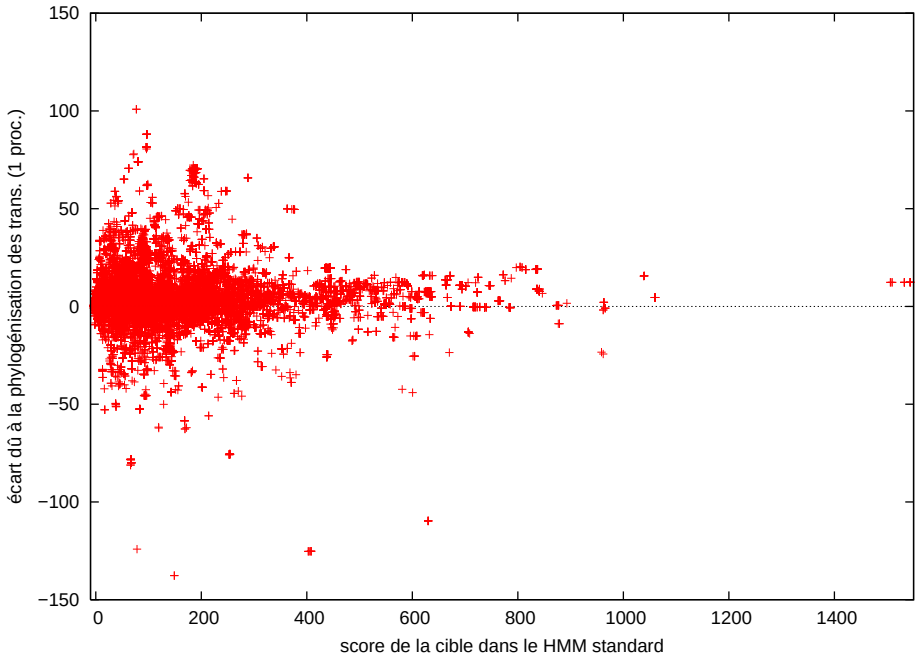


Figure 16.1. *Écarts de log-vraisemblance constatés sur le banc SABmark entre les scores obtenus par les cibles dans les HMM où l'on a phylogénisé à la fois les architectures, les émissions sur les états Match et les transitions, avec un seul processus de substitution pour ces dernières, par rapport aux scores obtenus sans phylogénisation des transitions*

16.2.2 Détection

Les figures 16.3 et 16.4 donnent les courbes ROC dans le cas où les résultats du HMM standard ne sont pas pris en ligne de compte dans l'opération de maximum aboutissant au score d'une séquence dans le phylo-HMM. La phylogénisation de l'architecture seulement se révèle être alors la meilleure stratégie, et la phylogénisation des transitions dégrade même les résultats par rapport aux HMM profils dans lesquels les émissions sur les états Match sont phylogénisées en plus de l'architecture.

Les résultats obtenus lorsqu'on intègre les résultats du HMM standard dans la procédure de recherche du meilleur score d'une séquence sont donnés en figures 16.5 et 16.6. Ils sont en contraste net avec ceux obtenus sans considération des scores obtenus par les séquences dans les HMM standard. Cette fois-ci, la phylogénisation des transitions apporte un gain extrêmement modeste, mais un gain tout de même. Une légère amélioration est obtenue lorsqu'on phylogénise les transitions via un processus markovien unique plutôt que par deux processus distincts, l'un pour les transitions quittant les états Match et l'autre pour les transitions quittant les états de Délétion. Cette modélisation par un processus

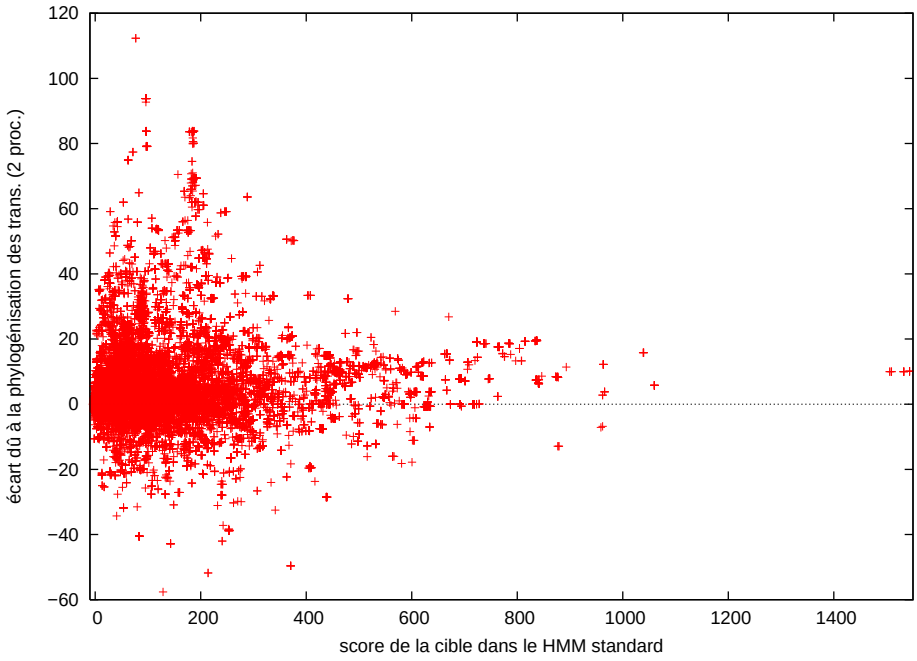


Figure 16.2. *Écarts de log-vraisemblance constatés sur le banc SABmark entre les scores obtenus par les cibles dans les HMM où l'on a phylogénisé à la fois les architectures, les émissions sur les états Match et les transitions, avec deux processus de substitution distincts pour ces dernières (M→ et D→), par rapport aux scores obtenus sans phylogénisation des transitions*

unique semble donc justifiée, même si le gain en détection n'est pas robuste lorsqu'on descend dans la liste des scores et qu'on agrège de plus en plus de résultats négatifs.

16.3 Sur le banc de test TreeFam

16.3.1 Vraisemblance des cibles

Comme pour le banc SABmark, on présente les résultats en termes d'accroissement de la log-vraisemblance par rapport à la situation dans laquelle on phylogénise l'architecture et les émissions sur les états Match, mais pas les transitions. On peut consulter les résultats en figure 16.7 et en figure 16.8.

Là encore, il semble y avoir une relation de linéarité entre le gain apporté par la phylogénisation des transitions et le score déjà obtenu par la séquence dans le HMM standard : en moyenne, plus la séquence cible est nettement reconnue par le HMM standard, plus

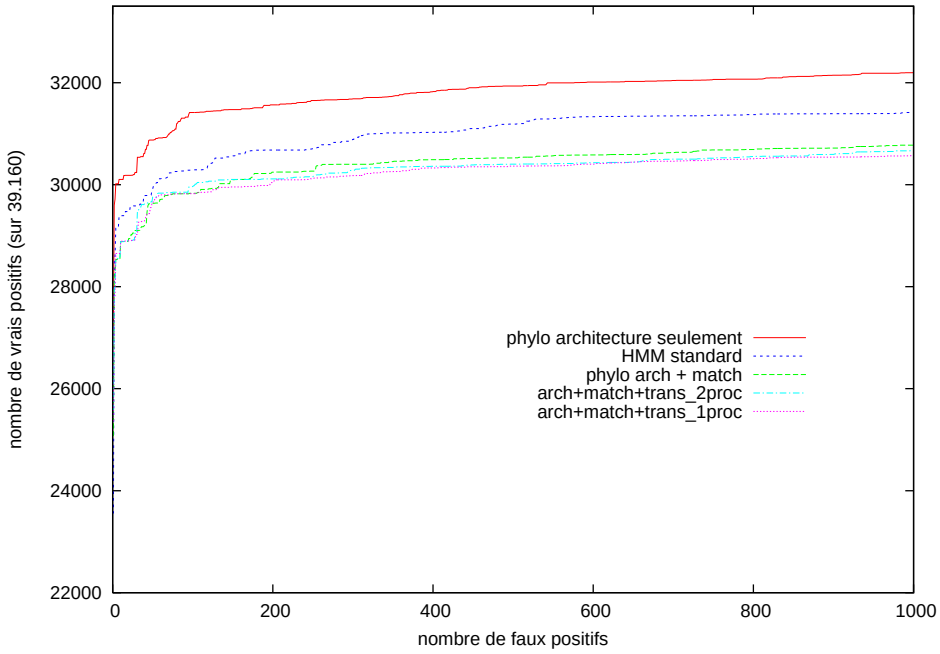


Figure 16.3. *Courbes ROC sur le banc de test SABmark, phylogénisation de l'architecture, des émissions sur les états Match et des transitions M→ et D→. Les scores donnés par les HMM standard ne sont pas pris en compte dans la détermination du score dans le modèle phylogénisé. On constate que dans ce cas, la phylogénisation d'un nombre croissant de paramètres (contenu des états Match et transitions) dégrade les performances par rapport à la phylogénisation de l'architecture seule.*

la phylogénisation est précise et apporte un gain de vraisemblance. La résumé statistique des distributions des écarts de score donne :

	minimum	1er quartile	médiane	moyenne	3e quartile	maximum
1 processus	-149,4	4,4	10,6	14,5	20,3	262,4
2 processus	-249,9	4,3	12,6	16,1	24,8	261,4

Encore une fois, la modélisation de l'évolution des transitions par deux processus distincts semble plus appropriée, en donnant aux cibles à prédire des scores légèrement plus élevés. Ce n'est toutefois pas très sensible, et la phylogénisation via deux processus distincts commet un peu plus d'erreurs nettes : dans la queue gauche des distributions ci-dessus, le premier décile se situe à $-4,4$ pour le cas « 2 processus » contre $-0,7$ pour le cas « 1 processus ».

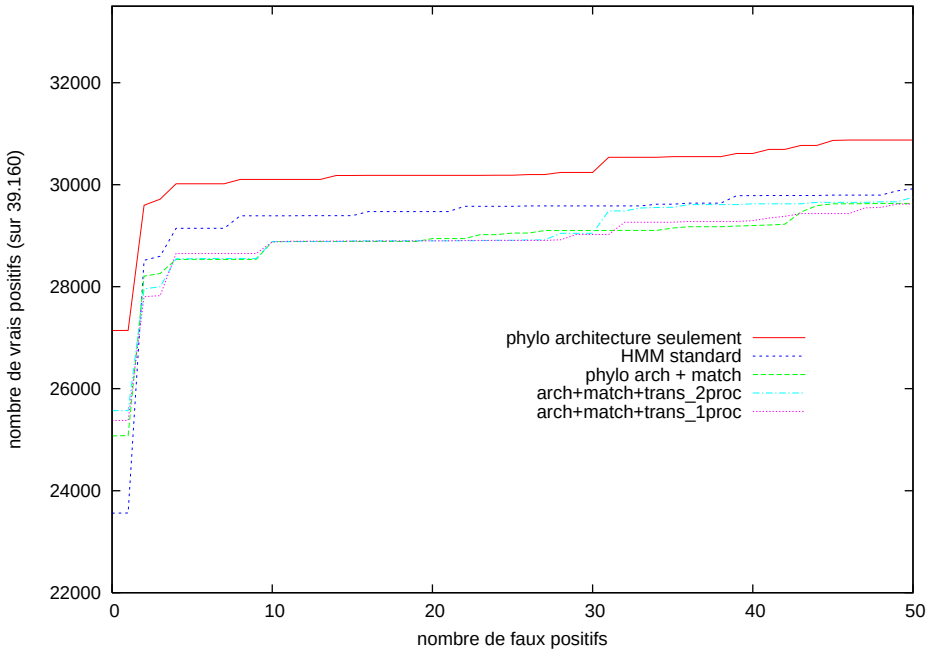


Figure 16.4. *Courbes ROC sur le banc de test SABmark, phylogénisation de l'architecture, des émissions sur les états Match et des transitions M→ et D→ (zoom). Les scores donnés par les HMM standard ne sont pas pris en compte dans la détermination du score dans le modèle phylogénisé. La phylogénisation des transitions apporte une amélioration dans le nombre de cibles positives détectées sans faux positif, par rapport à la situation où l'on phylogénise seulement les architectures et le contenu des états Match.*

16.3.2 Résultats en termes de détection

Ces résultats sont présentés en figures 16.9 et 16.10. On observe grosso modo le même comportement que sur le banc de test SABmark, avec cependant un gain plus net pour la phylogénisation à partir d'un processus markovien unique pour modéliser les transitions (figure 16.10). Il est à noter que la phylogénisation des transitions n'apporte globalement qu'un gain très faible par rapport au gain déjà obtenu par la phylogénisation des architectures et des contenus en émission des états Match (figure 16.9).

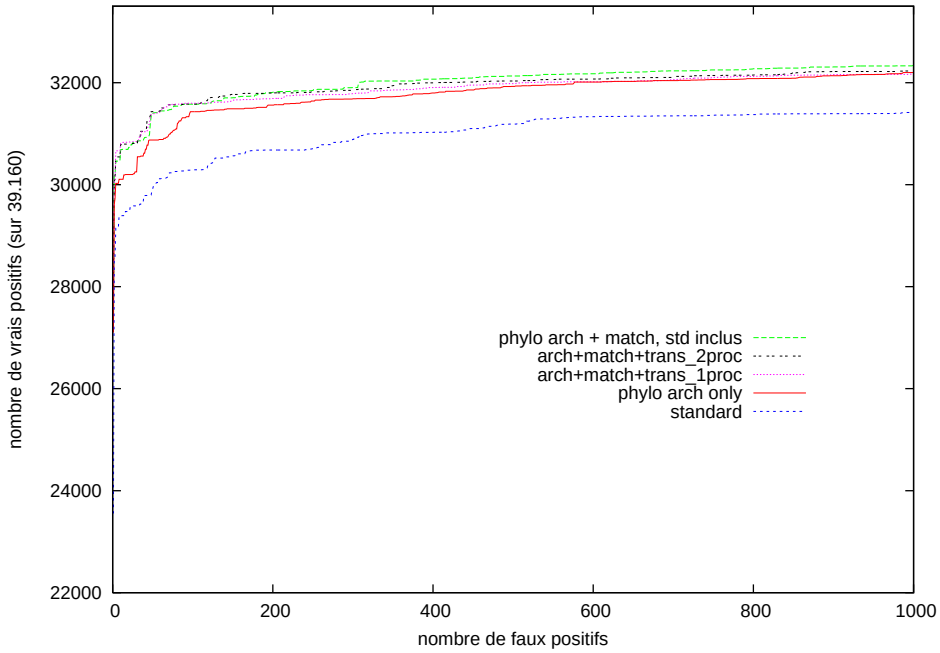


Figure 16.5. *Courbes ROC sur le banc de test SABmark, phylogénisation de l'architecture, des émissions sur les états Match et des transitions M→ et D→. Les résultats du HMM standard sont pris en compte dans le calcul du meilleur score d'une séquence. Même si la phylogénisation des transitions ne dégrade pas les performances en détection, on a bien l'impression ici qu'elle n'apporte aucun gain...*

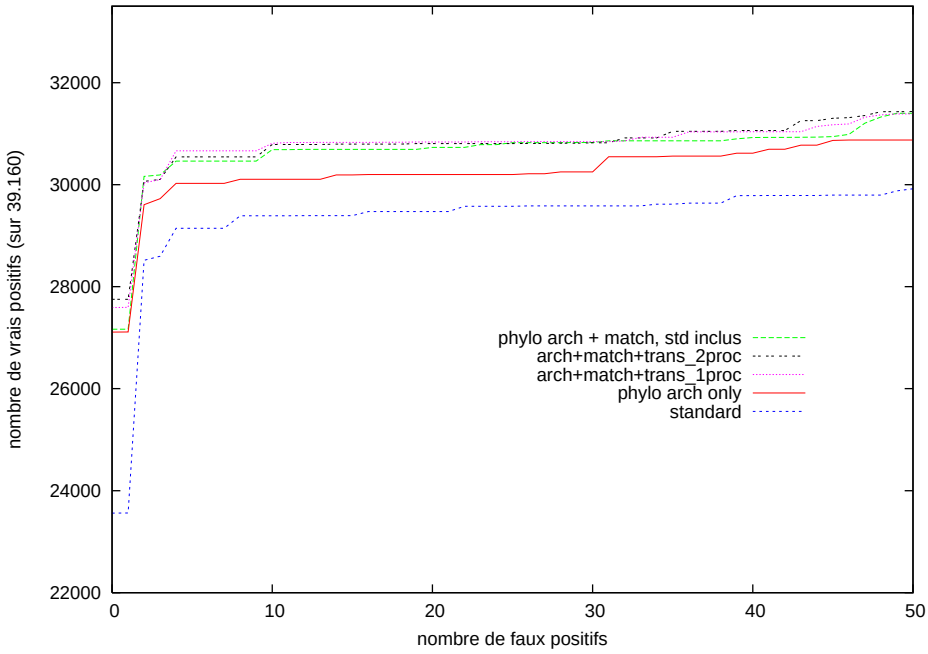


Figure 16.6. *Courbes ROC sur le banc de test SABmark, phylogénisation de l'architecture, des émissions sur les états Match et des transitions M→ et D→ (zoom). Les résultats du HMM standard sont pris en compte dans le calcul du meilleur score d'une séquence. On voit ici que la phylogénisation des transitions apporte un léger gain lorsqu'on considère une région correspondant à un faible taux de faux positifs.*

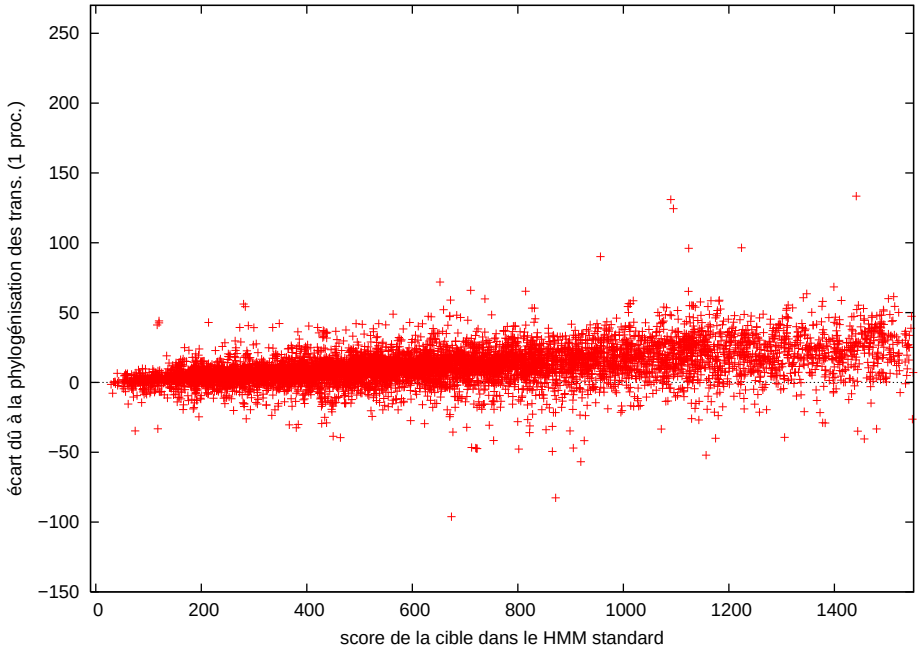


Figure 16.7. *Écarts de log-vraisemblance constatés sur le banc TreeFam entre les scores obtenus par les cibles dans les HMM où l'on a phylogénisé à la fois les architectures, les émissions sur les états Match et les transitions, avec un seul processus de substitution pour ces dernières, par rapport aux scores obtenus sans phylogénisation des transitions*

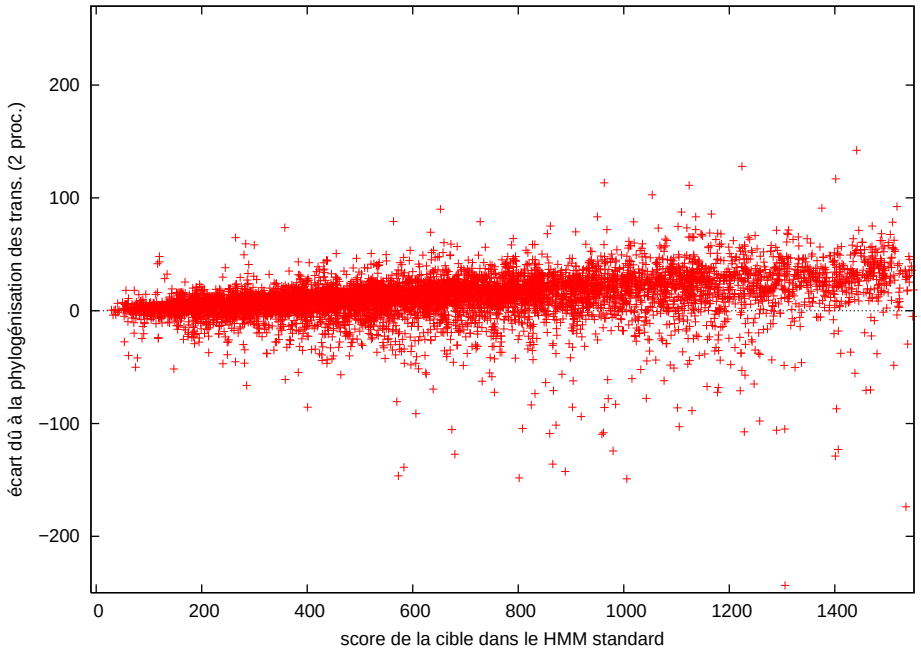


Figure 16.8. *Écarts de log-vraisemblance constatés sur le banc TreeFam entre les scores obtenus par les cibles dans les HMM où l'on a phylogénisé à la fois les architectures, les émissions sur les états Match et les transitions, avec deux processus de substitution distincts pour ces dernières (M→ et D→), par rapport aux scores obtenus sans phylogénisation des transitions*

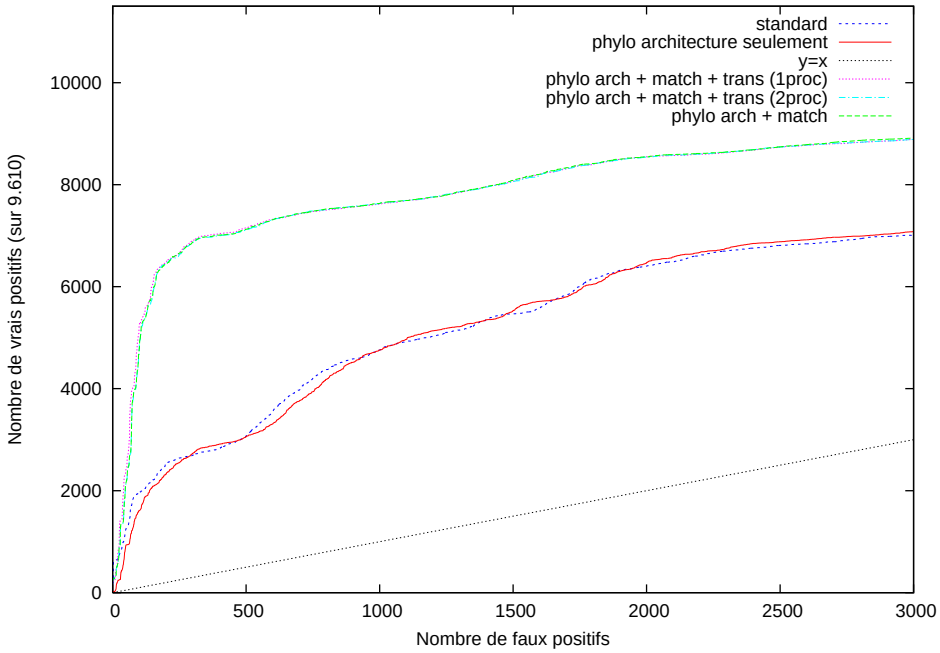


Figure 16.9. *Courbes ROC sur le banc de test TreeFam, phylogénisation de l'architecture, des émissions sur les états Match et des transitions M→ et D→. Les performances en détection apportées par la phylogénisation des transitions (en plus de l'architecture et des émissions sur les états Match) sont strictement comparables à celles sans phylogénisation des transitions.*

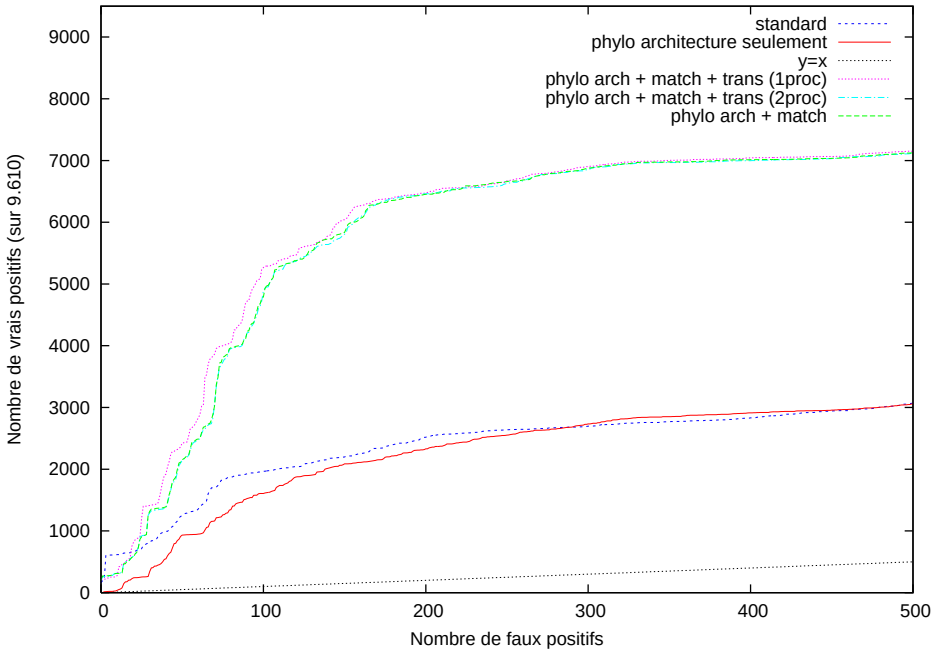


Figure 16.10. *Courbes ROC sur le banc de test SABmark, phylogénisation de l'architecture, des émissions sur les états Match et des transitions M→ et D→ (zoom). On relève ici un léger bénéfice apporté par la phylogénisation des transitions sur la base d'un processus de substitution unifié pour celles-ci. Ce bénéfice disparaît si l'on choisit d'utiliser deux processus distincts selon que les transitions quittent un état Match ou un état de Délétion.*

CHAPITRE

17 Phylogénisation du contenu en émission des états d'Insertion

Nous présentons ici les résultats obtenus par la phylogénisation du contenu des états d'Insertion, qui concerne les probabilités d'émission d'acides aminés sur les états en question.

La phylogénisation se fait conformément à ce qui a été présenté au chapitre 11 : pour un état d'Insertion donné, les vraisemblances partielles aux feuilles sont proportionnelles aux fréquences d'apparition des caractères dans la séquence correspondant à la feuille en question et dans la zone d'insertion correspondante. Le modèle substitutionnel que nous faisons agir le long des branches de la phylogénie est le modèle réversible constitué à partir des échangeabilités entre acides aminés données par la matrice LG [Le et Gascuel, 2008] et des fréquences d'équilibre correspondant au modèle émittif des états d'insertion dans HMMER. Ces fréquences sont données en figure 17.1.

Ala (A)	6,81	Gln(Q)	4,15	Leu (L)	6,76	Ser (S)	9,27
Arg (R)	5,51	Glu (E)	6,51	Lys (K)	6,87	Thr (T)	6,23
Asn(N)	5,48	Gly (G)	9,02	Met(M)	1,43	Trp (W)	1,02
Asp(D)	6,23	His (H)	2,41	Phe (F)	3,13	Tyr (Y)	2,69
Cys (C)	1,20	Ile (I)	3,71	Pro (P)	6,47	Val (V)	5,05

Figure 17.1. *Distribution a priori pour les émissions sur les états d'Insertion dans la suite logicielle* HMMER.

Le processus d'inférence phylogénétique du contenu des états d'Insertion aboutit bien à des phylo-HMM de nœuds qui présentent des distributions d'acides aminés différentes d'un état d'Insertion à l'autre, là où la très forte pondération accordée à la distribution a

priori au détriment des observations donne dans les HMM standard de HMMER des états d'Insertion tous identiques (voir section 3.3.4).

Cependant, de façon un peu surprenante, alors que les HMM construits en ajoutant à la phylogénisation des architectures et du contenu des états Match, celle du contenu des états d'Insertion, sont sensiblement différents des phylo-HMM sans phylogénisation des états d'Insertion, les deux donnent *exactement* les mêmes résultats, que ce soit en termes de vraisemblance des cibles à prédire ou de performances en détection, ce pour les deux jeux de test (SABmark et TreeFam). Pour toutes les familles, la liste des scores de séquences produite par un HMM phylogénisé { architecture + Match + Insert } est exactement la même que celle que produit son alter ego, c'est-à-dire le HMM phylogénisé { architecture + Match } estimé sur le même nœud de la phylogénie.

On peut avancer une explication à ce phénomène : si l'étape de phylogénisation de l'architecture est menée de façon pertinente, alors par construction les séquences cibles que l'on souhaite détecter avec un HMM phylogénisé *n'empruntent dans leur chemin de score optimal qu'une succession d'états Match*, lesquels correspondent justement aux colonnes retenues par l'étape de phylogénisation de l'architecture. Ainsi, une phylogénisation efficace des architectures des HMM, c'est-à-dire un choix pertinent de colonnes d'intérêt effectué pour un voisinage phylogénétique donné, rendrait quasiment inutiles les états d'Insertion.

CHAPITRE

18 Sur les longueurs des insertions

18.1 Mesure de la fiabilité du signal phylogénétique dans les alignements de longueurs d'insertion

Nous nous intéressons ici à quantifier le signal phylogénétique présent dans des alignements de caractères discrets mais quantitatifs, à savoir les longueurs des insertions réalisées par les séquences entre deux colonnes Match consécutives dans un HMM donné. La notion de *signal phylogénétique* mérite au préalable quelques explications.

Quel que soit le modèle évolutif adopté, un arbre phylogénétique correctement construit sur un alignement de caractères devrait être tel que la vraisemblance de celui-ci soit supérieure à celles correspondant au même arbre affecté par une permutation mélangeant les données à ses feuilles. Considérons un site donné X et un arbre phylogénétique $\mathcal{T}$ dont on ordonne les feuilles en affectant à chacune un indice allant de 1 à n. Les caractères observés X_i correspondent aux différents taxa aux feuilles. On appelle « signal phylogénétique » le degré d'accord entre les observations faites aux feuilles et la structure *attendue* des corrélations entre les caractères aux feuilles (on attend par exemple que deux feuilles proches voisines portent des caractères semblables).

18.1.1 La statistique K pour des tests de corrélation sur caractères quantitatifs

Dans [Blomberg *et al.*, 2003], les auteurs ont introduit une méthodologie de test du signal phylogénétique pour les analyses comparatives sur des caractères quantitatifs (c'est-à-dire représentant des valeurs ordonnées sur la droite réelle). Leurs mesures s'appuient

sur un modèle d'évolution brownien, dont nous rappelons ci-dessous les principales caractéristiques (cf chapitre 10.4.2).

Le mouvement brownien et la matrice induite de variance-covariance

Le mouvement brownien est celui d'un processus stochastique de Wiener, noté W. Si $W(t=0) = x_0$, alors la densité de probabilité correspondant à l'état $W(t)$ du processus à l'instant $t > 0$ est celle d'une gaussienne centrée en x_0 et de variance $\sigma^2 t$. La valeur $\sigma > 0$ est le seul paramètre du processus. Edwards et Cavalli-Sforza ont été les premiers [Edwards *et al.*, 1964] à suggérer que l'évolution des fréquences alléliques au sein d'une population pouvait être modélisée par un processus de Wiener évoluant le long des branches d'un arbre. Plus tard, de nombreux auteurs (notablement Joseph Felsenstein, 1985) ont popularisé cette conception et l'ont appliquée à divers caractères quantitatifs.

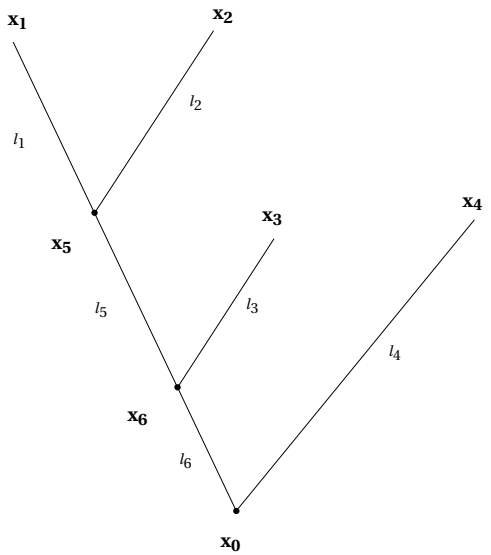


Figure 18.1. *Une phylogénie à quatre taxa. On représente les nœuds et les longueurs des branches.*

Considérons l'arbre phylogénétique orienté (la racine est l'espèce numérotée 0) présenté en figure 18.1. Soit un trait [1] X que l'on assimile à un processus de Wiener dont les réalisations aux nœuds de la phylogénie sont notées x_i (i étant l'indice du nœud correspondant). Les réalisations observables de ce processus de Wiener sont les x_i correspondant aux feuilles. La propriété fondamentale de la modélisation par mouvement brownien

1. On emploie souvent le mot « trait » pour parler d'un caractère quantitatif dont la valeur moyenne varie d'une espèce à l'autre. Il s'agit par exemple de la température corporelle des individus ou encore du diamètre de la boîte cranienne des mammifères, etc.

est que la variation d'un trait entre un nœud et l'un de ses fils séparé de lui par une branche de longueur l est distribuée selon une loi normale centrée en 0 et de variance égale à $\sigma^2 l$, où σ est le paramètre du processus de Wiener. En termes mathématiques, on a par exemple pour la figure 18.1 :

$$(x_1 - x_5) \sim \mathcal{N}(0, \sigma^2 l_1) \tag{18.1}$$

où $\mathcal{N}(\mu, \sigma^2)$ dénote la loi normale d'espérance μ et de variance σ^2.

Si l'on suppose connue la valeur x_0 du trait considéré à la racine de l'arbre, alors chacune des valeurs aux nœuds a accumulé une quantité aisément calculable de variance. Par exemple, en ce qui concerne la feuille n° 1, on a :

$$x_1 = (x_1 - x_5) + (x_5 - x_6) + (x_6 - x_0) + x_0 \tag{18.2}$$

La connaissance de x_0 en faisant ici une constante, les autres termes du membre droit de (18.2) sont indépendants les uns des autres (le mouvement brownien étant un processus markovien en temps continu), et chacun est distribué selon une loi normale centrée. Les variances s'ajoutent donc pour donner :

$$\mathrm{Var}(x_1) = \mathrm{Var}(x_1 - x_5) + \mathrm{Var}(x_5 - x_6) + \mathrm{Var}(x_6 - x_0) = l_1\sigma^2 + l_5\sigma^2 + l_6\sigma^2 = (l_1 + l_5 + l_6)\sigma^2 \tag{18.3}$$

La variance accumulée par un caractère sur une feuille est donc proportionnelle à la distance qui sépare celle-ci de la racine. Par exemple, x_1 est distribué selon une loi normale centrée en x_0 et de variance $(l_1 + l_5 + l_6)\sigma^2$.

On peut également vouloir s'intéresser aux covariances entre les caractères portés par deux nœuds distincts. On rappelle que par définition $\mathrm{cov}(X, Y) = E[(X - E(X))(Y - E(Y))]$. Par exemple, pour les taxa numérotés 1 et 2 et pour le trait considéré, la covariance est égale à :

$$\begin{aligned} \mathrm{cov}(x_1, x_2) &= E[(x_1 - E(x_1))(x_2 - E(x_2))] \\ &= E[(x_1 - x_0)(x_2 - x_0)] \\ &= E\left[((x_1 - x_5) + (x_5 - x_6) + (x_6 - x_0))\left((x_2 - x_5) + (x_5 - x_6) + (x_6 - x_0)\right)\right] \end{aligned} \tag{18.4}$$

Le produit de sommes de variables aléatoires qui apparaît ci-dessus se développe en une somme de produits, dont tous sauf deux sont d'espérance nulle. En effet, les variations de valeur du caractère x correspondant aux différentes branches de l'arbre suivent toutes *indépendamment les unes des autres* des lois gaussiennes centrées en 0. On a donc par exemple $E[(x_1 - x_5)(x_2 - x_5)] = E[x_1 - x_5]E[x_2 - x_5] = 0$. Seuls persistent les termes de variance :

$$\mathrm{cov}(x_1, x_2) = E\left[(x_5 - x_6)^2\right] + E\left[(x_6 - x_0)^2\right] = \mathrm{Var}(x_5 - x_6) + \mathrm{Var}(x_6 - x_0) = (l_5 + l_6)\sigma^2 \tag{18.5}$$

De façon similaire, on peut définir les covariances entre deux nœuds quelconques de la phylogénie : celles-ci sont systématiquement proportionnelles à la longueur du chemin menant de la racine jusqu'au plus récent ancêtre commun aux deux nœuds en question. Ainsi, logiquement, deux feuilles très proches entre elles tout en étant éloignées de la racine, partageront une grande partie du chemin les séparant de la racine et admettront donc une covariance élevée.

Un arbre construit sur n taxa nous permet donc de définir une matrice V de taille $n \times n$ dite de « variance-covariance », telle que $\forall i \neq j \; V_{ij} = \text{cov}(x_i, x_j)$ et $\forall i \; V_{ii} = \text{Var}(x_i)$. Cette matrice encode directement l'information contenue dans l'arbre en termes de longueur des chemins partagés. L'arbre représenté en figure 18.1 donne une matrice :

$$V = \sigma^2 \begin{bmatrix} l_1 + l_5 + l_6 & l_5 + l_6 & l_6 & 0 \\ l_5 + l_6 & l_2 + l_5 + l_6 & l_6 & 0 \\ l_6 & l_6 & l_3 + l_6 & 0 \\ 0 & 0 & 0 & l_4 \end{bmatrix} \tag{18.6}$$

En termes matriciels, si l'on note $\vec{x} = (x_1, x_2, x_3, x_4)$, alors ce vecteur $\vec{x}$ suit une distribution multivariée normale définie par la valeur du trait considéré à la racine et par la matrice de variance-covariance :

$$\vec{x} \sim \mathcal{N}(x_0 \mathbf{1}, V) \tag{18.7}$$

où $\mathbf{1}$ est le vecteur $(1, 1, 1, 1)$.

18.1.2 Estimation de la valeur du trait à la racine

La modélisation de l'évolution d'un caractère par mouvement brownien se fait en réalité à partir de deux paramètres :

1. la valeur du caractère à la racine de l'arbre, c'est-à-dire la valeur du processus à $t = 0$,
2. la vitesse globale d'évolution, qui correspond à la variance σ^2 du processus.

Le premier de ces deux paramètres est inconnu et demande donc à être estimé à partir des valeurs aux feuilles. Considérons une phylogénie minimale reliant une racine v_0 à deux feuilles v_1 et v_2, par des branches de longueurs respectives l_1 et l_2. On note x_1 la valeur observé pour le caractère d'intérêt en v_1, et x_2 l'observation en v_2. Soit x_0 la valeur (inconnue) du caractère d'intérêt à la racine de l'arbre. La modélisation par mouvement brownien implique alors que x_1 et x_2 sont toutes deux des variables aléatoires tirées selon des distributions normales centrées en x_0, la première ayant pour variance $l_1\sigma^2$ et la seconde $l_2\sigma^2$. La densité de probabilité liée aux observations x_1 et x_2 *sachant* x_0 est donc (nous l'appelons L car c'est une forme de vraisemblance) :

$$L = \frac{1}{\sigma\sqrt{2\pi l_1}} e^{-\frac{1}{2}\frac{(x_1-x_0)^2}{l_1\sigma^2}} \frac{1}{\sigma\sqrt{2\pi l_2}} e^{-\frac{1}{2}\frac{(x_2-x_0)^2}{l_2\sigma^2}} = \frac{1}{2\pi\sigma^2\sqrt{l_1 l_2}} e^{-\frac{1}{2\sigma^2}\left[\frac{(x_1-x_0)^2}{l_1} + \frac{(x_2-x_0)^2}{l_2}\right]} \tag{18.8}$$

Estimer x_0 au maximum de vraisemblance revient donc à minimiser le terme

$$Q = \frac{(x_1 - x_0)^2}{l_1} + \frac{(x_2 - x_0)^2}{l_2} \tag{18.9}$$

or :

$$\frac{dQ}{dx_0} = -2\frac{(x_1 - x_0)}{l_1} - 2\frac{(x_2 - x_0)}{l_2} \tag{18.10}$$

et en résolvant $dQ/dx_0 = 0$, on trouve :

$$\hat{x_0} = \frac{l_2}{(l_1 + l_2)}x_1 + \frac{l_1}{(l_1 + l_2)}x_2 \tag{18.11}$$

Autrement dit, l'estimateur au maximum de vraisemblance pour x_0 est une moyenne pondérée des valeurs aux feuilles, le coefficient affecté à chaque feuille étant inversement proportionnel à la distance séparant celle-ci de la racine : plus une feuille est proche de la racine, plus elle a de poids dans l'estimation de la valeur à la racine, ce qui conforte notre intuition.

Un raisonnement de proche en proche s'applique, et l'on peut calculer ainsi l'estimateur au maximum de vraisemblance de la valeur portée à la racine pour un arbre comportant un nombre quelconque de feuilles [Felsenstein, 1985; Garland Jr *et al.*, 1999]. On appellera désormais cette estimateur « moyenne phylogénétique ».

La statistique K de Blomberg, Garland et Ives

La statistique K introduite par [Blomberg *et al.*, 2003] consiste à mesurer combien la phylogénie testée constitue un modèle explicatif pour les données observées, en comparaison avec une phylogénie en étoile comprenant n branches rayonnantes de longueur identique. Nous décrivons dans la suite le mécanisme de calcul de cette statistique, en reprenant largement les notations utilisées par [Blomberg *et al.*, 2003].

Soit $\vec{X}$ le vecteur constitué des n observations aux feuilles et $\hat{x}$ la moyenne phylogénétique calculée comme ci-dessus. L'erreur quadratique moyenne issue des observations, calculée par rapport à un modèle phylogénétique en étoile où l'on attend donc des valeurs aux feuilles suivant indépendamment les unes des autres (et donc avec des covariances nulles) des lois normales centrées en $\hat{x}$, s'écrit :

$$\mathrm{MSE}_0 = \frac{{}^t\left(\vec{X} - \hat{x}\right)\left(\vec{X} - \hat{x}\right)}{n-1} \quad , \tag{18.12}$$

où ${}^t A$ dénote la matrice transposée de A.

En réalité les observations $\vec{X}$ ne sont pas décorrélées les unes des autres, et l'erreur quadratique moyenne issue des observations *en prenant en compte un arbre candidat* $\mathscr{T}$ doit être évaluée après une transformation linéaire de $\vec{X}$ en $\vec{U}$ pour que les erreurs quadratiques soient décorrélées les unes des autres. Cette transformation s'écrit :

$$\vec{U} = D\vec{X} \tag{18.13}$$

et implique une procédure dite « méthode des moindres carrés généralisée ». On se reportera à [Grafen, 1989; Martins et Hansen, 1997; Garland Jr et Ives, 2000] pour les détails de la procédure. L'essentiel est de savoir que la matrice de transformation linéaire D vérifie $DV^tD = I_n$, où V est la matrice de variance-covariance correspondant à l'arbre candidat $\mathscr{T}$ et I_n la matrice carrée identité.

L'erreur quadratique moyenne restant après la prise en compte du système de covariances et de corrélations sur $\vec{X}$ est donc :

$$\mathrm{MSE}_{\mathscr{T}} = \frac{{}^t\left(\vec{U} - \hat{x}\right)\left(\vec{U} - \hat{x}\right)}{n-1} \tag{18.14}$$

Le ratio $\mathrm{MSE}_0/\mathrm{MSE}_{\mathscr{T}}$ quantifie la pertinence de l'arbre $\mathscr{T}$ par rapport aux données observées $\vec{X}$: un ratio élevé implique un forte corrélation entre le « signal phylogénétique » présent dans les données et les structures de variance et covariance induites par l'arbre $\mathscr{T}$. À l'inverse, un ratio tendant vers 0 indique la faiblesse ou l'absence de signal phylogénétique dans les données. Cela étant, les ratios $\mathrm{MSE}_0/\mathrm{MSE}_{\mathscr{T}}$ ne sont pas comparables d'un arbre à l'autre, mais dépendent de la taille et de la forme des arbres : un arbre avec de nombreuses bifurcations proches de la racine donne par exemple une faible valeur attendue pour ce ratio $\mathrm{MSE}_0/\mathrm{MSE}_{\mathscr{T}}$, alors qu'on s'attendra à une valeur plus élevée de ce ratio lorsque les hauteurs des nœuds internes sont plus uniformément réparties.

On montre que la valeur attendue pour le ratio $\mathrm{MSE}_0/\mathrm{MSE}_{\mathscr{T}}$ se calcule directement à partir de la matrice de variance-covariance décrivant $\mathscr{T}$:

$$\left(\frac{\mathrm{MSE}_0}{\mathrm{MSE}_{\mathscr{T}}}\right)_{\text{attendu}} = \frac{1}{n-1}\left(\mathrm{tr}(V) - \frac{n}{\sum\sum V^{-1}}\right), \tag{18.15}$$

où $\mathrm{tr}\,V$ est la trace de la matrice V, c'est-à-dire la somme de ses éléments diagonaux, et $\sum\sum V^{-1}$ la somme de tous les éléments de V^{-1}.

La statistique K peut enfin être formulée :

$$K = \frac{\left(\frac{\mathrm{MSE}_0}{\mathrm{MSE}_{\mathscr{T}}}\right)_{\text{observé}}}{\left(\frac{\mathrm{MSE}_0}{\mathrm{MSE}_{\mathscr{T}}}\right)_{\text{attendu}}} \tag{18.16}$$

Une valeur de K supérieure à 1 indique donc des corrélations entre voisins proches dans l'arbre qui sont supérieures à ce qu'on attend sous l'hypothèse d'une évolution par mouvement brownien. C'est donc le signe d'un fort signal phylogénétique. Nous utilisons la librairie R *picante* [Kembel *et al.*, 2010] pour calculer les statistiques K, grâce aux fonctions `Kcalc`, `phylosignal` et `multiPhylosignal`. La lecture et la manipulation des arbres sont implémentés en utilisant la librairie APE que l'on doit à Emmanuel Paradis [Paradis *et al.*, 2004].

18.1.3 Quantification du signal phylogénétique dans nos jeux de données en ce qui concerne la longueur des inserts

Sur les familles TreeFam

Sur les 409 familles de test TreeFam, la totalité des HMMs construits avec la règle consistant à représenter une colonne de l'alignement par un état Match si et seulement si celle-ci comprend au plus 50% de gaps donne un nombre total de 8.262 sites d'insertion dans lesquelles plus d'une séquence insère effectivement des caractères (évidemment, nul signal phylogénétique ne saurait être construit sur la base d'une observation isolée sur une seule feuille).

Si l'on s'intéresse à la distribution des p-valeurs pour la variable statistique K calculée sur ces 8.262 colonnes de longueurs d'insertion, on constate que parmi ces 8.262 sites, 4.700 ont une p-valeur pour K qui est inférieure à 0,01 et 7.010 une p-valeur inférieure à 0,1.

On préfère finalement regarder les distributions non pas des p-valeurs, mais directement des valeurs de K elles-mêmes (voir table 18.1). On effecture trois constructions de HMM, à chaque fois en prenant un critère simple pour déterminer les colonnes Match (il ne s'agit là que de tester la présence d'un signal phylogénétique dans les longueurs d'insertion) : une colonne de l'alignement est modélisée par un état Match du HMM si et seulement si le ratio de gaps contenus dans ladite colonne est inférieur à un certain seuil. Abaisser ce seuil, c'est donc rendre plus exigeantes les conditions pour former une colonne Match, et la diminution du nombre de colonnes Match entraîne dans l'absolu une diminution du nombre de sites d'insertion. Mais chacun de ces sites ayant en contrepartie une probabilité plus élevée de contenir au moins deux insertions effectives sur deux taxa différents, il n'est pas évident que le nombre d'observations pour nos alignements de zones d'insertion diminue pour autant. C'est même l'inverse qui se produit, une diminution du seuil du ratio entraînant une augmentation des observations de longueurs d'insert phylogénétiquement analysables.

Il semble donc que la présence ou l'absence de signal phylogénétique dans les lon-

	ratio maximal de gaps par colonne Match	0,2	0,5	0,8
	nb de zones d'insertion présentant au moins deux inserts	9465	8262	5975
statistique sur K	minimum	7.10^{-6}	3.10^{-6}	3.10^{-6}
	1er quartile	0,58	0,57	0,57
	médiane	1,32	1,30	1,31
	moyenne	1,96	1,93	1,91
	3e quartile	2,73	2,72	2,66
	maximum	12,89	13,09	12,39

Table 18.1. *Statistiques sur K pour les alignements TreeFam*

	ratio maximal de gaps par colonne Match	0,2	0,5	0,8
	nb de zones d'insertion présentant au moins deux inserts	1055	2756	1325
statistique sur K	minimum	0,06	0,03	0,07
	1er quartile	0,60	0,53	0,38
	médiane	0,90	0,83	0,61
	moyenne	1,13	1,20	1,16
	3e quartile	1,29	1,35	1,00
	maximum	12,67	15,00	15,00

Table 18.2. *Statistiques sur K pour les alignements SABmark*

gueurs d'insertion ne soit pas directement dépendante du mécanisme de sélection des colonnes Match. La valeur médiane de K étant toujours supérieure à 1, on peut dire que les longueurs d'insert présentent en majorité un signal phylogénétique.

Sur SABmark

Sur les 420 familles de SABmark, on obtient les statistiques présentées en table 18.2.

On voit que le signal semble en moyenne légèrement moins conservé qu'en ce qui

concerne les familles TreeFam, mais tout à fait comparable.

Par manque de temps et parce qu'on peut légitimement s'attendre à des résultats médiocres (eu égard au peu de gains apporté par la phylogénisation des autres transitions du HMM et à l'absence totale de gains apportée par la phylogénisation du contenu en émission des zones d'insertion), nous ne présentons pas ici les résultats de l'implémentation de la phylogénisation des longueurs d'insertion selon les schémas décrits au chapitre 10.

Conclusion

Nous avons montré dans cette thèse une stratégie globale de détermination des paramètres des HMM profils basée sur la phylogénie. Il s'agit là d'une tentative pour construire des modèles qui s'appuient le plus efficacement possible à la fois sur l'information « horizontale » liée aux séquences prises comme enchaînements de positions dans un alignement, et sur l'information phylogénétique reliant entre elles les séquences par des mécanismes d'évolution inférés.

La stratégie décrite porte des fruits tant en termes de vraisemblance des protéines homologues que de détection de celles-ci dans de grandes bases de données. Comme on pouvait s'y attendre, le procédé de phylogénisation est sensible à la fois à la qualité des alignements et à celle de la reconstruction phylogénétique. La décision qui a été la nôtre de construire des modèles ancestraux en des points plus ou moins arbitraires des phylogénies, à savoir les nœuds, est simple à concevoir mais n'est pas nécessairement la plus pertinente : on aurait pu décider d'échantillonner l'espace des HMM profils ancestraux en d'autres points. En effet, la décision de retenir les résultats donnés par les HMM standard pour se contenter d'*enrichir* ceux-ci avec les « hits » plus nets renvoyés par les HMM ancestraux, aurait pu être évitée en choisissant les points de reconstruction ancestrale *en fonction* de la topologie et des longueurs de branche de l'arbre. En échantillonnant par exemple à proximité d'une séquence assez divergente plutôt qu'en un nœud interne de la phylogénie forcément distant d'une telle séquence singulière, on eût peut-être obtenu des familles de HMM ancestraux plus sensibles à la diversité des séquences présentes dans les ensembles d'apprentissage.

Un autre enseignement que l'on peut tirer de notre travail est que finalement, la puissance des HMM profils doit beaucoup à des caractéristiques qui les rapprochent des profils

classiques qui les précédaient, basées sur des matrices de score position-dépendants : on remarque dans notre étude que les gains les plus nets sont obtenus via la phylogénisation des architectures de HMM et du contenu en termes d'émission sur les états Match. La phylogénisation des architectures des HMM est ici une caractéristique essentielle, permettant de capter la diversité des différents assemblages de domaines que l'on peut trouver lorsqu'on est face à un alignement de protéines distantes.

Nous attirons l'attention sur le fait que la faiblesse du support phylogénétique des caractères que sont les longueurs d'insertion ne permet sans doute pas d'obtenir des gains importants de la phylogénisation de ces longueurs d'insertion. Ceci étant, nous constatons que lorsqu'on travaille sur des familles constituées de séquences homologues très distantes, comme c'est le cas avec le banc de test SABmark, les alignements construits ne sont pas toujours pertinents et induisent des arbres phylogénétiques de grande taille. On peut alors remettre en question la fiabilité des familles proposées par ces bases de données, ou encore la capacité des logiciels d'alignement couramment utilisés à produire des alignements phylogénétiquement corrects lorsque les séquences contiennent de nombreux gaps.

Nous avons présenté systématiquement, d'une part les résultats d'amélioration de la vraisemblance des cibles à prédire, d'autre part ceux concernant l'amélioration des capacités de détection amenée par nos modèles phylo-HMM. Alors que les résultats du deuxième type ne sont pas en tant que tels contestables, un doute persiste en ce qui concerne les premiers, notamment lorsqu'on considère des figures comme la figure 16.8. Le fait de choisir comme « vraisemblance issue du modèle phylogénisé » le score maximal parmi $2n-1$ scores (si l'on compte le HMM standard en plus des HMM de nœud) introduit sans nul doute un biais. En effet, le maximum d'un ensemble de variables aléatoires distribuées normalement et centrées suit lui-même une loi d'espérance positive. Pour savoir si les accroissements de vraisemblance donnés par la phylogénisation sont statistiquement significatifs, il nous faut établir une procédure de test tenant compte de ce fait. Nous envisageons une telle procédure, que nous souhaitons publier en même temps que la méthode décrite ici. En parallèle, nous mettrons à disposition de la communauté le code de notre algorithme de construction de HMM ancestraux.

Enfin, nous sommes conscients qu'un axe d'amélioration important pour notre travail consiste à réfléchir sur la sélection des points de l'arbre phylogénétique que nous sélectionnons comme point d'intérêt pour y construire des HMM profils. Nous avons choisi par commodité de sélectionner tous les nœuds que compte la phylogénie support, mais nous aurions pu faire un choix différent et notamment éviter de phylogéniser sur les feuilles, ce qui peut sembler contre-productif (car on ignore alors les observations sur la feuille cible, cf. discussion en fin de chapitre 13.2).

Bibliographie

J. Adachi, M. Hasegawa et Institute of Statistical Mathematics : *MOLPHY version 2.3 : programs for molecular phylogenetics based on maximum likelihood*. Institute of Statistical mathematics Tokyo, 1996. Cité pages 110 et 198.

S.F. Altschul, R. Bundschuh, R. Olsen et T. Hwa : The estimation of statistical parameters for local alignment score distributions. *Nucleic Acids Research*, 29(2):351, 2001. Cité page 53.

S.F. Altschul, W. Gish, W. Miller, E.W. Myers et D.J. Lipman : Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990. Cité page 25.

S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, Z. Zhang, W. Miller et D.J. Lipman : Gapped blast and psi-blast : a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389, 1997. Cité page 25.

G.J. Barton et M.J.E. Sternberg : A strategy for the rapid multiple alignment of protein sequences : Confidence levels from tertiary structure comparisons. *Journal of Molecular Biology*, 198(2):327–337, 1987. Cité page 27.

L.E. Baum : An equality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, 3:1–8, 1972. Cité page 41.

M.P. Berger et P.J. Munson : A novel randomized iterative strategy for aligning multiple protein sequences. *Computer applications in the biosciences : CABIOS*, 7(4):479, 1991. Cité page 27.

H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, TN Bhat, H. Weissig, I.N. Shindyalov et P.E. Bourne : The protein data bank. *Nucleic acids research*, 28(1):235, 2000. Cité page 199.

E. Birney, J.A. Stamatoyannopoulos, A. Dutta, R. Guigó, T.R. Gingeras, E.H. Margulies, Z. Weng, M. Snyder, E.T. Dermitzakis, R.E. Thurman *et al.* : Identification and analysis of functional elements in 1% of the human genome by the encode pilot project. *Nature*, 447:799–816, 2007. Cité page 4.

S.P. Blomberg, T. Garland Jr et A.R. Ives : Testing for phylogenetic signal in comparative data : behavioral traits are more labile. *Evolution*, 57(4):717–745, 2003. Cité pages 221 et 225.

B. Boussau, S. Blanquart, A. Necsulea, N. Lartillot et M. Gouy : Parallel adaptations to high temperatures in the Archaean eon. *Nature*, 456(7224):942–945, 2008. Cité page 82.

N.S. Boutonnet, M.J. Rooman, M.E. Ochagavia, J. Richelle et S.J. Wodak : Optimal protein structure alignments by multiple linkage clustering : application to distantly related proteins. *Protein engineering*, 8(7):647, 1995. Cité page 175.

S.E. Brenner, P. Koehl et M. Levitt : The astral compendium for protein structure and sequence analysis. *Nucleic Acids Research*, 28(1):254, 2000. Cité page 199.

M. Brown, R. Hughey, A. Krogh, I.S. Mian, K. Sjölander et D. Haussler : Using Dirichlet mixture priors to derive hidden Markov models for protein families. *In Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, pages 47–55, 1995. Cité page 46.

R. Bundschuh : Rapid significance estimation in local sequence alignment with gaps. *Journal of Computational Biology*, 9(2):243–260, 2002. Cité page 52.

D.R. Caffrey, S. Somaroo, J.D. Hughes, J. Mintseris et E.S. Huang : Are protein–protein interfaces more conserved in sequence than the rest of the protein surface ? *Protein Science*, 13(1):190–202, 2004. Cité page 68.

J. Castresana : Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. *Molecular Biology and Evolution*, 17(4):540, 2000. Cité page 63.

J.M. Chandonia, G. Hon, N.S. Walker, L. Lo Conte, P. Koehl, M. Levitt et S.E. Brenner : The astral compendium in 2004. *Nucleic acids research*, 32(suppl 1):D189, 2004. Cité page 199.

M. Cline, C. Barrett et K. Karplus : Getting the most from your hidden Markov models, 1999. URL http://compbio.soe.ucsc.edu/ismb99.tutorial.html. Cité page 54.

F. Crick : Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970. Cité page 4.

F.H. Crick : On protein synthesis. *In Symposia of the Society for Experimental Biology*, volume 12, page 138, 1958. Cité page 4.

M.O. Dayhoff, R.M. Schwartz et B.C. Orcutt : A model of evolutionary change in proteins. *Atlas of protein sequence and structure*, 5(suppl 3):345–351, 1978. Cité pages 11, 17, 79 et 97.

M.O. Dayhoff, R.M. Schwartz et B.C. Orcutt : A model of evolutionary change in proteins. *In Atlas of protein sequence and structure*, volume 5(suppl.3), pages 353–358. M.O. Dayhoff, National biomedical research foundation, Washington DC., 1979. Cité page 35.

A.P. Dempster, N.M. Laird et D.B. Rubin : Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977. Cité page 40.

C.B. Do, M.S.P. Mahabhashyam, M. Brudno et S. Batzoglou : Probcons : probabilistic consistency-based multiple sequence alignment. *Genome Research*, 15(2):330, 2005. Cité page 29.

J.B. Domelevo Entfellner et O. Gascuel : Une approche phylo-hmm pour la recherche de séquences. *In Journées Ouvertes en Biologie, Informatique et Mathématiques*, 2008a. Cité page vii.

J.B. Domelevo Entfellner et O. Gascuel : A new phylo-hmm paradigm to search for sequences. *In Mathematics and Informatics in Evolution and Phylogeny Conference*, 2008b. Cité page vii.

J.L. Doob : The Brownian movement and stochastic equations. *The annals of Mathematics*, 43(2):351–369, 1942. Cité pages 160 et 162.

R. Durbin, S. Eddy, A. Krogh et G. Mitchison : *Biological sequence analysis*. Cambridge University Press, Cambridge, UK, 1998. Cité pages 30, 38 et 47.

S.R. Eddy : A probabilistic model of local sequence alignment that simplifies statistical significance estimation. *PLoS computational biology*, 4(5):e1000069, 2008. Cité page 53.

S.R. Eddy : A new generation of homology search tools based on probabilistic inference. *In Genome Inform*, volume 23(1), pages 205–11, 2009. Cité page 53.

R.C. Edgar : MUSCLE : multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792, 2004. ISSN 0305-1048. Cité page 26.

R.C. Edgar et K. Sjölander : Satchmo : sequence alignment and tree construction using hidden markov models. *Bioinformatics*, 19(11):1404, 2003. Cité page 116.

A.W.F. Edwards, L.L. Cavalli-Sforza, V.H. Heywood et J. McNeill : Reconstruction of evolutionary trees. *In Phenetic and phylogenetic classification*, volume 6, pages 67–76, 1964. Cité page 222.

J. Felsenstein : Maximum likelihood and minimum-steps methods for estimating evolutionary trees from data on discrete characters. *Systematic Zoology*, pages 240–249, 1973. Cité pages 80 et 82.

J. Felsenstein : Evolutionary trees from dna sequences : a maximum likelihood approach. *Journal of molecular evolution*, 17(6):368–376, 1981. Cité pages 80 et 82.

J. Felsenstein : Phylogenies and the comparative method. *American Naturalist*, pages 1–15, 1985. Cité page 225.

J. Felsenstein : Phylogenies and quantitative characters. *Annual Review of Ecology and Systematics*, 19:445–471, 1988. ISSN 0066-4162. Cité pages 158 et 159.

J. Felsenstein et G.A. Churchill : A hidden markov model approach to variation among sites in rate of evolution. *Molecular Biology and Evolution*, 13(1):93, 1996. Cité pages 89 et 106.

R.D. Finn, J. Mistry, J. Tate, P. Coggill, A. Heger, J.E. Pollington, O.L. Gavin, P. Gunasekaran, G. Ceric, K. Forslund *et al.* : The pfam protein families database. *Nucleic acids research*, 38(suppl 1):D211, 2010. Cité page 12.

W.M. Fitch : Toward defining the course of evolution : minimum change for a specific tree topology. *Systematic Biology*, 20(4):406, 1971. Cité page 76.

W.M. Fitch et E. Margoliash : A method for estimating the number of invariant amino acid coding positions in a gene using cytochrome c as a model case. *Biochemical Genetics*, 1 (1):65–71, 1967. Cité page 84.

N. Galtier et M. Gouy : Inferring pattern and process : maximum-likelihood implementation of a nonhomogeneous model of DNA sequence evolution for phylogenetic analysis. *Molecular Biology and Evolution*, 15(7):871–879, 1998. Cité page 82.

T. Garland Jr et A.R. Ives : Using the past to predict the present : confidence intervals for regression equations in phylogenetic comparative methods. *The American Naturalist*, 155(3):346–364, 2000. Cité page 226.

T. Garland Jr, P.E. Midford et A.R. Ives : An introduction to phylogenetically based statistical methods, with a new method for confidence intervals on ancestral values. *American Zoologist*, 39(2):374, 1999. Cité page 225.

M. Gerstein, E.L.L. Sonnhammer et C. Chothia : Volume changes in protein evolution. *Journal of molecular biology*, 236(4):1067–1078, 1994. Cité page 61.

P.A. Goloboff, C.I. Mattoni et A.S. Quinteros : Continuous characters analyzed as such. *Cladistics*, 22(6):589–601, 2006. Cité page 158.

O. Gotoh : An improved algorithm for matching biological sequences. *Journal of molecular biology*, 162(3):705–708, 1982. Cité page 22.

O. Gotoh : Optimal alignment between groups of sequences and its application to multiple sequence alignment. *Computer applications in the biosciences : CABIOS*, 9(3):361, 1993. Cité page 27.

O. Gotoh : A weighting system and aigorithm for aligning many phylogenetically related sequences. *Computer applications in the biosciences : CABIOS*, 11(5):543, 1995. Cité page 29.

A. Grafen : The phylogenetic regression. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 326(1233):119–157, 1989. Cité page 226.

R. Grantham : Amino acid difference formula to help explain protein evolution. *Science*, 185(4154):862, 1974. Cité page 28.

M. Gribskov, A.D. McLachlan et D. Eisenberg : Profile analysis : detection of distantly related proteins. *Proceedings of the National Academy of Sciences*, 84(13):4355, 1987. Cité pages 35 et 39.

S. Guindon et O. Gascuel : A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Systematic biology*, 52(5):696, 2003. Cité pages 77, 126 et 178.

S. Henikoff et J.G. Henikoff : Automated assembly of protein blocks for database searching. *Nucleic Acids Research*, 19(23):6565, 1991. Cité page 50.

S. Henikoff et J.G. Henikoff : Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, 89(22):10915, 1992. Cité pages 19, 21, 49, 54, 58 et 91.

S. Henikoff et J.G. Henikoff : Position-based sequence weights. *Journal of Molecular Biology*, 243(4):574–578, 1994. Cité pages 56 et 58.

M. Hirosawa, Y. Totoki, M. Hoshida et M. Ishikawa : Comprehensive study on iterative algorithms of multiple sequence alignment. *Computer applications in the biosciences : CABIOS*, 11(1):13, 1995. Cité page 27.

I. Holmes : Using guide trees to construct multiple-sequence evolutionary hmms. *Bioinformatics*, 19(suppl 1):i147, 2003. Cité pages 88 et 89.

I. Holmes et W.J. Bruno : Evolutionary hmms : a bayesian approach to multiple alignment. *Bioinformatics*, 17(9):803, 2001. Cité pages 88, 89 et 90.

X. Huang et W. Miller : A time-efficient, linear-space local similarity algorithm. *Advances in Applied Mathematics*, 12(3):337–357, 1991. Cité page 29.

L. Jin et M. Nei : Limitations of the evolutionary parsimony method of phylogenetic analysis. *Molecular Biology and Evolution*, 7(1):82, 1990. Cité page 84.

F. Johansson et H. Toh : Relative von neumann entropy for evaluating amino acid conservation. *Journal of Bioinformatics and Computational Biology*, 8(5):809–823, 2010. Cité page 70.

D.T. Jones, W.R. Taylor et J.M. Thornton : The rapid generation of mutation data matrices from protein sequences. *Computer applications in the biosciences : CABIOS*, 8(3):275, 1992. Cité page 79.

S. Karlin et S.F. Altschul : Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences*, 87(6):2264, 1990. Cité page 52.

K. Katoh, K. Kuma, H. Toh et T. Miyata : Mafft version 5 : improvement in accuracy of multiple sequence alignment. *Nucleic acids research*, 33(2):511, 2005. Cité page 29.

K. Katoh, K. Misawa, K. Kuma et T. Miyata : MAFFT : a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic acids research*, 30(14):3059, 2002. ISSN 0305-1048. Cité pages 26, 28 et 200.

K. Katoh et H. Toh : Recent developments in the mafft multiple sequence alignment program. *Briefings in bioinformatics*, 9(4):286, 2008. Cité page 29.

S.W. Kembel, P.D. Cowan, M.R. Helmus, W.K. Cornwell, H. Morlon, D.D. Ackerly, S.P. Blomberg et C.O. Webb : Picante : R tools for integrating phylogenies and ecology. *Bioinformatics*, 26(11):1463, 2010. Cité page 227.

W.J. Kent : Blat—the blast-like alignment tool. *Genome research*, 12(4):656, 2002. Cité page 25.

P. Klosterman, A. Uzilov, Y. Bendaña, R. Bradley, S. Chao, C. Kosiol, N. Goldman et I. Holmes : Xrate : a fast prototyping, training and annotation tool for phylo-grammars. *BMC bioinformatics*, 7(1):428, 2006. Cité page 201.

A. Krogh : Hidden Markov models for labeled sequences. *In Pattern Recognition, 1994. Vol. 2-Conference B : Computer Vision & Image Processing., Proceedings of the 12th IAPR International. Conference on*, volume 2, pages 140–144. IEEE, 1994. Cité pages 36 et 39.

A. Krogh, M. Brown, I.S. Mian, K. Sjölander et D. Haussler : Hidden Markov models in computational biology. applications to protein modeling. *Journal of Molecular Biology*, 235(5):1501–1531, 1994. Cité pages 36, 39, 43, 91, 100, 109, 116 et 150.

A. Krogh et G. Mitchison : Maximum entropy weighting of aligned sequences of proteins or dna. *In Proc. Int. Conf. on Intelligent Systems in Molecular Biology*, volume 3, pages 215–221, 1995. Cité pages 56 et 59.

S. Kullback et R.A. Leibler : On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. Cité page 66.

E.S. Lander, L.M. Linton, B. Birren, C. Nusbaum, M.C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh *et al.* : Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001. Cité page 4.

S.Q. Le et O. Gascuel : An improved general amino acid replacement matrix. *Molecular biology and evolution*, 25(7):1307, 2008. ISSN 0737-4038. Cité pages 79, 135, 175 et 219.

H. Li, A. Coghlan, J. Ruan, L.J. Coin, J.K. Heriche, L. Osmotherly, R. Li, T. Liu, Z. Zhang, L. Bolund *et al.* : Treefam : a curated database of phylogenetic trees of animal gene families. *Nucleic acids research*, 34(suppl 1):D572, 2006. Cité pages ix et 174.

A. Löytynoja et N. Goldman : An algorithm for progressive multiple alignment of sequences with insertions. *Proceedings of the National Academy of Sciences of the United States of America*, 102(30):10557, 2005. ISSN 0027-8424. Cité pages 31, 90 et 126.

A. Löytynoja et N. Goldman : Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. *Science*, 320(5883):1632, 2008. Cité pages 31, 90 et 126.

R. Lüthy, I. Xenarios et P. Bucher : Improving the sensitivity of the sequence profile method. *Protein Science*, 3(1):139–146, 1994. Cité page 55.

M. Madera et J. Gough : A comparison of profile hidden Markov model procedures for remote homology detection. *Nucleic acids research*, 30(19):4321, 2002. Cité page 47.

E.P. Martins et T.F. Hansen : Phylogenies and the comparative method : A general approach to incorporating phylogenetic information into the analysis of interspecific data. *The American Naturalist*, 149(4):646–667, 1997. Cité page 226.

S.C. Meyer : The origin of biological information and the higher taxonomic categories. *In Proc. of the Biological Society of Washington*, volume 117, pages 213–239, 2004. Cité page 73.

G. Mitchison et R. Durbin : Tree-based maximal likelihood substitution matrices and hidden Markov models. *Journal of Molecular Evolution*, 41(6):1139–1151, 1995. ISSN 0022-2844. Cité pages vi, vii, 88, 91, 94, 98, 100, 108, 110, 137, 140, 144, 145, 197 et 198.

G.J. Mitchison : A probabilistic treatment of phylogeny and sequence alignment. *Journal of molecular evolution*, 49(1):11–22, 1999. ISSN 0022-2844. Cité pages 88, 100, 101, 137, 140 et 197.

D.A. Morrison et J.T. Ellis : Effects of nucleotide sequence alignment on phylogeny estimation : A case study of 18s rdnas of apicomplexa. *Molecular Biology and Evolution*, 14 (4):428, 1997. Cité page 63.

R. Mott : Maximum-likelihood estimation of the statistical distribution of smith-waterman local sequence similarity scores. *Bulletin of Mathematical Biology*, 54(1):59–75, 1992. Cité page 52.

A.G. Murzin, S.E. Brenner, T. Hubbard et C. Chothia : Scop : a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540, 1995. Cité pages 12, 174, 198 et 199.

S.B. Needleman et C.D. Wunsch : A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970. Cité page 22.

M. Nei et T. Gojobori : Simple methods for estimating the numbers of synonymous and nonsynonymous nucleotide substitutions. *Molecular biology and evolution*, 3(5):418, 1986. Cité page 84.

C. Notredame, D.G. Higgins et J. Heringa : T-coffee : a novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, 302(1):205–217, 2000. Cité page 29.

B.K. Øksendal : *Stochastic differential equations : an introduction with applications*. Springer Verlag, 2003. Cité pages 160 et 162.

R. Olsen, R. Bundschuh et T. Hwa : Rapid assessment of extremal statistics for gapped local alignment. *In Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 211–222. AAAI Press : Menlo Park, CA, 1999. Cité page 53.

E. Paradis, J. Claude et K. Strimmer : Ape : analyses of phylogenetics and evolution in r language. *Bioinformatics*, 20(2):289, 2004. Cité page 227.

F.M.G. Pearl, CF Bennett, J.E. Bray, A.P. Harrison, N. Martin, A. Shepherd, I. Sillitoe, J. Thornton et C.A. Orengo : The cath database : an extended protein family resource for structural and functional genomics. *Nucleic acids research*, 31(1):452, 2003. Cité page 12.

W.R. Pearson et D.J. Lipman : Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8):2444, 1988. Cité page 25.

W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling *et al.* : *Numerical recipes*, volume 3. Cambridge Univ Press, 2007. Cité page 131.

B. Qian et R.A. Goldstein : Distribution of indel lengths. *Proteins : Structure, Function, and Bioinformatics*, 45(1):102–104, 2001. Cité page 151.

B. Qian et R.A. Goldstein : Detecting distant homologs using phylogenetic tree-based HMMs. *Proteins : Structure, Function, and Bioinformatics*, 52(3):446–453, 2003. ISSN 1097-0134. Cité pages vi, vii, 108, 109, 110, 111, 133, 135, 136, 137, 140, 167, 168, 197, 198, 200, 202 et 203.

B. Qian et R.A. Goldstein : Performance of an iterated T-HMM for homology detection. *Bioinformatics*, 20(14):2175, 2004. ISSN 1367-4803. Cité pages 111, 133, 135, 136, 137, 140 et 168.

L.R. Rabiner : A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. Cité page 37.

N. Saitou et M. Nei : The neighbor-joining method : a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, 4(4):406, 1987. Cité page 26.

C. Sander et R. Schneider : The hssp data base of protein structure-sequence alignments. *Nucleic acids research*, 21(13):3105, 1993. Cité page 96.

F. Servant, C. Bru, S. Carrère, E. Courcelle, J. Gouzy, D. Peyruc et D. Kahn : Prodom : automated clustering of homologous domains. *Briefings in Bioinformatics*, 3(3):246, 2002. Cité page 12.

C.E. Shannon : A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948. Cité page 65.

I.N. Shindyalov et P.E. Bourne : Protein structure alignment by incremental combinatorial extension (ce) of the optimal path. *Protein engineering*, 11(9):739, 1998. Cité pages 110, 175 et 197.

J. Shlens : Notes on Kullbak-Leibler divergence and likelihood theory, 2007. URL http://www.snl.salk.edu/~shlens/kl.pdf. Cité page 66.

P.R. Sibbald et P. Argos : Weighting aligned protein or nucleic acid sequences to correct for unequal representation. *Journal of Molecular Biology*, 216(4):813–818, 1990. Cité page 57.

A. Siepel et D. Haussler : Combining phylogenetic and hidden markov models in biosequence analysis. *In Proceedings of the seventh annual international conference on Research in computational molecular biology*, pages 277–286. ACM, 2003. Cité page 103.

A. Siepel et D. Haussler : Combining phylogenetic and hidden markov models in biosequence analysis. *Journal of Computational Biology*, 11(2-3):413–428, 2004a. Cité pages 88 et 103.

A. Siepel et D. Haussler : Phylogenetic estimation of context-dependent substitution rates by maximum likelihood. *Molecular Biology and Evolution*, 21(3):468, 2004b. Cité page 107.

A. Siepel et D. Haussler : Phylogenetic hidden Markov models. *Statistical methods in molecular evolution*, pages 325–351, 2005. Cité pages vi et 88.

C.J.A. Sigrist, L. Cerutti, E. De Castro, P.S. Langendijk-Genevaux, V. Bulliard, A. Bairoch et N. Hulo : Prosite, a protein domain database for functional characterization and annotation. *Nucleic acids research*, 38(suppl 1):D161, 2010. Cité page 12.

K. Sjölander, K. Karplus, M. Brown, R. Hughey, A. Krogh, I.S. Mian et D. Haussler : Dirichlet mixtures : a method for improved detection of weak but significant protein sequence homology. *Computer applications in the biosciences : CABIOS*, 12(4):327, 1996. Cité pages 45 et 50.

T.F. Smith et M.S. Waterman : Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981. Cité page 24.

R.R. Sokal et C.D. Michener : A statistical method for evaluating systematic relationships. *Univ. Kans. Sci. Bull.*, 38:1409–1438, 1958. Cité page 26.

D.J. States et W. Gish : Combined use of sequence similarity and codon bias for coding region identification. *Journal of computational biology*, 1(1):39–50, 1994. ISSN 1066-5277. URL http://view.ncbi.nlm.nih.gov/pubmed/8790452. Cité page 25.

A.R. Subramanian, M. Kaufmann et B. Morgenstern : Dialign-tx : greedy and progressive approaches for segment-based multiple sequence alignment. *Algorithms Mol. Biol*, 3(6), 2008. Cité pages 26 et 27.

G. Talavera et J. Castresana : Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments. *Systematic Biology*, 56 (4):564, 2007. Cité page 63.

W.R. Taylor : The classification of amino acid conservation. *Journal of Theoretical Biology*, 119(2):205–218, 1986. Cité pages 10 et 11.

J.W. Thomas, J.W. Touchman, R.W. Blakesley, G.G. Bouffard, S.M. Beckstrom-Sternberg, E.H. Margulies, M. Blanchette, A.C. Siepel, P.J. Thomas, J.C. McDowell *et al.* : Comparative analyses of multi-species sequences from targeted genomic regions. *Nature*, 424 (6950):788–793, 2003a. Cité page 106.

P.D. Thomas, A. Kejariwal, M.J. Campbell, H. Mi, K. Diemer, N. Guo, I. Ladunga, B. Ulitsky-Lazareva, A. Muruganujan, S. Rabkin *et al.* : Panther : a browsable database of gene products organized by biological function, using curated protein family and subfamily classification. *Nucleic acids research*, 31(1):334, 2003b. Cité page 12.

J.D. Thompson, D.G. Higgins et T.J. Gibson : CLUSTAL W : improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, 22(22):4673, 1994a. ISSN 0305-1048. Cité pages 26, 29 et 90.

J.D. Thompson, D.G. Higgins et T.J. Gibson : Improved sensitivity of profile searches through the use of sequence weights and gap excision. *Computer applications in the biosciences : CABIOS*, 10(1):19, 1994b. Cité pages 55 et 59.

J.D. Thompson, B. Linard, O. Lecompte et O. Poch : A comprehensive benchmark study of multiple sequence alignment methods : Current challenges and future perspectives. *PLoS ONE*, 6:e18093, 03 2011. URL http://dx.doi.org/10.1371%2Fjournal.pone.0018093. Cité page 90.

J.D. Thompson, F. Plewniak et O. Poch : Balibase : a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, 15(1):87, 1999a. Cité page 90.

J.D. Thompson, F. Plewniak et O. Poch : A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Research*, 27(13):2682, 1999b. Cité page 90.

J.L. Thorne, H. Kishino et J. Felsenstein : An evolutionary model for maximum likelihood alignment of dna sequences. *Journal of Molecular Evolution*, 33(2):114–124, 1991. Cité pages 88 et 89.

J.L. Thorne, H. Kishino et J. Felsenstein : Inching toward reality : an improved likelihood model of sequence evolution. *Journal of Molecular Evolution*, 34(1):3–16, 1992. Cité pages 88 et 89.

G.E. Uhlenbeck et L.S. Ornstein : On the theory of the Brownian motion. *Physical Review*, 36(5):823, 1930. Cité pages 159 et 160.

I. Van Walle, I. Lasters et L. Wyns : Consistency matrices : quantified structure alignments for sets of related proteins. *Proteins : Structure, Function, and Bioinformatics*, 51(1):1–9, 2003. Cité page 175.

I. Van Walle, I. Lasters et L. Wyns : Sabmark—a benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics*, 21(7):1267, 2005. Cité pages ix et 174.

J.C. Venter, M.D. Adams, E.W. Myers, P.W. Li, R.J. Mural, G.G. Sutton, H.O. Smith, M. Yandell, C.A. Evans, R.A. Holt *et al.* : The sequence of the human genome. *Science*, 291(5507): 1304, 2001. Cité page 4.

M. Vingron et P. Argos : A fast and sensitive multiple sequence alignment algorithm. *Computer applications in the biosciences : CABIOS*, 5(2):115, 1989. Cité page 56.

M. Vingron et P.R. Sibbald : Weighting in sequence space : a comparison of methods in terms of generalized sequences. *Proceedings of the National Academy of Sciences*, 90 (19):8777, 1993. Cité page 56.

G. Weiss : Time-reversibility of linear stochastic processes. *Journal of Applied Probability*, pages 831–836, 1975. Cité page 162.

S. Whelan et N. Goldman : A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Molecular Biology and Evolution*, 18(5):691, 2001. Cité pages 79, 109 et 136.

Z. Yang : Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. *Molecular Biology and Evolution*, 10(6):1396, 1993. ISSN 0737-4038. Cité pages 84 et 85.

Z. Yang : Maximum likelihood phylogenetic estimation from dna sequences with variable rates over sites : approximate methods. *Journal of Molecular Evolution*, 39(3):306–314, 1994. Cité page 85.

Z. Yang : Phylogenetic analysis using parsimony and likelihood methods. *Journal of Molecular Evolution*, 42(2):294–307, 1996. Cité page 77.

Z. Yang : Paml : a program package for phylogenetic analysis by maximum likelihood. *Computer applications in the biosciences : CABIOS*, 13(5):555, 1997. Cité pages 110, 178 et 198.

J. Zhang et M. Nei : Accuracies of ancestral amino acid sequences inferred by the parsimony, likelihood, and distance methods. *Journal of molecular evolution*, 44:139–146, 1997. Cité page 77.

Abstract

Statistical modelling of homologous sequences through profile HMM disregards the phylogenetic links between those. Here we present models harnessing an efficient combination of horizontal and vertical features, simultaneously figuring sequences as chains of aminoacids and products of an evolutionary process. Such models belong to the phylo-HMM family introduced in the '90s (e.g. Mitchison & Durbin). Focusing on the detection of remote homologues in databases, we develop a framework for an exhaustive derivation of phylo-HMM parameters basing on the phylogeny. The models we build are ancestral reconstruction HMM, output by a process of phylogenetic inference of conserved positions, Match and Insert emission probabilities, and transition probabilities. Finally, we propose new models of evolution for transitions between states of the HMM and for insert lengths. The training framework we describe has been implemented and tried on testbenches of homologous sequences. It brings improved likelihoods and a better discriminative power on detecting remote homologues in large databases of proteins sequences.

Keywords: *HMM, models, biological sequences, phylogenies, phylo-HMM, tree-HMM, remote homologues*

Résumé

La modélisation statistique de séquences homologues par HMM profils laisse de côté l'information phylogénétique reliant les séquences. Nous proposons ici des modèles combinant efficacement analyse longitudinale (séquences protéiques vues comme des enchaînements d'acides aminés) et verticale (séquences vues comme étant le produit d'une évolution le long des branches d'un arbre phylogénétique). De tels modèles appartiennent à la famille des phylo-HMM, introduite dans le courant des années 1990 (Mitchison & Durbin). Notre objectif étant la détection d'homologues distants dans les bases de données, nous décrivons une méthodologie de dérivation complète des paramètres des phylo-HMM profils basée sur la phylogénie : les modèles que nous proposons sont des HMM de reconstruction ancestrale, issus d'un processus d'inférence phylogénétique des positions conservées, des probabilités d'émission de caractères sur les états Match et Insertion, ainsi que des probabilités de transition entre états du HMM. Nous suggérons notamment une nouvelle modélisation pour l'évolution des transitions entre états du HMM, ainsi qu'un modèle de type Ornstein-Uhlenbeck pour l'évolution des longueurs des insertions. Contraintes évolutives et contraintes longitudinales sont ainsi simultanément prises en compte. Le processus d'apprentissage développé a été implémenté et testé sur une base de données de familles de séquences homologues, mettant en évidence des gains à la fois en termes de vraisemblance accrue des homologues distants et en termes de performance lorsqu'il s'agit de détecter ceux-ci dans les grandes bases de données protéiques.

Mots clefs : *HMM, modélisation, séquences biologiques, phylogénies, phylo-HMM, tree-HMM, homologues distants*

LIRMM — 161, rue Ada — 34095 Montpellier Cedex 5 — France