Méthodes efficaces pour reconstruire de grandes phylogénies suivant le principe du maximum de vraisemblance

Vincent Ranwez

NUMERO D'IDENTIFICATION

ACADEMIE DE MONTPELLIER

UNIVERSITE DE MONTPELLIER II

– SCIENCES ET TECHNIQUES DU LANGUEDOC –

Thèse de Doctorat de l'Université

Discipline :	Informatique
Formation Doctorale :	Informatique
Ecole Doctorale :	Information, Structures, Systèmes

Méthodes efficaces pour reconstruire de grandes phylogénies suivant le principe du maximum de vraisemblance

par

Vincent Ranwez

Soutenance prévue le *6 novembre 2002 à 14h* devant le jury composé de :

M. Manolo Gouy,	Directeur de recherche, CNRS Lyon,	Rapporteur
M. Alain Guénoche,	Chargé de recherche, CNRS/LIM Marseille,	Rapporteur
Mme. Nora Benhabiles,	Responsable de la bio-informatique chez Clinigenetics,	Examinateur
M. Nicolas Galtier,	Chargé de recherche, CNRS/UM2 Montpellier,	Examinateur
M. Olivier Gascuel,	Directeur de recherche, CNRS/LIRMM Montpellier,	Directeur de Thèse
M. Alain Jean-Marie,	Professeur, Université Montpellier II/LIRMM,	Examinateur

Introduction générale

Différentes hypothèses sur le transformisme des espèces ont été proposées dès l'antiquité gréco-romaine. La première réelle tentative d'élaborer une théorie de l'évolution est généralement attribuée à Lamarck (1809). Sa théorie s'opposait de manière directe à la théorie dominante de la préformation et du fixisme des espèces. Cette dernière, fortement soutenue par l'ordre religieux mais aussi par la plupart des scientifiques de l'époque et notamment par Cuvier, est restée la théorie dominante jusqu'à la parution de "L'Origine des Espèces" de Darwin (1859). La théorie de l'évolution fournit alors une nouvelle manière d'aborder la classification des espèces. Darwin souligne le lien étroit entre classification et phylogénie : "le lien que nous révèlent partiellement nos classifications, lien déguisé comme il l'est par divers degrés de modifications, n'est autre que la communauté de descendance, la seule cause connue de la similitude des êtres organisés". Cette vision ajoute une dimension temporelle à la classification et, depuis Darwin, les arbres sont utilisés comme support graphique pour représenter simultanément l'aspect temporel de l'évolution et les groupements d'espèces qui en découlent.

Cette vision évolutionniste change radicalement la manière d'appréhender la classification, il ne s'agit plus d'établir une classification pratique du vivant, mais de retrouver un ordre naturel intrinsèque. Pourtant les méthodes de classification sont longtemps restées basées sur la comparaison de caractères morphologiques, et ce n'est qu'assez récemment que des classifications prenant en compte un modèle d'évolution sont apparues. Dans les années soixante, la biologie moléculaire a donné accès aux génomes des espèces et l'apparition des premiers ordinateurs a fourni des outils capables de traiter ces nouvelles données. Plusieurs publications ont alors montré que les données moléculaires permettent de reconstruire des phylogénies cohérentes avec les classifications antérieures (fondées sur l'étude des fossiles et des caractères morphologiques).

Ces premiers résultats ont montré l'intérêt de disposer de méthodes efficaces pour reconstruire de manière fiable et automatique l'histoire évolutive d'un ensemble de séquences. L'étude et l'amélioration de ces méthodes constituent une discipline (la reconstruction phylogénétique) à la frontière des mathématiques, de la biologie et de l'informatique. Les phylogénies ainsi reconstruites permettent de disposer d'informations jusque là inaccessibles et sont utilisées pour aborder de nombreux problèmes biologiques (Harvey, May et Nee 1996). Les deux exemples suivants, développés dans (Page et Holmes 1998), montrent l'étendue des champs d'application de la reconstruction phylogénétique.

Jusqu'à une période récente, les organismes cellulaires étaient divisés en deux grandes familles, ceux dont les cellules possèdent un noyau (les eucaryotes) et ceux qui n'en possède pas (les procaryotes). En s'appuyant sur la phylogénie de séquences moléculaires évoluant lentement, Woese et Fox (1977) ont montré qu'il existait deux groupes très différents des procariotes : les eubactéries et les archaebactéries. Malgré l'absence de noyau dans leurs cellules, l'étude de la phylogénie moléculaire a montré que les archaebactéries sont, à certains égards, plus proches des eucaryotes que des eubactéries. Ainsi, grâce à la biologie moléculaire, l'arbre universel du vivant s'est enrichi d'une branche supplémentaire, et un des problèmes récurrents aujourd'hui est d'enraciner cet arbre pour décider, *in fine*, de la position des archaebactéries.

L'analyse de phylogénies moléculaires permet également d'effectuer des études épidémiologiques. Dans ce cas, on utilise des séquences évoluant très rapidement. Par exemple, dans les années 90, le "*center for disease control*" d'Atlanta, à reçu un rapport surprenant concernant une jeune femme séropositive. En effet, d'après ce rapport, le seul lien entre cette patiente et le virus était d'avoir consulté un dentiste porteur du virus. Après enquête, il s'est avéré que d'autres patients de ce dentiste avaient, eux aussi, contracté le virus du SIDA. Le "*center for disease control*" a donc réalisé une analyse moléculaire des souches du virus présentes chez le dentiste, chez ses patients, et chez d'autres malades n'ayant jamais consulté ce dentiste. La phylogénie moléculaire de ces souches virales à permis de confirmer que le dentiste avait effectivement contaminé ses patients. Des précautions sanitaires supplémentaires ont donc pu être mises en place pour éviter ce type de contamination.

Ces deux exemples, et il en existe de nombreux autres, notamment en pharmacologie ou pour l'amélioration des plantes, montrent l'importance d'avoir des méthodes permettant d'obtenir des phylogénies moléculaires fiables. La méthode la plus fiable actuellement pour reconstruire une phylogénie à partir de séquences nucléotidiques, semble être la méthode du maximum de vraisemblance. Cette méthode utilise un modèle mathématique du processus d'évolution des séquences pour définir la probabilité qu'une phylogénie puisse produire les séquences observées, et cherche la phylogénie pour laquelle cette probabilité est maximale. Les méthodes classiques pour rechercher la phylogénie de vraisemblance maximale deviennent très coûteuses en temps de calcul lorsque le nombre de séquences augmente. Lorsque l'on souhaite reconstruire la phylogénie d'un grand nombre de séquences, il est donc impossible d'utiliser directement ce type de méthodes. Dans cette thèse, nous cherchons donc à définir des heuristiques efficaces pour reconstruire de grandes phylogénies suivant le principe du maximum de vraisemblance.

Il existe actuellement deux types de méthodes permettant, d'une certaine manière, de reconstruire de grandes phylogénies suivant le principe du maximum de vraisemblance : les méthodes de distances et les méthodes de quadruplets. Toutes deux divisent le problème initial en sous-problèmes contenant peu de séquences. Elles peuvent alors

résoudre rapidement chacun de ces sous-problèmes, puis combiner les solutions obtenues pour proposer une phylogénie de l'ensemble des séquences. Les méthodes de quadruplets divisent le problème initial en sous-problèmes de quatre séquences, et reconstruisent la phylogénie globale en s'appuyant sur la topologie (strucuture) obtenue pour chaque quadruplet. Les méthodes de distances divisent le problème initial en sous-problèmes ne contenant que deux séquences, et reconstruisent la phylogénie globale en s'appuyant sur les distances obtenues pour chaque paire de séquences.

Les travaux présentés dans cette thèse étudient les faiblesses de ces deux types de méthodes afin d'en proposer de nouvelles variantes ayant des performances supérieures à celles des heuristiques existantes. Ces travaux ont fait l'objet de trois publications internationales, qui sont fournies en annexe, et sur lesquelles nous nous appuyons tout au long de ce mémoire.

La rédaction de cette thèse est organisée en quatre chapitres.

- Le premier chapitre introduit les notions préalables sur les données moléculaires et leurs modélisations ainsi que sur les arbres utilisés pour représenter les phylogénies.
- Le second chapitre décrit les principales méthodes de reconstruction phylogénétique.
- Le troisième chapitre est constitué d'un état de l'art sur les méthodes de quadruplets et d'un résumé de nos travaux sur ces méthodes. Il décrit l'algorithme Weight Optimisation (Ranwez et Gascuel 2001a ; Ranwez et Gascuel 2001b) qui améliore sensiblement les performances de la méthode de quadruplets proposée par (Strimmer et Von Haeseler 1996 ; Strimmer, Goldman et Von Haeseler 1997). Il résume également les observations qui nous ont conduits à abandonner l'utilisation des méthodes de quadruplets en reconstruction phylogénétique (Ranwez et Gascuel 2001b).
- Le quatrième chapitre décrit la méthode NJ+TRIPLEML (Ranwez et Gascuel 2002). TRIPLEML est une nouvelle manière d'estimer les distances utilisées par la méthode de distances dite de Neighbor Joining (NJ) (Saitou et Nei 1987). TRIPLEML estime l'ensemble des distances utilisées par ces méthodes à partir d'une approche de maximum de vraisemblance sur des triplets de séquences (ou de groupes de séquences). L'utilisation de TRIPLEML permet d'augmenter significativement la fiabilité des arbres reconstruits par NJ et ses variantes.

Chapitre 1 Notions préliminaires

Ce chapitre présente les notions fondamentales permettant de comprendre les fondements et les difficultés de la reconstruction phylogénétique. Il est constitué de trois parties. La première partie introduit les définitions et les propriétés des arbres qui sont utilisés pour représenter des histoires évolutives. La seconde partie rappelle les propriétés et les définitions importantes concernant les données moléculaires. La troisième partie présente les modèles d'évolution les plus couramment utilisés pour modéliser l'évolution des séquences nucléotidiques.

1.1 Arbres – phylogénies

Depuis Darwin, les arbres sont utilisés comme support pour représenter l'aspect temporel de l'évolution et les regroupements d'espèces qui en découlent. Cette partie présente la terminologie relative à cette représentation, donne un aperçu de la combinatoire des phylogénies et précise les principales propriétés des phylogénies valuées.

1.1.1 Définitions

L'hypothèse fondamentale de la reconstruction phylogénétique est que l'histoire évolutive des espèces se déroule par spéciations successives. Suivant cette hypothèse, une lignée ancestrale peut, par spéciation, donner le jour à deux nouvelles lignées, et une phylogénie peut être représentée par un arbre similaire à ceux de la Figure 1. Ce paragraphe introduit plusieurs concepts qui sont fréquemment utilisés tout au long de cette thèse. Certaines des notions évoquées ci-dessous s'appuient sur des notions élémentaires de la théorie des graphes. Des définitions formelles de ces notions sont proposées par Berge (1970).

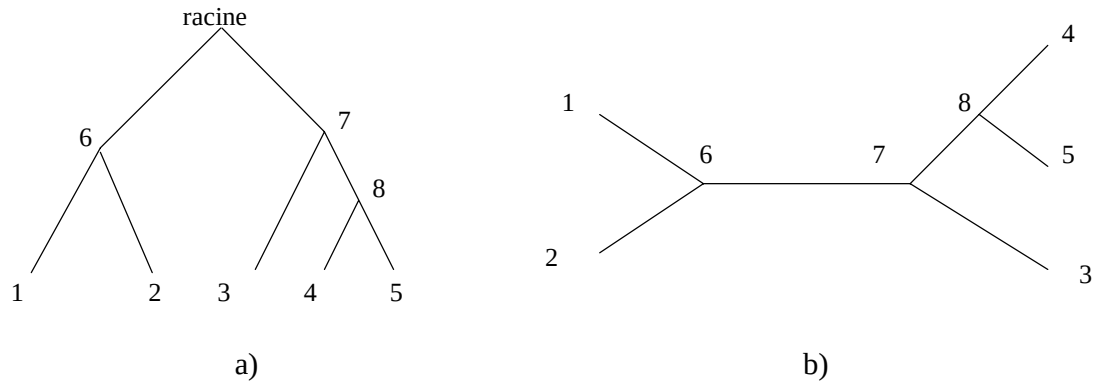


Figure 1: représentations arborées d'une phylogénie

a) Arbre enraciné à 5 feuilles, la notation parenthésée de cet arbre est ((1,2),(3,(4,5)) ;
b) Version non enracinée de a), une notation parenthésée de cet arbre est ((1,2),3,(4,5)) ;

Une phylogénie enracinée T pour un ensemble de séquences $E = \{1,2,\ldots n\}$ est un arbre dont les feuilles sont bijectivement associées aux séquences de E, et qui possède un seul sommet de degré 2 qui en est la racine. Les feuilles de la phylogénie représentant les séquences contemporaines étudiées sont parfois appelées nœuds externes. Les autres nœuds de l'arbre, qui correspondent à d'hypothétiques séquences ancestrales des séquences étudiées, sont parfois appelés nœud internes. Un nœud interne peut aussi être vu comme un événement de spéciation. La racine de la phylogénie est le seul nœud interne de degré deux qui représente une spéciation. Les nœuds internes de degré supérieur à trois peuvent représenter le fait que plusieurs spéciations ont eu lieu et que l'on ne sait pas dans quel ordre elles se sont produites. Ils peuvent également représenter le fait qu'une seule spéciation a engendré plus de deux nouvelles espèces. On considère généralement que ce dernier phénomène est peu probable, d'où la définition suivante : une phylogénie enracinée est dite complètement résolue si tous ses nœuds internes, autres que sa racine, sont de

degrés 3. Une phylogénie qui ne possède pas de nœud de degré deux est dite non-enracinée.

Soit T une phylogénie contenant une branche (n_1, n_2) qui relie le nœud n_1 au nœud n_2. Si l'on retire cette branche de T, on obtient deux composantes connexes. Soit E_1 et E_2 les sous-ensembles d'espèces (les feuilles de T) appartenant respectivement à ces deux composantes : $E_1|E_2$ constitue alors une bipartition de E. On dit que la bipartition $E_1|E_2$ est induite par la phylogénie T ou par la branche (n_1, n_2). Par exemple dans la Figure 1.b, où l'ensemble des espèces étudiées est $E = \{1,2,3,4,5\}$, la branche (1, 6) induit la bipartition $\{1\}|\{2,3,4,5\}$ et la branche (6, 7) induit la bipartition $\{1,2\}|\{3,4,5\}$. L'ensemble des bipartitions induites par T caractérise complètement la phylogénie T (Buneman 1971). On peut donc comparer deux phylogénies en comparant leurs bipartitions. En particulier, toutes les phylogénies portant sur le même ensemble de séquences (ayant les mêmes feuilles) ont en commun les bipartitions induites par leurs arêtes externes ; ces bipartitions ne sont donc pas réellement informatives. Dans le cas où, comme dans la Figure 1.b, l'ensemble des espèces étudiées est $E = \{1,2,3,4,5\}$, il y a donc cinq bipartitions triviales : $\{1\}|\{2,3,4,5\}$, $\{2\}|\{1,3,4,5\}$, $\{3\}|\{1,2,4,5\}$, $\{4\}|\{1,2,3,5\}$ et $\{5\}|\{1,2,3,4\}$. Pour comparer deux phylogénies T_1 et T_2 on utilise souvent la distance dite de "Robinson et Foulds" qui correspond au nombre de bipartitions (non triviales) qui sont induites par une seule des deux phylogénies (Robinson et Foulds 1981).

Les méthodes de reconstruction phylogénétique reconstruisent généralement des phylogénies non-enracinées. Dans la suite de cette thèse, nous parlerons toujours, sauf mention contraire, de phylogénie non-enracinée.

1.1.2 Aspect combinatoire

Le nombre de phylogénies distinctes possibles pour un ensemble de n séquences contemporaines augmente très rapidement lorsque n augmente.

En partant d'une phylogénie T_i ayant i feuilles, on peut obtenir une phylogénie T_{i+1} ayant $(i+1)$ feuilles en greffant la feuille supplémentaire sur une branche quelconque de T_i. Les phylogénies ainsi obtenues sont toutes distinctes, et possèdent toutes deux branches de plus que la phylogénie T_i. Pour 2 séquences, il existe une seule phylogénie, et elle possède une seule branche. Pour $n \geq 2$, le nombre de branches d'une phylogénie complètement résolue ayant n feuilles est donc $1+2(n-2)$ soit $2n-3$. L'insertion de la $i^{\text{ème}}$ feuille peut donc se faire sur $2(i-1)-3 = 2i-5$ branches, et le nombre phylogénies complètement résolues distinctes à n feuilles vaut :

$$\prod_{i=3}^{n}(2i-5) = 1\times 3\times 5\times 7\times\ldots\times(2n-5) \qquad \textbf{(1)}$$

Pour 4 feuilles, il y a donc 3 phylogénies complètement résolues possibles ; il y en a 15 pour 5 feuilles, et plus de 2 millions pour 10 feuilles. Comme nous le verrons dans le

chapitre suivant, cette rapide augmentation du nombre de topologies possibles en fonction du nombre de séquences étudiées est un problème majeur en reconstruction phylogénétique.

1.1.3 Phylogénies valuées

Nous avons pour l'instant omis l'aspect temporel de la phylogénie. Cet aspect temporel peut être représenté par une valuation positive des branches de l'arbre, leurs longueurs, et on dira alors que la phylogénie est valuée. Dans une phylogénie valuée, la longueur d'une branche (n_1, n_2) représente la distance évolutive qui sépare n_1 de n_2 et la distance entre deux nœuds correspond à la somme des longueurs des branches qui sont sur le chemin reliant ces deux nœuds. Par exemple, dans la phylogénie de la Figure 1.b, la distance séparant les feuilles 1 et 3 est la somme des longueurs des trois branches (1, 6), (6, 7) et (7, 3) qui constituent le chemin qui relie ces deux feuilles. En particulier, pour chaque couple de feuilles *i*, *j* d'une phylogénie valuée positivement, on peut obtenir la distance d_{ij} qui sépare ces deux éléments de *E*. La mesure de distance *d* ainsi obtenue est appelée distance d'arbre.

Ainsi toute phylogénie valuée positivement correspond à une distance d'arbre sur *E*. Inversement, toute distance d'arbre correspond à une phylogénie unique (Smolenskii 1969). La *condition des quatre points* (Zarestkii 1965 ; Buneman 1971) fournit une caractérisation pratique des distances d'arbres. Une mesure δ sur *E*, qui est positive ($\delta_{ij} \geq 0$), réflexive ($\delta_{ij} = 0 \Leftrightarrow i = j$) et symétrique ($\delta_{ij} = \delta_{ji}$), est une distance d'arbre si et seulement si, pour tout sous-ensemble $\{x, y, z, t\}$ contenant quatre éléments de *E*, les deux plus grandes des trois sommes $\delta_{xy} + \delta_{zt}$, $\delta_{xz} + \delta_{yt}$ et $\delta_{xt} + \delta_{yz}$ sont égales. De plus, si l'on suppose, par exemple, que $\delta_{xy} + \delta_{zt}$ est la plus petite de ces trois sommes, alors il existe au moins une bipartition qui sépare *x*, *y* de *z* et *t*. Plus précisément, il existe une intersection (non vide) commune entre les chemins (*x*, *z*) et (*y*, *t*) d'une part, et les chemins (*x*, *t*) et (*y*, *z*) d'autre part. Si l'on note $\delta_{xy,zt}$ la longueur de cette intersection, on a :

$$\delta_{xy} + \delta_{zt} = \delta_{xz} + \delta_{yt} - 2\delta_{xy,zt} = \delta_{xt} + \delta_{yz} - 2\delta_{xy,zt} \qquad \textbf{(2)}$$

Par exemple dans la figure 1.b, on a $\delta_{12} + \delta_{45} = \delta_{14} + \delta_{25} - 2\delta_{68} = \delta_{15} + \delta_{24} - 2\delta_{68}$.

1.2 Données moléculaires

On sait aujourd'hui que l'ADN (l'Acide DésoxyriboNucléique), est le support de l'information génétique et de la transmission héréditaire. L'ADN est donc, à l'évidence, une source privilégiée d'information pour la reconstruction phylogénétique. De manière très simplifiée, on peut considérer que l'information génétique contenue dans l'ADN est transcrite en ARN (AcideRiboNucléique) puis en une chaîne d'acides aminés qui constitue une protéine. Il est donc possible d'utiliser des séquences d'ADN, d'ARN ou d'acides aminés pour reconstruire des phylogénies.

Dans cette partie, nous introduisons la terminologie relative aux séquences moléculaires. Nous expliquons ensuite les raisons pour lesquelles ces séquences n'évoluent pas toutes à la même vitesse, et l'importance de ce phénomène en reconstruction phylogénétique. Puis, nous présentons les raisons pour lesquelles la phylogénie des séquences ne correspond pas de manière systématique à la phylogénie des espèces dont elles sont issues.

1.2.1 Définitions

L'ADN contient quatre types de bases différents. Chaque base est associée à un sucre pour former un nucléotide et ces nucléotides sont liés entre eux par des liaisons phosphates constituant ainsi une longue chaîne. Les quatre bases présentes dans l'ADN sont regroupés en deux familles distinctes : d'une part les purines : adénine (*A*) et guanine (*G*), d'autres part les pyrimidines : cytosine (*C*) et thymine (*T*). Ces bases sont complémentaires ; chaque base pyrimidique est liée à une base purique : l'adénine à la thymine et la guanine à la cytosine. L'ADN est constitué de deux chaînes polynucléotidiques complémentaires reliées par des liaisons hydrogènes. La connaissance de la séquence nucléique d'une de ces deux chaînes permet d'établir la séquence de l'autre chaîne.

L'ARN est un polynucléotide proche de l'ADN, on retrouve les deux familles de bases dans la composition de l'ARN, mais la thymine est remplacée par l'uracile (*U*) qui est une autre pyrimidine. L'ARN est obtenu à partir de l'ADN en utilisant la complémentarité des bases. La transcription de l'ADN crée une séquence d'ARN complémentaire à la séquence d'ADN initiale. Il existe plusieurs sortes d'ADN (nucléaire, mitochondrial, chloroplastique) et d'ARN (messager, de transfert, ribosaumaux). Un ARN messager, est un ARN transcrit à partir d'un gène protéique qui est destiné à être traduit en protéine. Lorsque les parties non codantes d'un ARN messager sont enlevées par l'épissage on parle d'ARN messager *mature*.

Les acides aminés sont les constituants de base des protéines ; chaque protéine est définie par une séquence particulière d'acides aminés. Il existe 20 acides aminés, chacun étant codé par un mot de trois nucléotides (codons). Il existe au moins un codon pour chaque acide aminé, mais plusieurs codons peuvent représenter le même acide aminé, et certains codons ne représentent aucun acide aminé (codons STOP). Ce code génétique, qui permet de traduire les codons en acides aminés, est universel.

1.2.2 Vitesses d'évolution des séquences

Pour pouvoir reconstruire la phylogénie d'un ensemble de séquences moléculaires, il faut que ces séquences soient homologues, c'est-à-dire, issues d'une même séquence ancestrale. Lorsque l'on étudie un ensemble de séquences homologues, l'importance de la similitude entre ces séquences est un facteur important qui conditionne la qualité de la reconstruction phylogénétique. En effet, il est impossible de reconstruire la phylogénie de ces séquences, si elles sont toutes semblables ou si elles ne partagent aucun point commun. Or cette similitude dépend à la fois du temps depuis lequel les séquences ont divergé et de la vitesse

d'évolution des séquences. On ne peut pas modifier l'époque à laquelle les événements évolutifs ont eu lieu, par contre on peut choisir des séquences qui évoluent plus ou moins rapidement suivant que l'on cherche à reconstruire une phylogénie portant sur des événements récents ou anciens.

La vitesse d'évolution correspond au pourcentage d'éléments de la séquence qui sont modifiés en une unité de temps. Pour qu'une modification apparaisse sur une séquence et soit transmise, il faut que cette transformation n'empêche pas le bon fonctionnement de l'organisme. Plus une séquence peut subir de modifications neutres plus cette séquence va pouvoir évoluer rapidement. Il est facile de voir qu'en fonction du type de données moléculaires que l'on considère, le nombre d'événements neutres possibles, et donc la vitesse d'évolution, varie. Par exemple, des modifications de l'ARN messager n'affectent pas forcément l'ARN messager mature, puisqu'elles peuvent être éliminées lors de l'épissage. De même, des modifications de l'ARN messager mature n'affectent pas forcément la séquence d'acides aminés qu'il engendre, puisque plusieurs codons représentent le même nucléotide.

Pour des raisons similaires à celles évoquées ci-dessus, la vitesse d'évolution d'une séquence dépend également de "l'importance de ce qu'elle code", c'est-à-dire des contraintes fonctionnelles qui pèsent sur elle. Il est donc également important de choisir des séquences dont les contraintes fonctionnelles correspondent à l'histoire que l'on souhaite reconstruire. Pour reconstruire la phylogénie de sous-populations d'une même espèce, on peut, par exemple, étudier les séquences de pseudo-gènes (*i.e.*, des séquences d'ADN ressemblant à des gènes mais qui ne sont pas transcrites et qui donc mutent rapidement). Pour étudier des histoires plus anciennes on peut, par exemple, considérer les séquences codant le cytochrome C, impliqué dans la respiration cellulaire (ces séquences sont donc fortement contraintes et évoluent lentement).

La vitesse d'évolution des séquences étant liée aux contraintes fonctionnelles qui portent sur elles, il est raisonnable de penser que des séquences homologues, qui subissent des contraintes fonctionnelles comparables, évoluent approximativement à la même vitesse et que cette vitesse est constante au cours du temps. Cette hypothèse, dite de l'horloge moléculaire, est basée sur les observations du taux d'évolution des protéines (notamment les α-protéines) faites par Zuckerkandl et Pauling (1962). En acceptant cette hypothèse, il est possible de dater la divergence entre deux espèces en divisant la quantité d'événements évolutifs qui séparent leurs séquences moléculaires par la vitesse d'évolution de ces séquences. En effet, sous cette hypothèse, la vitesse d'évolution de ces séquences, peut être estimée à partir de séquences homologues présentes chez des espèces pour lesquelles la date de divergence est déjà connue (par des données fossiles par exemple).

On sait aujourd'hui que cette "horloge moléculaire" est rarement fiable. Les mutations ne se produisent pas de manière indépendante, créant ainsi des épisodes d'accumulation de mutations suivis d'arrêts évolutifs qui font varier la vitesse d'évolution au cours du temps

(Gillespie 1984). Cette hypothèse de l'horloge moléculaire reste cependant un outil efficace pour certaines études phylogénétiques.

Sous l'hypothèse de l'horloge moléculaire, la racine d'une phylogénie valuée doit être équidistante de toutes ses feuilles. Cela fournit un critère permettant d'enraciner une telle phylogénie. Une autre manière d'enraciner une phylogénie consiste à ajouter une séquence dont on sait que sa divergence par rapport aux autres séquences est antérieure aux divergences propres au groupe étudié. La racine de la phylogénie initiale correspond alors à l'endroit où cette dernière séquence se greffe. Cette méthode d'enracinement par "*outgroup*" est applicable, que l'hypothèse de l'horloge moléculaire soit ou non vérifiée. La phylogénie n'est généralement pas directement reconstruite en incluant la séquence d'outgroup, car la présence d'une séquence très différente des autres peut fausser la reconstruction.

1.2.3 Phylogénies de séquences et phylogénies de taxons

A travers l'histoire des séquences moléculaires, on s'intéresse généralement à l'histoire des organismes d'où sont issues ces séquences, et à travers l'histoire de ces organismes c'est

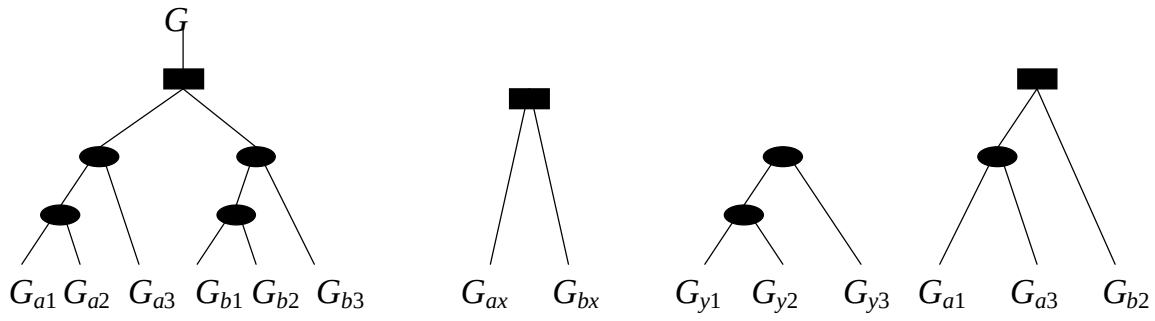


Figure 2 : phylogénie et duplication de gènes

l'histoire des "groupes" auxquels ces organismes appartiennent que l'on cherche à reconstruire. Ce paragraphe présente rapidement les raisons qui font que ces deux extrapolations sont loin d'être immédiates.

Pour pouvoir reconstruire la phylogénie d'un ensemble de séquences moléculaires, il faut que l'histoire évolutive que l'on cherche existe réellement, et donc que ces séquences soient issues d'une même séquence ancestrale ; on dit qu'elles sont homologues. La phylogénie d'un ensemble de séquences moléculaires peut refléter l'histoire des gènes qui contiennent ces séquences ou celle des êtres vivants qui possèdent ces gènes. Ces deux histoires ne sont pas forcément identiques (par exemple dans le cas de duplications

géniques) il faut donc s'assurer que les séquences que l'on utilise permettent bien de reconstruire l'histoire à laquelle on s'intéresse. Au cours de leur évolution les gènes peuvent être modifiés localement ou dupliqués. La Figure 2 illustre les problèmes liés à la duplication génique en phylogénie moléculaire, au travers de l'évolution d'un gène *G*. Dans cet exemple, une duplication du gène a produit deux copies *Ga* et *Gb* qui sont présentes dans l'espèce *E*. Cette espèce a ensuite engendré trois espèces différentes E_1, E_2, et E_3 qui contiennent chacune leur propre version de *Ga* et *Gb*. La Figure 2.a représente la phylogénie des six versions du gène ancestral *G* obtenue à la suite de cette duplication suivie de deux spéciations.

On peut obtenir la phylogénie des duplications de *G* (Figure 2.b) en considérant les différentes versions de *G* présentes dans une même espèce *x*, et la phylogénie des espèces en considérant les versions de *G* issues d'une même duplication *y* (Figure 2.c). Par contre, une phylogénie reconstruite à partir de trois versions de *G* issues de plusieurs espèces et de duplications différentes n'a pas de sens biologique (Figure 2.d). Pour reconstruire la phylogénie d'un ensemble d'espèces on considère des gènes orthologues, c'est-à-dire des gènes homologues dont l'histoire ne contient pas de duplication.

Les phénomènes de duplication ne sont pas les seuls à compliquer l'analyse phylogénétique. Il arrive, par exemple, que des espèces échangent une partie de leur matériel génétique, on parle alors de transferts horizontaux. Lors de la reconstruction phylogénétique, on suppose que la similitude entre les séquences est due au fait qu'elles possèdent un ancêtre commun proche, or les transferts horizontaux augmentent la similitude entre des séquences qui n'ont pas forcément de parents communs proches. Les transferts horizontaux peuvent jouer un rôle considérable dans l'évolution des séquences et fausser complètement les reconstructions phylogénétiques issues d'analyses de séquences moléculaires.

Comme nous venons de le voir, il existe plusieurs phénomènes biologiques qui peuvent perturber la reconstruction phylogénétique, et à cause desquels la phylogénie reconstruite pour les séquences ne reflète pas forcément la phylogénie des organismes contenant ce patrimoine génétique. De plus on ne s'intéresse généralement pas à l'histoire de ces organismes particuliers mais plutôt à l'histoire des taxons auxquels ils appartiennent. Un taxon est un ensemble d'organismes dont l'unité est admise et qui a des relations de parenté avec d'autres ensembles comparables. On parle de taxon pour désigner indifféremment les souris, les caniches ou une souche particulière d'un virus. Chaque organisme ayant un patrimoine génétique qui lui est propre, les phylogénies reconstruites dépendent en partie des représentants que l'on utilise pour chaque taxon (Philippe et Douzery 1994).

Pour reconstruire une phylogénie à partir de séquences moléculaires, il est nécessaire de choisir avec soin ces séquences. C'est une étape fondamentale qui ne peut être effectuée

correctement qu'au cas par cas et avec de solides connaissances en biologie moléculaire et en taxonomie.

1.2.4 Alignement des séquences

A partir des années 70, les techniques de séquençage (permettant d'obtenir ces données moléculaires) ont très rapidement évolué. Les études ont d'abord concerné des fragments d'ADN puis des gènes entiers et enfin l'intégralité d'un chromosome. Le séquençage de l'ADN a notamment beaucoup progressé grâce à la technique "d'amplification en chaîne par polymérase" (Mullis et Faloona 1987). Il est maintenant devenu plus facile d'obtenir des séquences d'ADN correspondant aux taxons que l'on souhaite étudier, soit par séquençage, soit en utilisant les banques de données qui mettent gratuitement à disposition les résultats de très nombreux séquençages déjà effectués.

De même qu'il faut s'assurer que les séquences étudiées sont orthologues, il faut aussi s'assurer que les nucléotides ayant la même position dans les différentes séquences sont issus d'un même nucléotide ancestral. Cette étape d'alignement des séquences permet par la suite de comparer des nucléotides qui sont effectivement comparables. Aligner les séquences est une étape difficile. En effet, au cours de l'évolution, des nucléotides ont pu être modifiés ou même supprimés tandis que d'autres ont pu être insérés au sein de la séquence. Prenons l'exemple donné dans (Caraux et al. 1995), où on dispose des trois séquences suivantes :

$$S_1 = AGAATAGCCA$$

$$S_2 = AGGATAGGA$$

$$S_3 = AGTATGGA$$

Un alignement possible de ces séquences est :

	1	2	3	4	5	6	7	8	9	10
S_1	*A*	*G*	*A*	*A*	*T*	*A*	*G*	*C*	*C*	*A*
S_2	*A*	*G*	*G*	*A*	*T*	*A*	*G*	*G*	•	*A*
S_3	*A*	*G*	*T*	*A*	*T*	•	*G*	*G*	•	*A*

Cet alignement peut s'expliquer de la manière suivante : aux positions 1, 2, 4, 5, 7 et 10 il n'y a eu aucun événement mutationel ; il y a eu deux substitutions à la position 3 ; il y a eu une suppression à la position 6 dans la séquence S_3 ; en position 8 une substitution dans la séquence S_1 ; en position 9 une insertion dans la séquence S_1. Il existe d'autres explications possibles de cet alignement. On peut par exemple considérer qu'il y a eu deux suppressions (dans les séquences S_2 et S_3) à la position 9 au lieu d'une insertion (dans la séquence S_1). Mais la première interprétation est celle qui s'explique par un nombre minimum

d'événements évolutifs, c'est donc en ce sens l'explication la plus parcimonieuse, et celle que l'on retiendra. De la même manière, parmi tous les alignements existants, on tend toujours à conserver le plus parcimonieux.

D'un point de vue informatique, ce problème s'apparente à la recherche de la "distance d'édition" entre les séquences. Dans le cas de la distance d'édition, on cherche l'alignement qui minimise le nombre d'événements mutationels (nombre de substitutions + nombre de délétions + nombre d'insertions). En biologie, la probabilité d'une substitution est supérieure à celle d'une délétion suivie d'une insertion. Cela est pris en compte lors de l'alignement en attribuant des coûts distincts aux différents événements mutationels. Dans tous les cas, on sait résoudre le problème grâce à des techniques de programmation dynamique en $O(l^2)$, où l est la longueur de la plus longue des deux séquences. Lorsque l'on souhaite reconstruire la phylogénie de n séquences nucléotidiques, la première étape consiste à aligner ces n séquences. Trouver le meilleur alignement pour n séquences est un problème difficile dont la complexité en temps est en $O(n^l)$. De plus, les biologistes ne cherchent pas exactement l'alignement de coût le plus faible. Leur objectif est avant tout de mettre en correspondance des positions pour lesquelles ils sont sûrs qu'il y a homologie. Dans les régions stables des séquences, *i.e.* les régions dans lesquelles on peut proposer un alignement sans insertion ni délétion, il est très probable que les nucléotides, alignés sur un même site, sont effectivement homologues. Les biologistes utilisent donc généralement un algorithme heuristique afin d'obtenir un premier alignement raisonnable qui privilégie les grandes régions stables. Puis, ils affinent cet alignement à la main en se servant de logiciels interactifs. Finalement, seules les positions pour lesquelles il n'y a eu ni insertion ni délétion sont conservées pour la reconstruction phylogénétique, car l'alignement est plus fiable pour ces positions et il est difficile de prendre en compte correctement les événements évolutifs d'insertions et de délétions au cours de la reconstruction phylogénétique.

L'alignement des séquences définit la correspondance entre leurs nucléotides. Les séquences alignées constituent les données de la reconstruction phylogénétique. L'alignement est donc une étape cruciale qui conditionne la qualité de la reconstruction phylogénétique. Dans le cadre de cette thèse, nous ne nous intéressons qu'à la phase de reconstruction de la phylogénie et non aux étapes préalables qui permettent d'obtenir les données qui sont utilisées lors de cette reconstruction. Nous supposerons donc que nous disposons des séquences alignées, sans événement de type insertion ou délétion. Par exemple, dans le cas des trois séquences S_1, S_2 et S_3 décrites ci-dessus nous supposerons que nous disposons de séquences alignées et pour lesquelles les positions 6 et 9 de l'alignement ont été supprimées soit :

$$S_1 = AGAATGCA$$

$$S_2 = AGGATGGA$$

$$S_3 = AGTATGGA$$

Ce jeu de données est constitué de trois séquences de même longueur, les positions homologues entre les diverses séquences sont placées sur une même colonne. Une colonne du jeu de séquences alignées est appelée site. Le numéro *s* d'une colonne identifie un site particulier que l'on appelle "site *s*", dans cet exemple il y a 8 sites que l'on numérote de 1 à 8.

1.3 Modèles de l'évolution moléculaire

Comme nous le verrons dans le chapitre suivant, ces modèles jouent un rôle central en reconstruction phylogénétique. En effet, ils sont utilisés par de nombreuses méthodes de reconstruction phylogénétique et permettent de simuler des jeux de données qui sont utilisés pour comparer les performances de ces différentes méthodes.

Dans un premier temps, nous détaillons les modélisations de séquences non-codantes. Nous commençons par présenter les hypothèses générales sur lesquelles reposent les modélisations. Nous décrivons ensuite les modèles les plus fréquemment utilisés. Dans un second temps, nous évoquons le cas des séquences codantes.

1.3.1 Hypothèses sous-jacentes

La modélisation de l'évolution moléculaire consiste à exprimer en terme probabiliste la transformation d'une séquence. L'évolution d'une séquence le long d'une branche (n_1, n_2) transforme la séquence S_1, présente initialement sur le nœud père n_1, en une séquence S_2, présente sur le nœud fils n_2. Le modèle d'évolution décrit, pour chaque site, les probabilités de passage d'un état initial au nœud père, à un état final au nœud fils. Les différents états possibles d'un site sont les quatre nucléotides *A*, *C*, *G* et *T* qui composent une séquence d'ADN alignée. En effet, aucun des modèles couramment employés ne traite les insertions et les délétions, c'est une des hypothèses faites par les modèles d'évolution. La plupart des modèles couramment utilisés en phylogénie font les hypothèses suivantes :

-H0 : les séquences évoluent exclusivement par le mécanisme de substitution nucléotidique.

-H1 : les sites sont indépendants : les modifications qui affectent un site au cours du temps, ne sont affectées ni par l'état du reste de la séquence ni par les modifications qui affectent les autres sites. Sous cette hypothèse, il suffit de connaître le processus d'évolution des sites pour connaître celui des séquences.

-H2 : les sites sont identiquement distribués : le processus d'évolution est le même pour tous les sites et ne dépend donc pas de la position du site dans la séquence. Le modèle d'évolution et ses paramètres sont donc les mêmes pour tous les sites.

-H3 : le processus d'évolution est markovien ou "sans mémoire" : l'évolution d'une séquence ne dépend que de l'état actuel de cette séquence et non du processus évolutif qui a conduit à cette séquence.

-H4 : le processus d'évolution est homogène : il est le même pour toutes les branches de la phylogénie, le processus reste le même au cours du temps.

-H5 : le processus d'évolution est stationnaire : la probabilité d'observer une base *b* particulière ne dépend pas du moment de l'observation, et cette probabilité est notée π_b. Cette probabilité est donc la même pour toutes les séquences considérées.

-H6 : il y a au plus une mutation par unité de temps : sur un intervalle de temps infime *dt* il ne peut pas se produire plus d'une mutation. La probabilité de substitution d'une base *b* en une base *c* sur cet intervalle de temps *dt* est noté $M_{bc}dt$, et pour toutes bases *b* on a $M_{bA} + M_{bC} + M_{bT} + M_{bG} = 1$. Cette équation permet de déterminer une des valeurs M_{bc} en fonction des trois autres. On peut, par exemple, pour chaque nucléotide *b*, exprimer M_{bb} en fonction des trois autres taux de substitution de la base *b*. Le modèle ainsi obtenu à donc 12 paramètres libres. Pour un tel modèle, on peut ré-écrire le taux M_{bc} de substitution d'une base *b* en une base *c* différente sous la forme du produit $\pi_c R_{bc}$ avec $R_{bc} = M_{bc} / \pi_c$. Cette formulation permet de mettre en évidence les hypothèses qui sont faites par certains modèles sur les fréquences des différentes bases. On utilise souvent une représentation matricielle pour représenter l'ensemble de ces taux de substitution. Plutôt que d'utiliser directement la matrice markovienne *M*, les modèles sont souvent décrits à partir de la "matrice des taux instantanés", que l'on note *Q* et qui correspond simplement à *M-I* (avec *I* la matrice identité de dimension 4).

Sous ces hypothèses, la forme générale de la matrice *Q* est la suivante :

$$Q = \begin{pmatrix} \lambda_A & \pi_C R_{AC} & \pi_G R_{AG} & \pi_T R_{AT} \\ \pi_A R_{CA} & \lambda_C & \pi_G R_{CG} & \pi_T R_{CT} \\ \pi_A R_{GA} & \pi_C R_{GC} & \lambda_G & \pi_T R_{GT} \\ \pi_A R_{TA} & \pi_C R_{TC} & \pi_G R_{TG} & \lambda_T \end{pmatrix} \quad \textbf{(3)}$$

où les λ_X sont tels que la somme de chaque ligne vaut 0. On a donc, par exemple, $\lambda_A = -(\pi_C R_{AC} + \pi_G R_{AG} + \pi_T R_{AT})$.

La matrice des taux instantanés *Q*, fournit les taux de changement entre chaque couple de nucléotides. Chaque colonne indique les taux de changement relatifs à un nucléotide *X*, le terme λ_X représente les changements où ce nucléotide est transformé en un autre, les autres termes de la colonne représentent les changements où un autre nucléotide est transformé en *X*. Ainsi, si l'on connaît les probabilités d'apparition des quatre bases au temps *t*, on peut utiliser la matrice *Q* pour calculer ces probabilités à l'instant $(t + dt)$. En notant $X(t)$ la probabilité d'apparition de la base *X* à l'instant *t* et $X(t + dt)$ sa probabilité à l'instant $(t + dt)$ on obtient le système de quatre équations différentielles suivant :

$$\begin{cases} A(t+dt) = A(t) + \lambda_A A(t)dt + \pi_A R_{CA} C(t)dt + \pi_A R_{GA} G(t)dt + \pi_A R_{TA} T(t)dt \\ C(t+dt) = C(t) + \pi_C R_{AC} A(t)dt + \lambda_C C(t)dt + \pi_C R_{GC} G(t)dt + \pi_C R_{TC} T(t)dt \\ G(t+dt) = G(t) + \pi_G R_{AG} A(t)dt + \pi_G R_{CG} C(t)dt + \lambda_G G(t)dt + \pi_G R_{TG} T(t)dt \\ T(t+dt) = T(t) + \pi_T R_{AT} T(t)dt + \pi_T R_{CT} C(t)dt + \pi_G R_{GT} G(t)dt + \lambda_T T(t)dt \end{cases} \quad \textbf{(4)}$$

En notant $F(t) = (A(t), C(t), G(t), T(t))$ le vecteur ligne des probabilités des quatre bases à l'instant *t*, et $F(t+dt)$ celui des probabilités à l'instant $(t+dt)$, on peut écrire l'équation (4) sous la forme matricielle suivante :

$$F(t+dt) = F(t) + F(t).Q.dt \quad \textbf{(5)}$$

soit :

$$\frac{dF(t)}{dt} = F(t).Q$$

Ce système se résout de manière classique et sa solution s'écrit (Cox et Miller 1977 ; Yang 1994) :

$$F(t) = F(0).e^{Qt}$$

où $F(0)$ désigne le vecteur vertical des probabilités ancestrales et e^{Qt} l'exponentielle matricielle de la matrice des taux. Par définition, l'exponentielle matricielle vaut :

$$e^{Qt} = \sum_{k=0}^{\infty} \frac{Q^k t^k}{k!} \quad \textbf{(6)}$$

et se calcule facilement en diagonalisant la matrice *Q*, c'est-à-dire en l'exprimant sous la forme d'un produit de trois matrices tel que : $Q = PDP^{-1}$ où *D* est une matrice diagonale et P^{-1} est la matrice inverse de *P*. En utilisant cette expression de la matrice *Q*, l'équation (6) devient :

$$e^{Qt} = \sum_{k=0}^{\infty} \frac{\left(PDP^{-1}\right)^k t^k}{k!} = \sum_{k=0}^{\infty} \left(\frac{PD^k t^k P^{-1}}{k!} t\right) = P\left(\sum_{k=0}^{\infty} \frac{D^k t^k}{k!}\right) P^{-1} = Pe^{Dt}P^{-1} \quad \textbf{(7)}$$

Il est facile de vérifier que l'exponentielle d'une matrice diagonale *D* correspond à la matrice obtenue en prenant l'exponentielle de ses termes diagonaux. La formule (7) permet donc de calculer simplement e^{Qt}. Ainsi, à partir des taux de substitution instantanés, il est possible d'obtenir les probabilités $P_{bc}(t)$ de changement d'un nucléotide *b* en un nucléotide *c* en un temps *t*. Ces probabilités définissent la matrice de probabilité $P(t) = e^{Qt}$ dont la forme générale est la suivante :

$$P(t) = \begin{pmatrix} 1-P_A(t) & P_{AC}(t) & P_{AG}(t) & P_{AT}(t) \\ P_{CA}(t) & 1-P_C(t) & P_{CG}(t) & P_{CT}(t) \\ P_{GA}(t) & P_{GC}(t) & 1-P_G(t) & P_{GC}(t) \\ P_{TA}(t) & P_{TC}(t) & P_{TG}(t) & 1-P_T(t) \end{pmatrix}$$

1.3.2 Pincipaux modèles d'évolution

Un grand nombre des modèles existants sont des cas particuliers de la matrice *Q* de l'équation (3). Ils sont obtenus en ajoutant des contraintes supplémentaires au modèle d'évolution représenté par cette matrice. En particulier, il font tous l'hypothèse suivante :

$$\forall b,c \in \{A,C,G,T\},\ R_{bc} = R_{cb}$$

Le modèle obtenu en ajoutant cette hypothèse est le modèle GTR (*general time reversible model*), qui est utilisé par Lanave, Preparata et al. (1984), décrit dans (Rodriguez et al. 1990) et dont la matrice des taux est la suivante :

$$Q = \begin{pmatrix} \lambda_A & \pi_C R_{AC} & \pi_G R_{AG} & \pi_T R_{AT} \\ \pi_A R_{AC} & \lambda_C & \pi_G R_{CG} & \pi_T R_{CT} \\ \pi_A R_{AG} & \pi_C R_{CG} & \lambda_G & \pi_T R_{GT} \\ \pi_A R_{AT} & \pi_C R_{CT} & \pi_G R_{GT} & \lambda_T \end{pmatrix} \quad \textbf{(8)}$$

où les λ_X sont tels que la somme de chaque ligne est nulle. Cette matrice n'est pas nécessairement symétrique, le modèle GTR ne suit donc pas un forcément un processus markovien réversible. Pour ce modèle, la probabilité *P* d'observer un nucléotide *b* qui évolue en une base *c* au cours d'un temps *dt*, est la même que la probabilité P_{rev} d'observer un nucléotide *c* qui évolue en une base *b* au cours d'un temps *dt*. Cette propriété est triviale dans le cas où b et c sont en fait la même base. Dans les autres cas, la probabilité a priori d'observer le nucléotide *b* est π_b, la probabilité qu'un nucléotide *b* se transforme en un nucléotide *c* en un temps *dt* vaut $\pi_c R_{bc}$; on a donc : $P = \pi_b.(\pi_c R_{bc})$. On obtient de même $P_{rev} = \pi_c(\pi_b R_{cb})$. Pour le modèle GTR, $R_{bc} = R_{cb}$ et donc $P = P_{rev}$. C'est en ce sens que le modèle GTR est dit réversible, même s'il ne correspond pas à un modèle markovien réversible au sens usuel.

Pour définir complètement le modèle GTR, il faut fixer les valeurs des 6 paramètres de type R_{XY} et des 3 paramètres de type π_X (le quatrième est tel que la somme de ces quatre paramètres vaille 1). Le modèle GTR est donc un modèle à 9 paramètres libres. Les autres modèles couramment utilisés en phylogénie sont des cas particuliers du modèle GTR, obtenus en réduisant le nombre de paramètres libres de la matrice (8). Par exemple, si l'on suppose que la probabilité à priori d'observer une base est la même pour toutes les bases

($\pi_A = \pi_C = \pi_G = \pi_T = 0.25$) et que pour deux bases distinctes b et c on a toujours $R_{bc} = 4\alpha$, alors on retrouve le modèle de Jukes et Cantor (1969) à 1 paramètre :

$$Q = \begin{pmatrix} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha \end{pmatrix} \quad \textbf{(9)}$$

A partir de cette matrice des taux on peut calculer la matrice des probabilités du modèle de Jukes et Cantor, grâce à l'équation (3) page 22, et l'on obtient la matrice $P(t)$ définie par les équations suivantes :

$$P_{bc}(t) = \begin{cases} \frac{1}{4} - \frac{1}{4}e^{-4\alpha t} & (b \neq c) \\ \frac{1}{4} + \frac{3}{4}e^{-4\alpha t} & (b = c) \end{cases} \quad \textbf{(10)}$$

Le modèle à deux paramètres de Kimura noté K2P (Kimura 1980), suppose lui aussi que les fréquences à priori sont toutes les mêmes, mais il distingue deux types de substitutions, les transitions et les transvertions. Rappelons que les nucléotides se regroupent en deux grandes familles : les purines (*A* et *G*), et les pyrimidines (*C* et *T*). Une transition transforme un nucléotide d'une famille en un nucléotide de la même famille ($A \leftrightarrow G$ et $C \leftrightarrow T$) alors qu'une transvertion transforme un nucléotide d'une famille en un nucléotide de l'autre famille ($A \leftrightarrow C$, $A \leftrightarrow T$, $G \leftrightarrow C$ et $G \leftrightarrow T$). En notant α le taux de transition, et β le taux de transvertion, la matrice des taux pour ce modèle est :

$$Q = \begin{pmatrix} -(\alpha+2\beta) & \beta & \alpha & \beta \\ \beta & -(\alpha+2\beta) & \beta & \alpha \\ \alpha & \beta & -(\alpha+2\beta) & \beta \\ \beta & \alpha & \beta & -(\alpha+2\beta) \end{pmatrix} \quad \textbf{(11)}$$

Et la matrice des probabilités de ce modèle est défini par les équations suivantes :

$$P_{bc}(t) = \begin{cases} \frac{1}{4} - \frac{1}{4}e^{-4\beta t} & (b \neq c, \text{ transversion}) \\ \frac{1}{4} - \frac{1}{2}e^{-2(\alpha+\beta)t} + \frac{1}{4}e^{-4\beta t} & (b \neq c, \text{ transition}) \\ \frac{1}{4} + \frac{1}{2}e^{-2(\alpha+\beta)t} + \frac{1}{4}e^{-4\beta t} & (b = c) \end{cases}$$

La vitesse d'évolution correspond au nombre de changements d'état d'un nucléotide en un temps élémentaire *dt*. Pour le modèle K2P, cette vitesse vaut donc $(\alpha + 2\beta)$. Le produit de

cette vitesse par un temps *t*, définit la distance entre deux séquences. En notant δ cette distance, on a donc $\delta = (\alpha + 2\beta)t$. Cette distance correspond au nombre moyen d'événements de type changement d'état qui se sont produits sur chaque site pour qu'une séquence se transforme en une autre. Si l'on note *K* le rapport α / β, on a alors :

$$P_{bc}(\delta) = \begin{cases} \frac{1}{4} - \frac{1}{4}e^{-4\delta/(2+K)} & (b \neq c, \text{ transversion}) \\ \frac{1}{4} - \frac{1}{2}e^{-2\delta(K+1)/(2+K)} + \frac{1}{4}e^{-4\delta/(2+K)} & (b \neq c, \text{ transition}) \\ \frac{1}{4} + \frac{1}{2}e^{-2\delta(K+1)/(2+K)} + \frac{1}{4}e^{-4\delta/(2+K)} & (b = c) \end{cases} \quad \textbf{(12)}$$

La Figure 3 page 27, issue de (Swofford et al. 1996) résume les relations qui existent entre les différents modèles d'évolution issus du modèle GTR.

Les modèles d'évolution décrits ci-dessus font l'hypothèse que tous les sites évoluent à la même vitesse. Plusieurs modèles discrets ou continus ont été proposés pour décrire la variabilité des vitesses d'évolution entre sites. Ces modèles font généralement l'hypothèse que la vitesse d'un site est la même dans tout l'arbre (Yang 1993).

La matrice des taux indique la vitesse d'évolution. Des sites qui évoluent suivant un même modèle mais à des vitesses différentes ont donc les mêmes matrices de substitution à un facteur multiplicatif près. En notant λ_s le facteur multiplicatif associé au site *s*, les probabilités $F_s(d + dt)$ d'observer les différentes bases à l'instant $(t + dt)$ se déduisent simplement des probabilités $F_s(t)$ d'observer ces bases à l'instant *t* à partir de l'équation (5) qui devient :

$$F_s(t + dt) = F_s(t) + F_s(t).(\lambda_s.Q).dt$$

On utilise généralement une loi gamma pour modéliser la distribution des vitesses d'évolution parmi les sites. L'intérêt de cette loi est qu'elle peut prendre une forme exponentielle ou gaussienne suivant la valeur de son paramètre de forme α. La forme exponentielle ($\alpha \leq 1$) indique que la plupart des sites évoluent à un taux faible mais que quelques sites peuvent évoluer à un taux très rapide, la variance des vitesses est alors élevée. La forme gaussienne ($\alpha >> 1$) traduit le fait que la majorité des sites évolue à un taux proche de la valeur moyenne, la variance des vitesses est alors faible (Yang 1996).

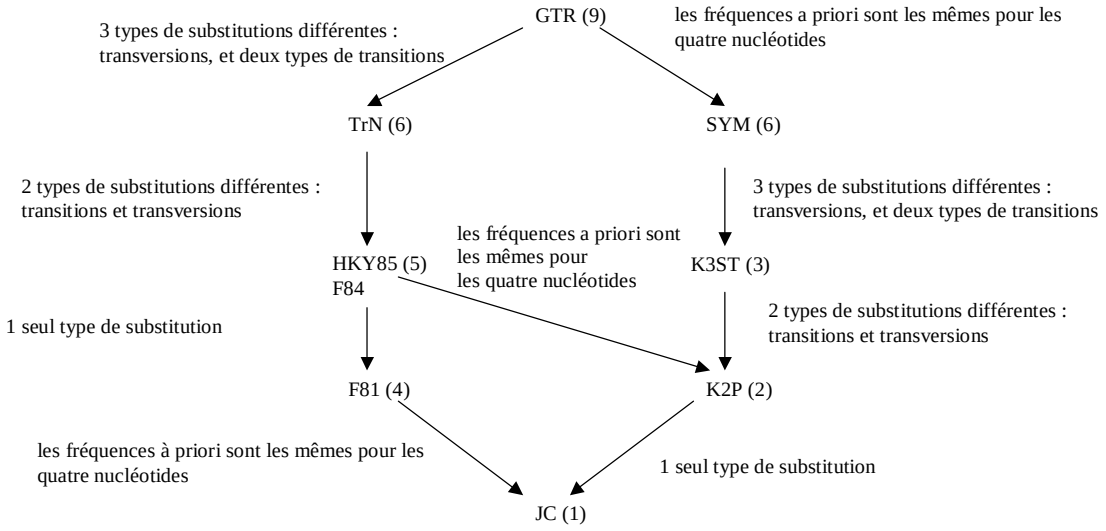


Figure 3 : principaux modèles d'évolution

Chaque flèche indique les restrictions nécessaires pour passer d'un modèle à un cas particulier de ce modèle. Pour chaque modèle, le nombre de ses paramètres et indiqué entre parenthèse. Les abréviations utilisées pour représenter les différents modèles sont les suivantes :

GTR est le "general time-reversible model" (Lanave et al. 1984)
TrN est le modèle de Tamura et Nei (1993)
SYM est le modèle décrit par Zharkikh et Li (1995)
F84 est le modèle utilisé dans le programme PHYLIP de Felsenstein (1993)
HKY85 est le modèle décrit par Hasegawa, Kishino et Yano (1985)
K3ST est un modèle qui distingue trois types de substitutions (Kimura 1981)
K2P distingue seulement deux types de substitutions (Kimura 1980)
F81 est un modèle décrit par Felsenstein (1981) qui est équivalent à celui décrit par Tajima et Nei (1982)
JC est le modèle introduit par Jukes et Cantor (1969)

1.3.3 Modélisation des séquences codantes

Les modèles décrits ci-dessus sont destinés à représenter l'évolution de séquences d'ADN ne codant pas pour des protéines. Pour les séquences codantes, les différents types de substitutions ont des conséquences variables sur la protéine codée. Certaines substitutions ne changent pas l'acide nucléique codé, d'autres entraînent le remplacement d'un acide aminé par un autre qui est proche sur le plan biochimique, d'autres enfin entraînent un changement radical.

Une première manière de prendre ces informations en compte consiste à utiliser un modèle de substitution entre les 20 acides aminés existants ou entre les 61 codons qui représentent un acide aminé. Ces modèles sont très riches en paramètres, puisque leurs matrices de substitution sont respectivement de dimension 20 et 61. De plus, comme chaque acide aminé (ou codon) est représenté par trois nucléotides, on dispose de moins d'informations pour ajuster les paramètres de ces modèles que pour ajuster ceux correspondant à des séquences non codantes. Généralement, les paramètres de ces modèles ne peuvent donc pas être estimés de manière correcte à partir des séquences étudiées. Les valeurs des taux de substitution sont donc en général estimées et fixées a priori en utilisant de très grands jeux de données. Cette approche a été proposée par Dayhoff, Schwartz et Orcutt (1978) (matrices PAM) et reprise notamment par Jones, Taylor et Thornton (1992) (matrice JTT).

Une seconde approche consiste à regrouper les sites en deux ou trois classes différentes en fonction de leur position au sein du codon. On applique ensuite à chacune de ces classes un modèle de substitution classique. Cette approche repose sur le fait que les trois positions d'un codon subissent des contraintes très différentes. En effet, la proportion des substitutions qui modifient la signification du codon est de 184/192 pour la première position, 190/192 pour la seconde et 64/192 pour la troisième. On voit donc que lorsque l'on traite des séquences codantes, il est important de traiter indépendamment les sites situés en troisième position.

Dans la suite de cette thèse, nous supposerons pour simplifier que nous disposons de séquences d'ADN non codantes qui sont correctement alignées, mais le propos s'applique de manière plus générale à d'autres types de séquences.

Chapitre 2 Principales méthodes de reconstruction phylogénétique

Ce chapitre décrit les principales méthodes de reconstruction phylogénétique. Dans un premier temps, nous décrivons les principes algorithmiques communs à l'ensemble de ces méthodes. Nous décrivons ensuite le principe des méthodes de distances, de parcimonie et du maximum de vraisemblance et nous détaillons certains aspects algorithmiques de ces deux dernières méthodes. Nous concluons ce chapitre sur une comparaison de ces différentes approches qui met en évidence la nécessité de développer de nouvelles méthodes ayant des performances intermédiaires entre la méthode du maximum de vraisemblance et les méthodes de distances.

2.1 Recherche de l'arbre optimum

Les méthodes de reconstruction phylogénétique reposent presque toujours sur l'optimisation d'un critère permettant de comparer les phylogénies possibles d'un ensemble de taxons. Comme nous l'avons vu au chapitre précédent, le nombre de phylogénies possibles pour un ensemble de n taxons est $3\times5\times7\times\ldots\times(2n-5)$ (équation (1) page 13), et ce nombre augmente très rapidement en fonction de n. Dès que le nombre de taxons étudiés dépasse une dizaine, il est impossible de considérer toutes les phylogénies possibles pour en trouver une qui soit optimale pour le critère considéré. Néanmoins, pour certains critères, par exemple celui optimisé par l'algorithme Q^* (Berry et Gascuel 2000), il est possible de trouver un arbre optimal en un temps polynomial. Cependant, pour la plupart des critères utilisés en reconstruction phylogénétique, la recherche d'un arbre optimal est un problème NP-difficile (Foulds et Graham 1982 ; Steel 1992). Dans ce cas, il est nécessaire d'utiliser une heuristique, de manière à proposer un arbre « satisfaisant » au sens du critère choisi, en un temps raisonnable.

Cette partie décrit les quatre approches sur lesquelles reposent la majorité des heuristiques utilisées en reconstruction phylogénétique pour chercher un arbre optimal. Les deux premières sont des processus de construction gloutons permettant d'obtenir un arbre satisfaisant au sens du critère considéré. Les deux autres approches effectuent des améliorations itératives permettant de garantir que l'arbre proposé est au moins un optimum local.

2.1.1 Processus agglomératif

Dans cette approche, on considère initialement une phylogénie complètement irrésolue qui contient un seul nœud interne relié à chacun des taxons étudiés. A chaque étape, la phylogénie courante contient un seul nœud interne u de degré supérieur à trois ; deux voisins de ce nœud particulier sont choisis pour être agglomérés. Pour réaliser cette agglomération, on ajoute un nouveau nœud interne dont les trois voisins sont les deux nœuds agglomérés et le nœud u. A chaque étape le nombre de voisins du nœud n diminue et le processus s'arrête lorsque u n'a plus que trois voisins. L'agglomération choisie est celle qui permet d'obtenir la meilleure phylogénie partiellement résolue au sens du critère

considéré. Le processus s'arrête lorsque l'on obtient une phylogénie complètement résolue ou lorsque le critère n'est amélioré par aucune des agglomérations possibles. La Figure 4 montre comment on peut obtenir la phylogénie complètement résolue de la Figure 1.b (page 12) en suivant ce processus agglomératif.

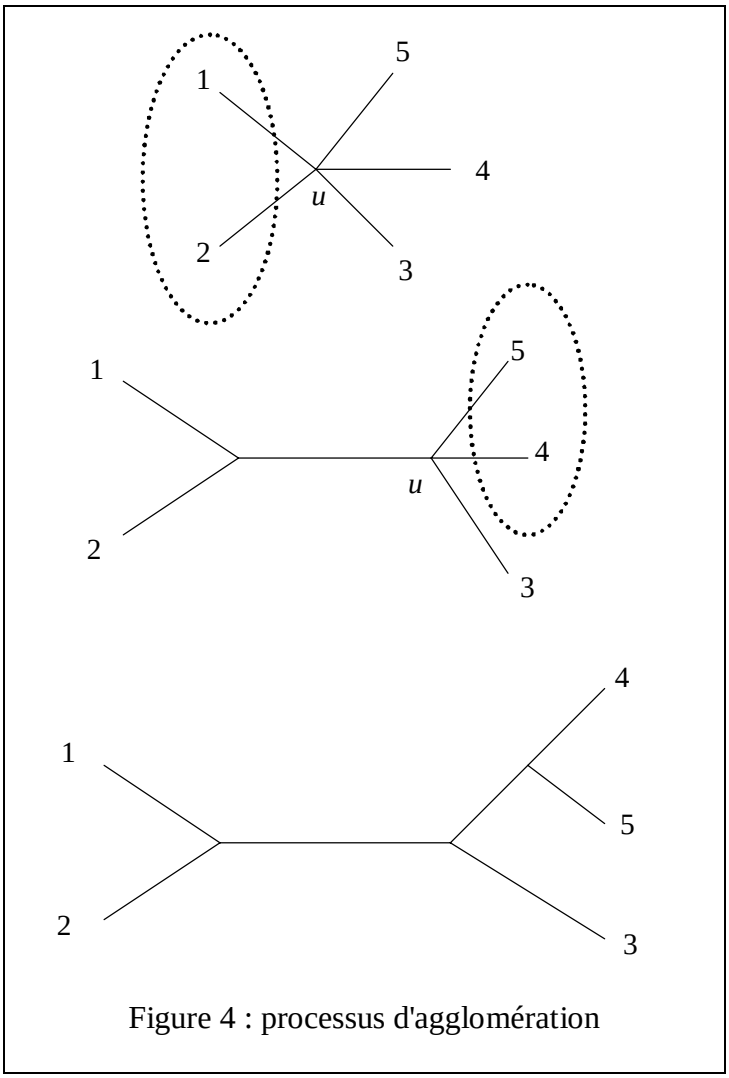


Figure 4 : processus d'agglomération

2.1.2 Processus d'insertion

Le processus d'insertion construit un arbre phylogénétique à partir d'une phylogénie à trois taxons sur laquelle sont successivement greffés les autres taxons. La branche sur laquelle le nouveau taxon est inséré est choisie de manière à ce que la phylogénie partielle ainsi obtenue soit la meilleure possible au sens du critère considéré.

La Figure 5 illustre la manière dont l'arbre de la Figure 1.b (page 12) peut être obtenu en suivant un processus d'insertion.

2.1.3 Ré-arrangement d'arbres

Lorsque l'on construit un arbre suivant un processus d'insertion ou d'agglomération, les choix effectués à une étape ne sont jamais remis en cause par la suite. Une fois qu'un taxon est inséré, il ne change plus de position par la suite. Ces approches « gloutonnes », où les choix sont définitifs, ont l'avantage d'être rapides, mais ne reconstruisent que rarement l'arbre optimum. Il est possible d'améliorer l'arbre reconstruit par un processus d'amélioration itérative en testant des ré-arrangements possibles de cet arbre. Si l'une de ces modifications de l'arbre améliore le critère que l'on cherche à optimiser, alors on conserve ce nouvel arbre. On teste ensuite les ré-arrangements possibles de ce nouvel arbre. Le processus s'arrête lorsqu'il n'est plus possible d'améliorer l'arbre en lui faisant subir un des ré-arrangements que l'on a définis comme étant possibles. Pour les méthodes utilisant ce type d'approche, le temps de calcul dépend donc notamment du nombre de ré-

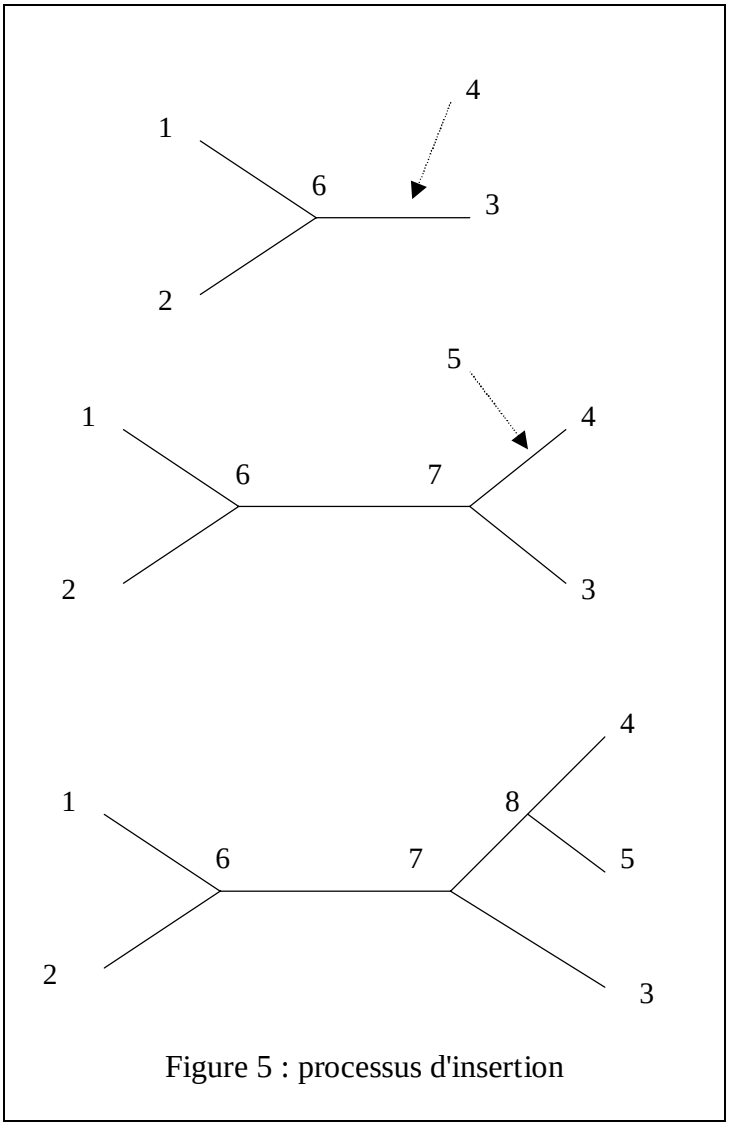


Figure 5 : processus d'insertion

arrangements effectués, ce qui complique l'analyse de leur complexité en temps de calcul.

Il existe plusieurs formes de ré-arrangements possibles. Les deux formes les plus communément utilisées en reconstruction phylogénétique sont le ré-arrangement par re-branchement de sous-arbres et celui par échange de sous-arbres voisins (Nearest-Neighbor Interchanges : NNI). Il est important de noter que plus le nombre de ré-arrangements autorisés est grand, plus on a de chances de trouver un arbre ayant une valeur de critère

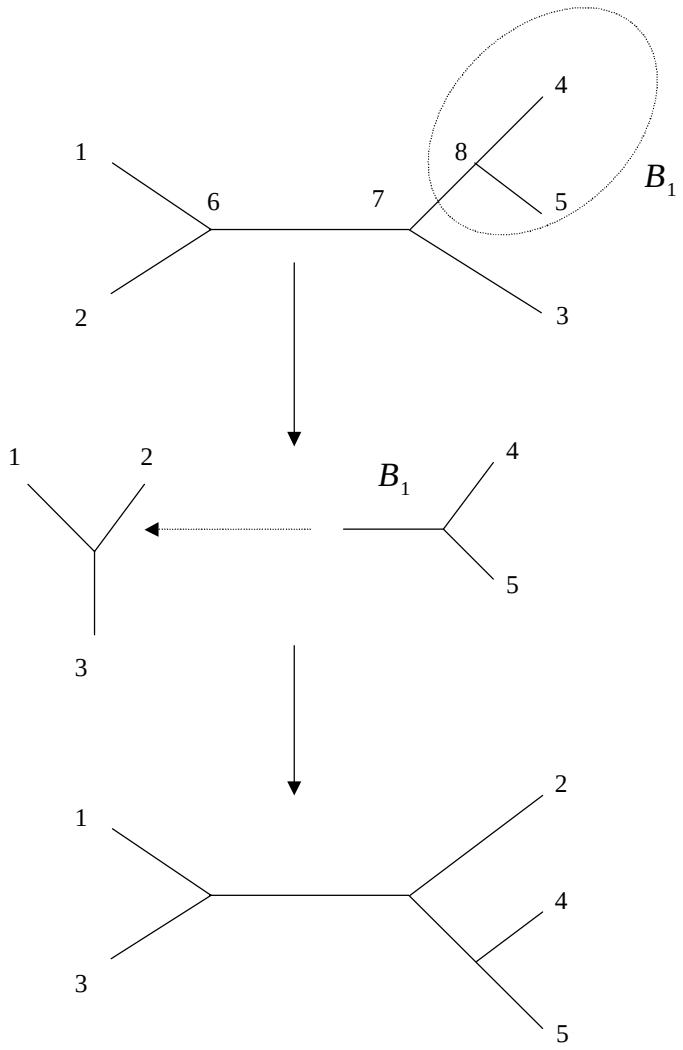


Figure 6 : réarrangement par re-branchement de sous-arbre

La première étape détache un sous-arbre de l'arbre initial *T* (le sous-arbre B_1 dans cet exemple). La seconde étape re-branche ce sous-arbre sur l'une des branches restantes de l'arbre *T* (la branche reliée au taxon 2 dans cet exemple).

élevé, mais plus le temps nécessaire pour tester ces réarrangements est élevé.

Dans le ré-arrangement par re-branchement, on considère que tout sous-arbre de *T* peut être détaché de *T*, puis greffé sur n'importe qu'elle arête de l'arbre *T* privé de ce sous-arbre (Figure 6).

Le ré-arrangement par NNI repose sur le fait qu'une branche interne d'un arbre *T* induit deux sous-arbres d'un coté (que l'on notera A_1 et A_2) et deux de l'autre (que l'on notera B_1 et B_2). Pour chaque branche de l'arbre *T*, le ré-arrangement par NNI teste l'échange de A_1 avec B_1 et l'échange de A_1 avec B_2 (les arbres obtenus par ces réarrangements sont les mêmes que ceux obtenus en échangeant A_2 avec B_2 et A_2 avec B_1) (Figure 7).

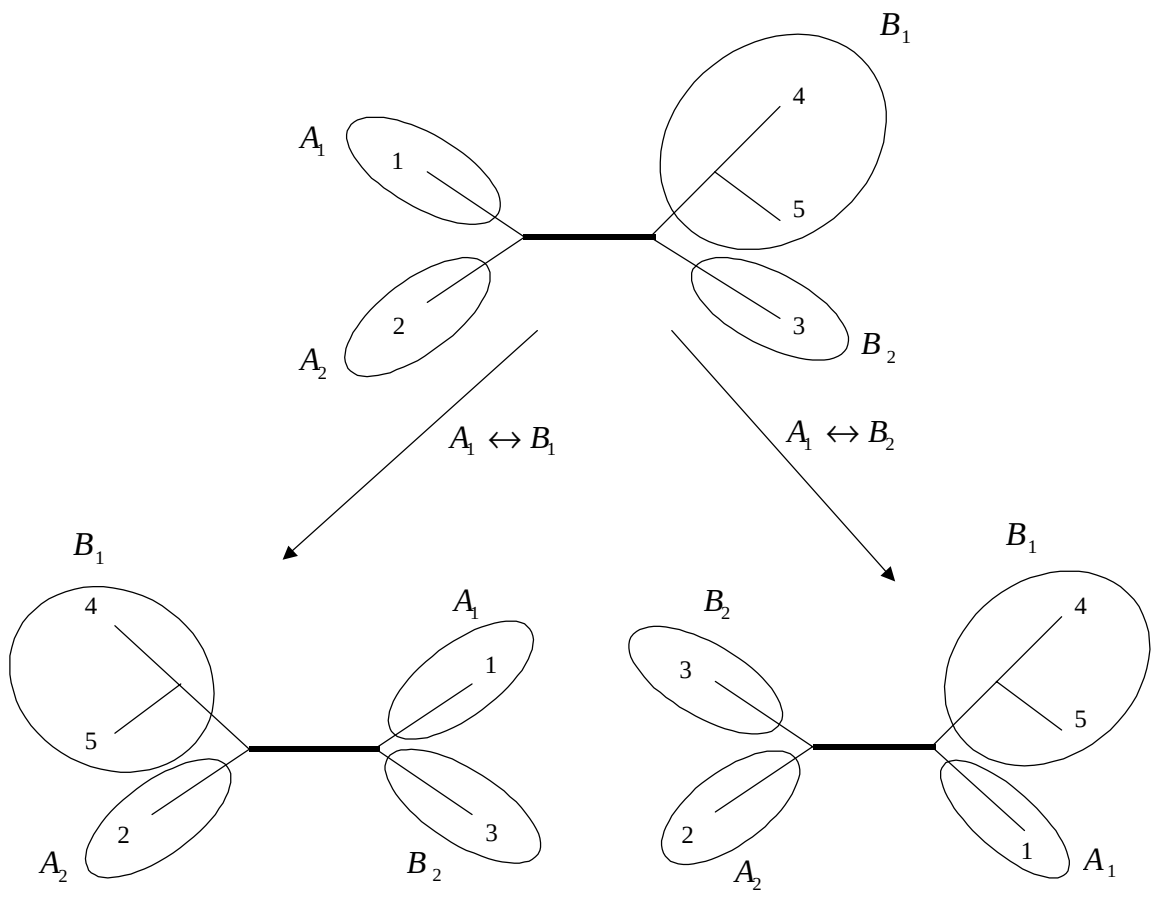


Figure 7 : réarrangement de type NNI

Les réarrangements de type NNI qui sont possibles pour la branche en gras correspondent à l'échange de A_1 avec B_1 et à l'échange de A_1 avec B_2.

Tous les arbres qui peuvent être obtenus à partir de l'arbre *T* suivant des réarrangements de type NNI, peuvent également être obtenus par des réarrangements de type re-branchement.

En effet, échanger A_1 et B_1 (respectivement B_2) est équivalent à re-brancher B_1 (respectivement B_2) sur l'arête qui relie A_1 au reste de l'arbre. Par contre, l'inverse n'est pas vrai. Par exemple, si on re-branche l'arbre A_1 sur la branche qui relie le nœud 4 au reste de l'arbre, on obtient un arbre qui ne correspond à aucun réarrangement de type NNI de l'arbre *T*.

2.2 Méthodes de distances

Les méthodes de distances reconstruisent la phylogénie d'un ensemble de taxons à partir de l'ensemble des distances évolutives δ_{ij} qui séparent chaque couple de séquences *i*, *j*.

Nous commençons par définir ce qu'est une distance évolutive. Nous insistons sur les distances correspondant aux modèles de Jukes et Cantor et de Kimura qui nous servirons à illustrer nos propos dans le quatrième chapitre. Nous décrivons ensuite les principales méthodes de distances.

2.2.1 Distances évolutives

La distance évolutive entre deux séquences est le nombre moyen de substitutions par site qui sont survenues depuis la divergence des deux séquences. En utilisant un modèle d'évolution, on peut estimer cette distance à partir des différences observées entre les deux séquences. Nous verrons au chapitre 4 (§ 4.2.4), comment cette estimation peut être obtenue en utilisant une approche de maximum de vraisemblance. Pour les modèles les plus simples, une formule analytique simple (induite par maximum de vraisemblance) permet d'obtenir directement une estimation de la distance évolutive.

Considérons par exemple le cas du modèle de Jukes et Cantor. Sa vitesse d'évolution correspond au nombre de changements d'état d'un nucléotide en un temps élémentaire *dt* et vaut donc 3α (équation (9) page 25). Suivant ce modèle, si S_1 et S_2 sont deux séquences ayant divergé depuis un temps *t* d'une même séquence ancestrale, chaque site a subi un taux de substitution de 3α entre la séquence ancestrale et chacune de ses séquences filles. La distance évolutive δ_{12}, qui sépare S_1 et S_2, peut être estimée par l'espérance du nombre de substitutions par site entre S_1 et S_2, soit $\delta_{12} = 2 \times 3\alpha t$. Suivant ce modèle, le taux $P_{\neq}(2t)$ de différence attendue entre deux séquences au bout d'un temps $2t$ vaut $3/4(1-e^{-8\alpha t})$ (équation (10) page 25). En notant *k* le nombre de différences observées entre les deux séquences, on peut utiliser k/l pour estimer $P_{\neq}(2t)$. On obtient alors l'équation suivante : $k/l = 3/4(1-e^{-8\alpha t})$. Ce qui permet d'estimer αt et d'obtenir, pour le modèle de Jukes et Cantor, l'estimateur $\delta_{12}(=6\alpha t)$ de la distance évolutive entre S_1 et S_2 :

$$\delta_{12} = \begin{cases} -\frac{3}{4}\ln(1-\frac{4k}{3l}) & \text{si } (k/l < 3/4) \\ \infty & \text{sinon} \end{cases} \qquad \textbf{(13)}$$

Pour le modèle à deux paramètres de Kimura, l'estimation de la distance correspond à $\delta_{12} = 2(\alpha + 2\beta)t$. Suivant ce modèle, le taux $P_s(2t)$ de transition attendue entre deux

séquences au bout d'un temps $2t$ vaut $(1-e^{-8\beta t})/2$ et le taux $P_v(2t)$ de transversion attendue vaut $(1-2e^{-4(\alpha+\beta)t}+e^{-8\beta t})/4$ (équation (12) page 26). En notant k_v le nombre de transversions observées entre les deux séquences, on peut utiliser k_v/l pour estimer $P_v(2t)$, on obtient alors l'équation suivante :

$$\beta t = -\frac{1}{8}\ln(1-2\frac{k_v}{l})$$

En utilisant cette estimation de βt, et en estimant $P_s(2t)$ par le taux de transition observé k_s/l, on obtient l'équation suivante :

$$\alpha t = \frac{1}{4}\ln(1-2\frac{k_s}{l}-\frac{k_v}{l})+\frac{1}{8}\ln(1-2\frac{k_v}{l})$$

Pour le modèle à deux paramètres de Kimura, l'estimateur δ_{12} de la distance évolutive entre S_1 et S_2 est donc :

$$\delta_{12} = \begin{cases} -\frac{1}{2}\ln(1-2\frac{k_s}{l}-\frac{k_v}{l})-\frac{1}{4}\ln(1-2\frac{k_v}{l}) & \text{si } (1-2\frac{k_s}{l}-\frac{k_v}{l}>0 \text{ et } 1-2\frac{k_v}{l}>0) \\ \infty & \text{sinon} \end{cases} \quad \textbf{(14)}$$

Ces deux exemples illustrent la manière dont l'estimation de la distance évolutive entre deux séquences est liée au modèle d'évolution adopté. En utilisant ces formules, on peut estimer la distance évolutive δ_{ij} qui sépare chaque couple de séquences S_i, S_j. A partir de ces données, les méthodes de distances peuvent inférer la phylogénie des séquences étudiées.

2.2.2 Méthodes agglomératives

A chaque étape, ces méthodes choisissent la paire de nœuds à agglomérer à partir d'une matrice de distances (δ_{ij}) où i (respectivement j) représente soit un des taxons étudiés soit un sous-arbre contenant plusieurs de ces taxons. Une fois que la paire {1,2} à agglomérer est sélectionnée, ces méthodes créent un nouveau nœud qui représente la racine du nouveau sous-arbre. Elles réduisent ensuite la matrice de distances en enlevant toutes les distances associées à 1 ou à 2, et en estimant les distances relatives au nouveau nœud. Ainsi après p agglomérations, la dimension r de la matrice de distances est égale à $n-p$.

Cette partie décrit les méthodes ADDTREE (Sattath et Tversky 1977), NJ (Saitou et Nei 1987), BIONJ (Gascuel 1997) et WEIGHBOR (Bruno, Socci et Halpern 2000). Ces méthodes agglomératives se différencient les unes des autres, par le critère qu'elles utilisent pour choisir la paire de nœuds à agglomérer et par la manière dont elles estiment les nouvelles distances.

2.2.2.1 ADDTREE

Le critère d'agglomération utilisé par ADDTREE (Sattath et Tversky 1977) repose sur la condition des quatre points (cf. § 1.3). Les distances évolutives dont on dispose ne sont que des approximations des distances réelles, il est donc illusoire d'espérer que (δ_{ij}) satisfasse exactement la condition des quatre points. Par contre, on peut raisonnablement espérer que même si les deux plus grandes des trois sommes ne sont pas égales, elles restent néanmoins toutes deux supérieures à la troisième somme. Cette hypothèse, permet d'utiliser une version affaiblie de la condition des quatre points pour inférer la topologie de chaque groupe de quatre taxons.

Si l'on agglomère les deux sous-arbres (éventuellement réduits à un seul taxon) 1 et 2, on ajoute une branche interne à la phylogénie courante qui sépare cette paire de toute les autres paires *x* et *y* ; on se trouve donc dans la configuration représentée par la Figure 8.

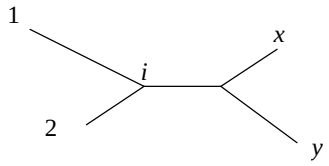


Figure 8 : configuration de 1, 2, *x* et *y* après l'agglomération de 1 et 2

Dans cette configuration, 1 et 2 sont voisins par rapport à *x* et *y* et, suivant la version affaiblie de la condition des quatre points (équation (2) page 14), on doit avoir :

$$\delta_{12} + \delta_{xy} \leq MIN(\delta_{1x} + \delta_{2y}, \delta_{1x} + \delta_{2y}).$$

Comme le souligne Gascuel (1994), cette inéquation peut être ré-écrite en utilisant la fonction de Heaviside notée $H(t)$ et qui vaut 1 lorsque $t \geq 0$ et 0 sinon. La condition pour que 1 et 2 soient voisins par rapport à *x* et *y* s'écrit alors :

$$H(\delta_{1x} + \delta_{2y} - \delta_{12} - \delta_{xy})H(\delta_{1y} + \delta_{2x} - \delta_{12} - \delta_{xy}) = 1$$

Le nombre de paires *x*, *y* pour lesquelles 1 et 2 sont voisins correspond au score de voisinage entre 1 et 2. Ce score, que l'on note V_{12}, est calculé grâce à l'équation :

$$V_{12} = \sum_{r \geq x > y \geq 3} [H(\delta_{1x} + \delta_{2y} - \delta_{12} - \delta_{xy})H(\delta_{1y} + \delta_{2x} - \delta_{12} - \delta_{xy})]$$

ADDTREE choisit la paire ayant le critère de voisinage maximal. Puis, il réduit la matrice de distances et estime les distances entre le nouveau nœud *i* (racine du sous-arbre regroupant 1 et 2) et tout autre nœud *j*, en utilisant la formule suivante :

$$\delta_{ij} = \frac{1}{2}(\delta_{1j} + \delta_{2j})$$

ADDTREE se programme de manière naturelle en $O(n^5)$ en effectuant chacune des $(n-3)$ agglomérations en $O(r^4)$. On peut réduire cette complexité en utilisant un raffinement algorithmique proposé par (Elemento et Gascuel 2002). Dans ce cas, seule la première agglomération est faite en $O(r^4)$ puis, grâce à la mémorisation des scores de voisinages les agglomérations suivantes sont effectuées en $O(r^3)$, on réduit ainsi la complexité de ADDTREE à $O(n^4)$.

2.2.2.2 NJ

Le critère d'agglomération utilisé par l'algorithme Neighbor Joining (NJ) introduit par (Saitou et Nei 1987) repose sur l'idée que l'évolution est économe. On cherche donc l'histoire évolutive la plus courte qui corresponde aux distances observées.

Dans cette approche, les longueurs des branches d'une phylogénie *T* induisant une matrice de distances (d_{ij}) sont ajustées à partir des valeurs de la matrice (δ_{ij}) de manière à minimiser le critère des moindres carrés non ordinaires :

$$\sum_{i \neq j} (\delta_{ij} - d_{ij})^2$$

L'arbre recherché par NJ est celui dont la somme des longueurs de branches (ainsi ajustées) est minimale. Dans ce qui suit, nous présentons la version simplifiée de NJ due à Studier et Keppler (1988). Gascuel (1994) montre que cette version est équivalente à la version originale, et décrit les interprétations possibles du critère d'agglomération utilisé.

En notant Q_{12} la valeur du critère correspondant à l'agglomération des sous-arbres 1 et 2 (éventuellement réduits à un seul taxon), la paire agglomérée par NJ est celle qui minimise

$$Q_{12} = (r-2)\delta_{12} - \Delta_1 - \Delta_2 \text{ avec } \Delta_x = \sum_{y=1}^{r} \delta_{xy}$$

Une fois que la paire 1, 2 à agglomérer est sélectionnée, NJ estime la longueur des branches (1, *i*) et (2, *i*) en utilisant les formules suivantes :

$$\delta_{1i} = \frac{1}{2}(\delta_{12} + \frac{\Delta_1 - \Delta_2}{r-2}) \quad \text{et} \quad \delta_{2i} = \frac{1}{2}(\delta_{12} + \frac{\Delta_2 - \Delta_1}{r-2}) \qquad \textbf{(15)}$$

Enfin, NJ réduit la matrice de distances en enlevant toutes les distances associées à 1 ou à 2, et en estimant les distances entre le nouveau nœud *i* et tout autre nœud *j*, en utilisant la formule suivante :

$$\delta_{ij} = \frac{1}{2}(\delta_{1j} - \delta_{1i}) + \frac{1}{2}(\delta_{2j} - \delta_{2i}) \qquad \textbf{(16)}$$

Les arbres reconstruits par NJ sont généralement très proches de ceux reconstruits par ADDTREE. Bien qu'ils semblent différents, les critères qu'ils optimisent sont en fait assez proches. En effet, (Gascuel 1994) montre qu'il est équivalent de choisir la paire qui optimise Q_{12} ou de choisir celle qui optimise une version pondérée du critère de voisinage de ADDTREE que l'on note V_{12} et qui est définie par :

$$V_{12} = \sum_{r \geq x > y \geq 3} [(\delta_{1x} + \delta_{2y} - \delta_{12} - \delta_{xy}) + (\delta_{1y} + \delta_{2x} - \delta_{12} - \delta_{xy})]$$

Par contre, la complexité en temps de NJ est plus faible que celle de ADDTREE. NJ, pré-calcule les valeurs Δ_x, il peut donc choisir en $O(r^2)$ la paire 1, 2 qui est remplacée par un nouveau nœud *i*. Puis il calcule Δ_i en $O(r)$ et met à jour les $O(r)$ valeurs Δ_x en ajoutant à chacune la valeur $(\delta_{ix} - \delta_{1x} - \delta_{2x})$. Chacune des $(n-3)$ agglomérations a donc une complexité de $O(r^2)$ et la complexité totale de NJ est $O(n^3)$. L'idée de mettre à jour les valeurs Δ_x au lieu de les re-calculer entièrement à chaque étape, est exploité dans la dernière version de PAUP (Swofford 2002) (communication personnelle de D. Bryant). De nombreux travaux ont été menés pour améliorer les performances de NJ, notamment en prenant en compte le fait que les distances sont obtenues à partir de séquences moléculaires. C'est ce que font des méthodes comme BIONJ (Gascuel 1997) et WEIGHBOR (Bruno, Socci et Halpern 2000) tout en conservant une complexité de $O(n^3)$.

2.2.2.3 BIONJ

Lorsque NJ estime les nouvelles distances δ_{ij} (équation (16) page 39), il donne le même poids $(1/2)$ à l'estimation issue de chacun des deux nœuds agglomérés. A chaque agglomération BIONJ utilise un critère statistique pour déterminer le poids qu'il accorde à chacune de ces estimations et la formule de réduction qu'il utilise devient donc :

$$\delta_{ij} = \lambda_{12}(\delta_{1j} - \delta_{1i}) + (1 - \lambda_{12})(\delta_{2j} - \delta_{2i}) \quad \text{avec } \lambda_{12} \in [0,1] \qquad \textbf{(17)}$$

La valeur de λ_{12} dépend uniquement de la paire 1, 2 qui est agglomérée. Les distances entre le nouveau nœud *i* et tout autre nœud *j* sont donc estimées avec la même valeur λ_{12}, et cette valeur est choisie de manière à minimiser la somme des variances des estimateurs δ_{ij}. Pour ajuster ces valeurs λ, BIONJ gère donc également une matrice de variance. A, chaque étape cette matrice est, elle aussi, réduite en utilisant une formule analogue à (17). BIONJ estime la variance des distance initiales δ_{xy} qui séparent deux séquences de

longueur l par δ_{xy} / l, et la covariance de deux distances initiales δ_{xy} et δ_{zt} par $\delta_{xy,zt} / l$ (le terme $\delta_{xy,zt}$ est défini comme pour l'équation (2) page 14). Ces estimations des variances et des covariances, sont valides autour de 0 pour tous les modèles d'évolution (Nei et Jin 1989 ; Bulmer 1991 ; Gascuel 1997).

2.2.2.4 WEIGBOR

Dans BIONJ la qualité des estimations des distances évolutives intervient uniquement lors de la réduction de la matrice de distances. WEIGHBOR prend également cette information en compte dans son critère d'agglomération. Au lieu d'utiliser un critère d'agglomération basé sur le principe d'évolution minimum, WEIGHBOR utilise un critère basé sur la vraisemblance. Pour cela, les distances sont modélisées par des variables aléatoires suivant une loi Gaussienne dont il faut évaluer la variance. Pour estimer cette variance, il se base sur celle de l'estimateur de Jukes et Cantor (le calcul de la variance de cet estimateur est détaillé dans la partie 4.1.3).

Le critère d'agglomération utilisé par WEIGHBOR repose sur le fait que lorsque (δ_{ij}) est une distance d'arbre, les nœuds 1 et 2 sont voisins si, et seulement si, ils respectent les deux propriétés suivantes :

1. Additivité : $\delta_{1x} - \delta_{2x}$ est indépendant du troisième nœud x
2. Positivité : $\delta_{1x} + \delta_{2y} - \delta_{12} - \delta_{xy} \geq 0$ pour tout couple de nœuds x, y différents de 1 et de 2

La première propriété repose sur le fait que si 1 et 2 constituent une paire de voisins reliés à un même nœud i, alors pour tout x (différent de 1 et de 2) on a $\delta_{1x} - \delta_{2x} = \delta_{1i} - \delta_{2i}$. La seconde repose sur la condition des quatre points. En pratique, la matrice (δ_{ij}) n'est pas une distance d'arbre et WEIGHBOR agglomère la paire qui s'éloigne le moins de ces deux critères. L'écart à l'un de ces critères est mesuré en s'appuyant sur la variance des estimateurs des distances. Par exemple, pour la paire 1, 2, l'écart au critère d'additivité est défini par :

$$\text{Additivité}(1,2) = \sum_{x \neq 1,2} \frac{\left((\delta_{1x} - \delta_{2x}) - \overline{(\delta_{1x} - \delta_{2x})}\right)^2}{\text{var}(\delta_{1x}) + \text{var}(\delta_{2x})}$$

où $\overline{(\delta_{1x} - \delta_{2x})}$ représente la valeur moyenne de $\left(\delta_{1x} - \delta_{2x}\right)$. Une formule analogue est utilisée pour mesurer l'écart au critère de positivité. Afin de conserver une complexité en $O(n^3)$, WEIGBOR utilise le critère d'additivité pour sélectionner les paires les plus intéressantes. Parmi les paires retenues, celle qui est finalement agglomérée est sélectionnée suivant le critère de positivité.

2.2.3 FITCH : une méthode d'insertion

Une manière possible d'estimer la différence qui sépare la matrice de distances (d_{ij}) d'une phylogénie valuée *T* et la matrice de distances (δ_{ij}) est d'utiliser une variante de la formule suivante :

$$\sum_{i \neq j} w_{ij} (\delta_{ij} - d_{ij})^2 \qquad \textbf{(18)}$$

où w_{ij} est le poids que l'on accorde à l'adéquation entre δ_{ij} et d_{ij}. Toutes les distances de la matrice initiale de distances (δ_{ij}) ne sont pas estimées avec la même fiabilité. Les poids w_{ij} permettent de prendre ce phénomène en compte en accordant moins d'importance aux distances dont les estimations sont les moins fiables. On peut alors choisir de ne pas prendre ce phénomène en compte ($w_{ij} = 1$), d'estimer la variance de chaque distance suivant la même estimation que BIONJ et d'utiliser l'inverse de cette estimation comme pondération ($w_{ij} = l / \delta_{ij}$). Dans ce dernier cas, si l'on suppose que toutes les séquences ont la même longueur, on peut simplement utiliser $w_{ij} = 1 / \delta_{ij}$. Enfin, une troisième pondération qui en pratique donne les meilleurs résultats est $w_{ij} = 1 / \delta_{ij}^2$.

Le programme FITCH (Felsenstein 1993) suit l'approche proposée par (Fitch et Margoliash 1967) et cherche à retrouver la phylogénie qui minimise le critère des moindres carrés pondérés c'est-à-dire le critère de l'équation (18). Il permet d'utiliser les différentes pondérations décrites ci-dessus ($w_{ij} = 1, w_{ij} = 1 / \delta_{ij}$ et $w_{ij} = 1 / \delta_{ij}^2$) Dans un premier temps, FITCH construit un arbre suivant le processus d'insertion, après chaque insertion il teste les réarrangements de type NNI. Puis, une fois que tous les taxons sont insérés, il effectue éventuellement une recherche plus longue en utilisant des réarrangements de type re-branchement de sous-arbres.

2.3 Méthodes de parcimonie

Dans cette partie, nous présentons de manière détaillée les méthodes de parcimonie. Les algorithmes utilisés par ces méthodes sont très proches de ceux utilisés par le maximum de vraisemblance. En particulier, ces approches utilisent toutes deux des variantes du "double parcours récursif". Ce type de parcours d'arbre est également utilisé par les méthodes de quadruplets et a été formalisée par (Berry et Gascuel 2000). L'étude des méthodes de parcimonie nous permet d'illustrer le fonctionnement du double parcours récursif sur un exemple très simple et complète ainsi la description plus formelle qui est fournie en annexe 1, dans notre article (Ranwez et Gascuel 2001, Pp. 91-93). De plus, à notre connaissance, il n'existe aucune autre description détaillée des techniques algorithmiques utilisées par les programmes de parcimonie.

Dans un premier temps, nous présentons le principe général des méthodes de parcimonie, et nous détaillons la manière dont on calcule la parcimonie d'un arbre. Nous illustrons

ensuite, par un exemple simple, la manière dont le double parcours récursif est utilisé pour rechercher l'arbre le plus parcimonieux.

2.3.1 Principe général et définitions

Les méthodes de parcimonie reposent sur l'idée que l'histoire évolutive la plus plausible est celle qui nécessite un minimum de mutations. Ces méthodes recherchent donc parmi tous les arbres possibles, et toutes les séquences possibles aux nœuds internes, la combinaison qui minimise le nombre total de mutations requis pour expliquer les données observées. Les différentes mutations peuvent avoir des coûts différents, et il est possible d'attribuer des poids différents à chacun des sites.

Dans cette approche, les sites peuvent être traités de manière indépendante. Le nombre de mutations correspondant à un jeu de séquences particulier est égal à la somme des mutations requises par chacun des sites. Les coûts attribués aux mutations sont généralement symétriques, autrement dit, le coût attribué à la mutation d'un nucléotide *b* en un nucléotide *c* est identique au coût de la mutation de *c* en *b*. Sous cette hypothèse, la parcimonie d'un arbre est la même, quelle que soit la position de la séquence ancestrale, et la parcimonie $P(T)$ d'un arbre T dont les séquences ancestrales sont inconnues, correspond à la parcimonie minimale des arbres T_s obtenus en associant des séquences nucléotidiques aux nœuds internes de T. Pour un tel arbre T_s les séquences ancestrales sont donc connues et on peut calculer la valeur de parcimonie de chacune de ses branches. La somme de ces valeurs définit la parcimonie de T_s . Nous verrons dans la suite, comment il est possible de calculer la parcimonie $P(T)$ correspondant à la définition ci-dessus sans considérer explicitement les arbres T_s.

Dans la suite de cette partie nous traitons uniquement le cas le plus simple, où tous les sites ont le même poids et toutes les mutations ont le même coût (cependant notre présentation se généralise aisément au cas où l'on pondère les sites et où l'on utilise des coûts de mutation symétrique). Dans ce cas simple, la valeur de parcimonie $P(S_i, S_j)$ qui sépare les deux séquences S_i et S_j correspond simplement au nombre de différences observées entre ces deux séquences. Les méthodes de parcimonies ne prennent donc en compte que les mutations entre S_i et S_j qui sont encore visibles.

2.3.2 Calcul de la parcimonie d'un arbre

La Figure 9 présente un cas simple où l'on considère quatre séquences de deux nucléotides. Dans cette figure, les substitutions requises sont indiquées par des astérisques, les solutions optimales nécessitent quatre substitutions (deux pour chaque sites). Les solutions (a) et (b) sont deux solutions optimales, qui différent par le choix des nucléotides ancestraux.

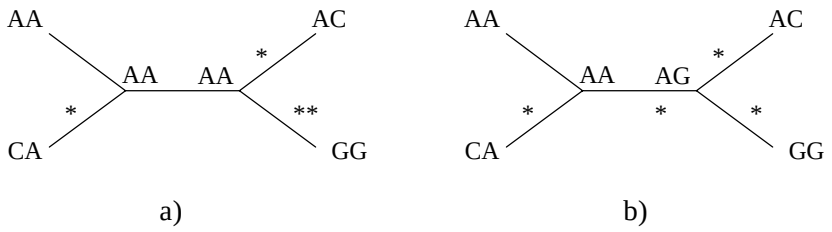


Figure 9 : deux histoires évolutives ayant la même valeur de parcimonie

Dans cette figure chaque substitution est représentée par une astérisque. Ces deux histoires évolutives nécessitent quatre substitutions, elles ont donc la même valeur de parcimonie.

Pour cette variante simple, le calcul de la parcimonie d'un arbre dont la topologie est fixée se fait en parcourant une seule fois l'arbre. Pour cela on choisit, arbitrairement une branche de l'arbre sur laquelle se trouvera la racine *r* de l'arbre. On oriente ainsi l'arbre étudié, et chacun de ses nœuds internes possède alors deux nœuds fils. Comme les sites peuvent être traités indépendamment les uns des autres, il suffit de décrire l'algorithme dans le cas où il n'y a qu'un seul site.

Pour chaque nœud i de l'arbre, on détermine la parcimonie $P(i)$ du sous-arbre dont ce nœud est racine et l'ensemble $X(i)$ des nucléotides qui peuvent être utilisés au nœud i pour obtenir $P(i)$. Le calcul de la parcimonie d'un arbre se fait par récurrence. Initialement, on ne connaît que les valeurs de P et de X des feuilles de l'arbre. En effet, pour chaque feuille f de l'arbre, $P(f) = 0$ et $X(f)$ est réduit au seul nucléotide observé à cette feuille. Les valeurs P et X des autres nœuds sont obtenues en parcourant l'arbre suivant l'ordre postfixe d'un parcours en profondeur. Un nœud i est donc traité après son fils gauche g et son fils droit d, et les valeurs $X(i)$ et $P(i)$ sont calculées à partir de $X(g)$, $X(d)$, $P(g)$ et $P(d)$ en suivant la règle suivante (Fitch 1971) :

$$\begin{cases} \text{si } X(g) \cap X(d) \neq \varnothing & \text{alors } X(i) = X(g) \cap X(d) & \text{et } P(i) = P(g) + P(d) \\ \text{sinon} & X(i) = X(g) \cup X(d) & \text{et } P(i) = P(g) + P(d) + 1 \end{cases} \quad \textbf{(19)}$$

La racine r de l'arbre est le dernier nœud traité, et $P(r) = P(T)$. La Figure 10 montre, sur la version enracinée de l'arbre de la Figure 1 (page 12) la manière dont fonctionne cet algorithme.

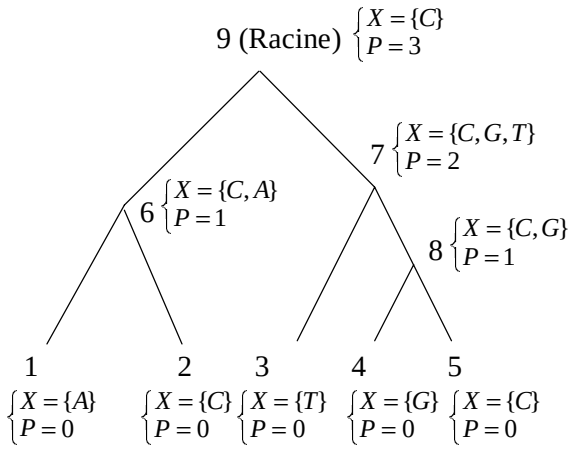


Figure 10 : calcul de la parcimonie d'un arbre

Le calcul de la parcimonie d'un arbre se fait par récurrence en calculant à partir des feuilles les valeurs de *P* et de *X* pour chaque nœud. En chaque nœud, la valeur de *P* représente la valeur de parcimonie du sous-arbre ayant ce nœud pour racine, et l'ensemble *X* contient les nucléotides possibles pour obtenir cette valeur de parcimonie.

Les valeurs de *X* et de *P* indiquées sur cet arbre correspondent au traitement d'un seul site. Pour chaque nœud *i*, il faut calculer les valeurs de *X* et de *P* pour tous les sites. On appelle l'ensemble de ces valeurs le vecteur de parcimonie associé au sous-arbre ayant le nœud *i* pour racine et on le note $PV(T_i)$. Pour un arbre de *n* feuilles et des séquences de longueur *l*, la complexité en temps pour calculer l'ensemble de ces vecteurs est $O(nl)$.

Le calcul des ensembles *X* peut être accéléré en codant chaque ensemble par un nombre de 4 bits indiquant la présence (le bit est à 1) ou l'absence (le bit est à 0) des quatre nucléotides dans cet ensemble. L'intersection de deux ensembles se fait alors par un simple "ET logique" bit à bit, leur union par un "OU logique" bit à bit, et un ensemble est vide si son codage vaut 0000. En utilisant cette technique il est rapide de calculer la parcimonie d'un arbre, cependant la recherche de l'arbre le plus parcimonieux est un problème NP-difficile (Foulds et Graham 1982).

2.3.3 Recherche de l'arbre le plus parcimonieux

Felsenstein (1993) utilise la même heuristique d'insertion et de ré-arrangement d'arbre dans le programme DNAPARS et dans le programme FITCH. Dans cette partie nous précisons certains détails algorithmiques utilisés au cours du processus d'insertion, cela nous permet d'introduire de manière simple des concepts également utilisés en maximum de vraisemblance.

Reconsidérons l'exemple du calcul de la parcimonie déjà vu plus haut (Figure 10). Dans cet exemple, la parcimonie de l'arbre est calculée en supposant que la racine de l'arbre est

sur la branche (6,7). Une fois que ce calcul est effectué, on peut rapidement calculer la valeur de parcimonie de l'arbre que l'on obtient en insérant un nouveau taxon *k* sur la branche (6,7). En effet, ce calcul peut se faire en considérant uniquement le vecteur de parcimonie du nœud/taxon *k* et celui du nœud 9 qui est inchangé. Ce qui permet de calculer la parcimonie de chaque site du nouvel arbre en $O(1)$.

D'une manière plus générale, pour évaluer l'insertion du taxon *k* sur une branche (*i*, *j*), il faut disposer des vecteurs de parcimonie $PV(T_i)$ et $PV(T_j)$. Une manière possible de calculer ces vecteurs est d'utiliser directement le calcul récursif de la parcimonie, tel qu'il est décrit dans la partie 2.3.2, en enracinant l'arbre *T* sur la branche (*i*, *j*). Pour cela, il faut, pour chaque nœud de l'arbre, considérer tous les sites possibles. Si l'arbre *T* possède *n* feuilles et que les séquences étudiées sont de longueurs *l*, alors le temps de calcul nécessaire pour obtenir $PV(T_i)$ et $PV(T_j)$ est $O(nl)$. Pour tester l'insertion de *k* sur l'ensemble des branches de *T*, il faut effectuer un calcul similaire pour chacune de ces branches. Le temps requis pour calculer tous ces vecteurs de parcimonie est donc, a priori, $O(n^2 l)$.

En utilisant la technique du "double parcours récursif" on peut calculer l'ensemble de ces vecteurs en $O(nl)$. Cette technique effectue deux parcours en profondeur de l'arbre *T*. Lors du premier parcours les nœuds de *T* sont traités suivant l'ordre postfixe, lors du second parcours ils sont traités suivant l'ordre préfixe.

Pour utiliser le double parcours récursif, on choisit arbitrairement une feuille qui sera la racine de l'arbre. On oriente ainsi l'arbre étudié, et chaque nœud interne possède alors un nœud père noté *a* (pour ancestral) et deux nœuds fils notés *g* et *d* (pour gauche et droit). Chaque nœud interne *i* est donc relié à trois sous-arbres disjoints de *T*, qui sont notés A_i, G_i et D_i et qui contiennent respectivement le nœud *a*, le nœud *g* et le nœud *d*. Le sous-arbre obtenu en reliant G_i et D_i par les branches (*g*,*i*) et (*d*,*i*) est enraciné en *i* et on le note $T_{i[GD]}$. De même, le sous-arbre obtenu en reliant A_i et G_i par les branches (*a*,*i*) et (*g*,*i*) est noté $T_{i[AG]}$, et celui obtenu en reliant A_i et D_i par les branches (*a*,*i*) et (*d*,*i*) est noté $T_{i[AD]}$. Un même sous-arbre possède plusieurs notation suivant le nœud que l'on utilise comme référence. En particulier, $G_i = T_{g[GD]}$, $D_i = T_{d[GD]}$, et le sous-arbre A_i correspond soit au sous-arbre $T_{a[AG]}$ soit au sous-arbre $T_{a[AD]}$. Pour référencer le sous-arbre A_i par rapport au nœud *a* on utilisera la notation $T_{a[AX]}$. La Figure 11 (page 46) résume ces notations. Dans cette figure, les sous-arbres $T_{i[GD]}$, $T_{i[AG]}$, $T_{i[AD]}$, $T_{g[GD]}$, $T_{d[GD]}$ et $T_{a[AX]}$ sont représentés par une flèche qui pointe vers le sous-arbre correspondant.

Par convention, on suppose que l'unique fils de la feuille *r* utilisée comme racine de *T*, est son fils gauche. Pour harmoniser les notations, le sous-arbre réduit au nœud racine *r* est noté $T_{r[AD]}$ et ceux réduits à une feuille *f* (différente de *r*) sont notés $T_{f[GD]}$.

Lors du double parcours récursif, le premier parcours postfixe en profondeur de l'arbre permet de calculer l'ensemble des vecteurs de parcimonie de type $PV(T_{i[GD]})$, le parcours préfixe en profondeur permet ensuite de calculer ceux de type $PV(T_{i[AG]})$ et $PV(T_{i[AD]})$.

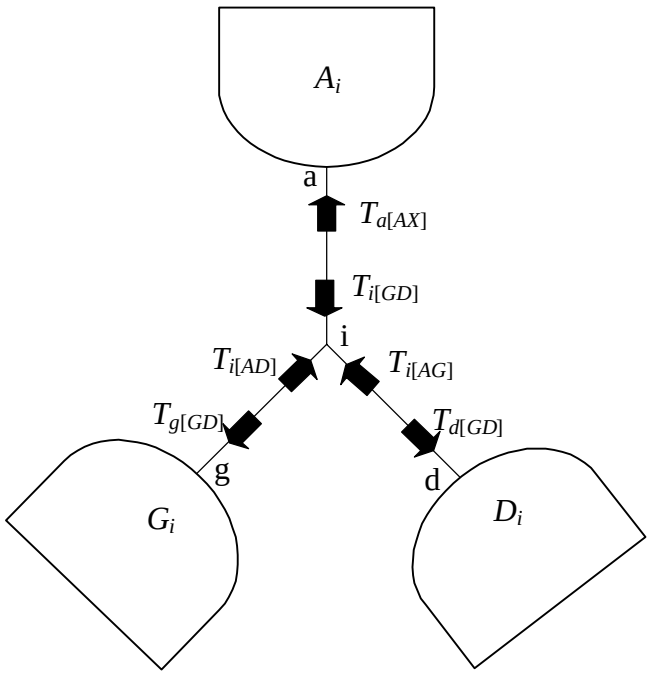


Figure 11 : notations utilisées lors du traitement d'un nœud *i*

Dans cette figure, les sous-arbres $T_{i[GD]}$, $T_{i[AG]}$, $T_{i[AD]}$, $T_{g[GD]}$, $T_{d[GD]}$ et $T_{a[AX]}$ sont représentés par une flèche qui pointe vers le sous-arbre correspondant.

Initialement on connaît uniquement les vecteurs de parcimonie des feuilles de l'arbre. Pour chaque feuille *f* autre que la racine, on connaît donc les vecteurs de parcimonie $PV(T_{f[GD]})$. Et l'on connaît également le vecteur $PV(T_{r[AD]})$ associé à la racine de *T*. Lors du parcours postfixe, un nœud est traité après ses fils, l'équation (19) page 43 permet de calculer, pour chaque nœud *i* de *T*, le vecteur $PV(T_{i[GD]})$ à partir de $PV(T_{g[GD]})$ et de $PV(T_{d[GD]})$.

Lors du second parcours préfixe, un nœud interne *i* est traité après son nœud père *a*. L'équation (19) permet, pour chaque nœud *i* de *T*, de calculer le vecteur $PV(T_{i[AD]})$ à partir de $PV(T_{a[AX]})$ et de $PV(T_{d[GD]})$ et de calculer $PV(T_{i[AG]})$ à partir de $PV(T_{a[AX]})$ et de $PV(T_{g[GD]})$.

La Figure 12, page 48, illustre les différentes étapes qui permettent de calculer l'ensemble des vecteurs de parcimonie grâce au double parcours récursif. Sur cette figure, les flèches entièrement noires indiquent les vecteurs de parcimonie disponibles au début de chacun des deux parcours. Les autres flèches indiquent les vecteurs qui sont calculés au cours de ces parcours. Les numéros en gras, à cotés des flèches, correspondent à l'étape au cours de laquelle le vecteur de parcimonie est calculé. Ces différentes étapes sont décrites ci-dessous. On dispose initialement de $PV(T_{3[AD]})$, $PV(T_{1[GD]})$, $PV(T_{2[GD]})$, $PV(T_{4[GD]})$ et $PV(T_{5[GD]})$ qui sont représentés par les flèches noires de la Figure 12.a (page 48).

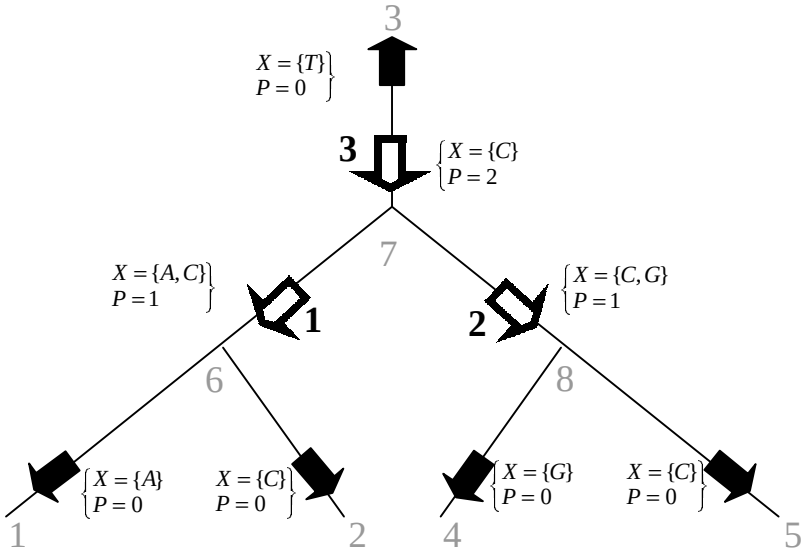
Lors du premier parcours postfixe en profondeur on obtient successivement (Figure 12.a page 48) :

1. $PV(T_{6[GD]})$ à partir de $PV(T_{1[GD]})$ et de $PV(T_{2[GD]})$
2. $PV(T_{8[GD]})$ à partir de $PV(T_{4[GD]})$ et de $PV(T_{5[GD]})$
3. $PV(T_{7[GD]})$ à partir de $PV(T_{6[GD]})$ et de $PV(T_{8[GD]})$

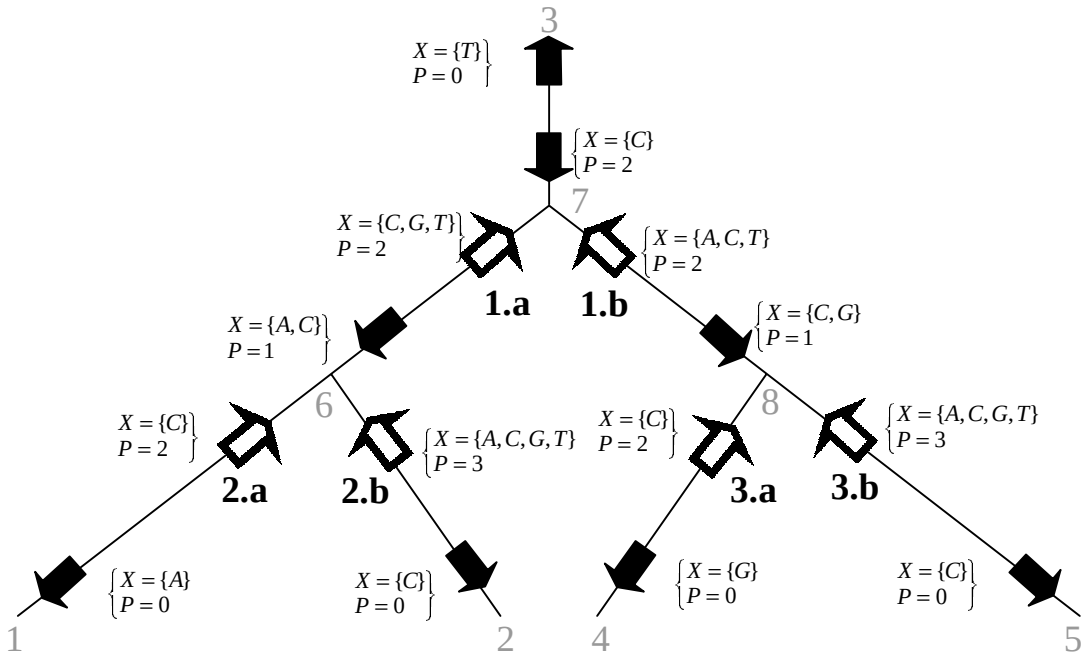
Ces valeurs sont ensuite utilisées lors du second parcours préfixe en profondeur pour calculer successivement (Figure 12.b page 48) :

1. les vecteurs de parcimonie manquant au nœud 7 :
    a. $PV(T_{7[AD]})$ à partir de $PV(T_{3[AD]})$ et de $PV(T_{8[GD]})$
    b. $PV(T_{7[AG]})$ à partir de $PV(T_{3[AD]})$ et de $PV(T_{6[GD]})$
2. les vecteurs de parcimonie manquant au nœud 6 :
    a. $PV(T_{6[AD]})$ à partir de $PV(T_{7[AD]})$ et de $PV(T_{2[GD]})$
    b. $PV(T_{6[AG]})$ à partir de $PV(T_{7[AD]})$ et de $PV(T_{1[GD]})$
3. les vecteurs de parcimonie manquant au nœud 8 :
    a. $PV(T_{8[AD]})$ à partir de $PV(T_{7[AG]})$ et de $PV(T_{5[GD]})$
    b. $PV(T_{8[AG]})$ à partir de $PV(T_{7[AG]})$ et de $PV(T_{4[GD]})$

Ainsi, après ces deux parcours de l'arbre T, on dispose de l'ensemble des vecteurs de parcimonie et donc des informations nécessaires pour tester en $O(1)$ l'insertion d'un nouveau taxon sur n'importe quelle branche de l'arbre T. Considérons par exemple, le cas où l'on souhaite insérer un nouveau taxon k associé au nucléotide G sur l'arbre T. La parcimonie de l'arbre obtenu en insérant k sur la branche (6,7) s'obtient à partir de $PV(T_{6[GD]})$ et de $PV(T_{7[AD]})$ et vaut 4, alors que la parcimonie obtenue en insérant k sur la branche (4,8) s'obtient à partir de $PV(T_{4[GD]})$ et de $PV(T_{8[AD]})$ et vaut 3.



a) parcours postfixe



b) parcours préfixe

Figure 12 : calcul de l'ensemble des vecteurs de parcimonies

Sur cette figure, les flèches entièrement noires indiquent les vecteurs de parcimonie disponibles au début de chacun des deux parcours. Les autres flèches indiquent les vecteurs qui sont calculés au cours de ces parcours ; les numéros en gras, à cotés de ces flèches, correspondent à l'étape au cours de laquelle le vecteur de parcimonie est calculé.

2.4 Maximum de vraisemblance

Les méthodes de maximum de vraisemblance utilisent un modèle mathématique du processus d'évolution des séquences pour définir la probabilité qu'une phylogénie puisse produire les séquences observées. Elles cherchent ensuite la phylogénie pour laquelle cette probabilité est maximale. Pour utiliser une méthode de maximum de vraisemblance, il faut donc être capable de choisir un modèle d'évolution, d'estimer ses paramètres, et de calculer la vraisemblance d'un arbre pour ce modèle.

La première partie présente des techniques possibles pour choisir un modèle d'évolution adapté à un jeu de données particulier. La seconde partie, détaille le calcul de la vraisemblance d'un arbre valué. La troisième partie, présente le calcul de la vraisemblance d'un arbre non-valué, et illustre la manière dont un parcours habile de l'arbre permet d'accélérer ce calcul. La dernière partie décrit les particularités du processus d'insertion qui est généralement utilisé pour trouver un arbre de vraisemblance élevée.

2.4.1 Choix du modèle d'évolution et ajustement de ses paramètres

L'approche par maximum de vraisemblance s'appuie explicitement sur un modèle de l'évolution, et le choix du modèle utilisé influence les performances de cette méthode (Kelsey, Crandall et Voevodin 1999). Si l'on choisit mal le modèle d'évolution, non seulement on risque de diminuer la probabilité de reconstruire l'arbre correct, mais il est même possible que la probabilité de reconstruire un arbre correct diminue lorsque la longueur des séquences augmente (Sullivan et Swofford 1997).

Même si on se limite aux 8 modèles de la Figure 3 (page 27) associés ou non avec une loi gamma pour modéliser la variation des sites, il faut néanmoins choisir parmi 16 modèles différents. Il existe plusieurs tests statistiques permettant de mesurer l'adéquation des données à un modèle particulier. Cela ne suffit pas pour comparer les modèles entre eux, en effet l'ajout de paramètres semble toujours augmenter la pertinence du modèle. Cependant l'utilisation d'un modèle complexe augmente les temps de calcul et nécessite d'estimer davantage de paramètres. Si le nombre de paramètres à estimer est trop important par rapport à la quantité d'information dont on dispose (la taille et le nombre de séquences), les paramètres du modèle risquent d'être mal estimés. Le modèle utilisé doit donc à la fois être suffisamment complexe pour expliquer les données, et suffisamment simple pour que ses paramètres puissent être correctement estimés. Lorsque deux modèles sont emboîtés (i.e. l'un généralise l'autre), on peut déterminer celui des deux modèles qui est le plus adapté pour traiter un jeu de données particulier. Pour cela, on utilise des tests statistiques tels que le "test des rapports de vraisemblance" ou le "critère d'Akaike".

Ces tests constituent donc une étape préliminaire à la reconstruction phylogénétique par une méthode de maximum de vraisemblance. Or ils utilisent une phylogénie des séquences

étudiées pour comparer la pertinence des différents modèles. Cela pourrait être problématique, mais l'étude de Posada et Crandall (2001) indique clairement que pour des séquences de taille raisonnable (plus de 100 nucléotides) la pertinence des tests reste quasiment la même que l'on utilise la véritable phylogénie ou que l'on utilise la phylogénie reconstruite avec NJ, et cela quel que soit le modèle utilisé pour calculer les distances fournies à NJ.

Le processus d'évolution le long d'un arbre dépend de la topologie et de la longueur des branches de cet arbre, mais aussi des paramètres du modèle d'évolution que l'on considère. Idéalement, il faudrait donc optimiser la vraisemblance pour l'ensemble de ces paramètres. Pour chaque arbre étudié, il faudrait alors ajuster les longueurs de ses branches et les valeurs des paramètres du modèle d'évolution. Ces ajustements nécessitent l'utilisation d'une méthode d'optimisation dans un espace de grande dimension. Une telle approche est coûteuse en temps de calcul, et selon (Swofford et al. 1996, p. 445), elle ne peut s'appliquer qu'à de très petits jeux de données contenant moins d'une dizaine de taxons.

En pratique, on utilise donc les mêmes valeurs de paramètres pour tous les arbres, et la vraisemblance d'un arbre n'est optimisée, par la suite, qu'en fonction des longueurs de branches. En effet, il est possible d'utiliser un arbre différent (mais proche) de l'arbre vrai, pour estimer correctement les paramètres de modèles issus du GTR et le paramètre de forme de la loi gamma (Yang, Goldman et Friday 1994). On peut donc utiliser un arbre reconstruit, par exemple, par NJ à la fois pour choisir un modèle pertinent et pour estimer de manière satisfaisante les paramètres de ce modèle.

2.4.2 Vraisemblance d'un arbre valué

Pour un modèle d'évolution donné, la vraisemblance d'un arbre est la probabilité que le processus d'évolution le long de cet arbre produise les séquences observées aux feuilles. Nous avons décrit un modèle stochastique de l'évolution, appelé GTR. Pour ce modèle, et pour les modèles qui en sont dérivés, nous avons décrit la manière dont il est possible de calculer la probabilité qu'un nucléotide *b* soit transformé en un nucléotide *c* pour une distance évolutive donnée δ. Dans cette partie, nous expliquons comment ces probabilités permettent de calculer la vraisemblance d'un arbre valué. Nous montrons ensuite comment ce calcul peut être étendu de manière à prendre en compte la variabilité des vitesses d'évolution entre sites.

2.4.2.1 Cas du modèle GTR

En notant S_a la séquence ancestrale (que l'on ne connaît pas), π_b la probabilité que le nucléotide *b* soit le nucléotide ancestral, et $L(S_a^s = b;T)$ la probabilité que ce nucléotide *b* ait évolué pour donner les *n* nucléotides observés au site *s* sur les feuilles de *T*, alors la probabilité associée au site *s* par rapport au nucléotide *b* est le produit de π_b et de $L(S_a^s = b;T)$. Ainsi, la vraisemblance de l'arbre *T*, dont la topologie et les longueurs de branches sont connues, est obtenue par la formule suivante :

$$L(T) = \prod_{s=1}^{l} \sum_{b \in \{A,C,G,T\}} \pi_b L(S_a^s = b; T) \quad \textbf{(20)}$$

Le terme $L(S_a^s = b; T)$ est calculé de manière récursive. Supposons que l'arbre T soit constitué de deux sous-arbres T_i et T_j, dont les séquences ancestrales sont respectivement S_i et S_j. Dans ce cas, la vraisemblance de T se calcule à partir de celles de T_i et de T_j. Si pour une distance évolutive δ, on note $P_{bc}(\delta)$ la probabilité qu'un nucléotide b devienne c, on a alors :

$$L(S_a^s = b; T) = \prod_{x \in \{i,j\}} \sum_{c \in \{A,C,G,T\}} P_{bc}(\delta_{ax}) L(S_x^s = c; T_x) \quad \textbf{(21)}$$

où δ_{ax} représente le taux d'évolution (*i.e.* la longueur de la branche) entre S_a et S_x, avec $x = i$ ou $x = j$. De manière similaire, la vraisemblance de T_i est calculée à partir des vraisemblances de ses sous-arbres. La récurrence s'arrête lorsque le sous-arbre est réduit à une seule feuille, notée T_f. Cette feuille est associée à une séquence S_f contemporaine et connue, qui définit la vraisemblance de T_f par :

$$\begin{cases} L(S_f^s = b; T_f) = 1 \text{ si } S_f^s = b \\ L(S_f^s = b; T_f) = 0 \text{ sinon} \end{cases} \quad \textbf{(22)}$$

2.4.2.2 Variabilité des vitesses d'évolution

Dans le chapitre précédent, nous avons vu qu'il est possible d'utiliser une loi gamma pour modéliser la distribution des vitesses d'évolution parmi les sites. L'utilisation d'une distribution continue complique le calcul de la vraisemblance. Par contre, il est facile d'utiliser une version discrète de la loi gamma. Cela revient à définir un ensemble fini de classes de sites variant à des vitesses différentes. On connaît la fréquence attendue des différents types de sites, ce qui permet d'estimer la probabilité qu'un site soit dans une classe donnée.

La distance étant le produit de la vitesse par le temps, il est équivalent de considérer que les sites évoluent à des vitesses plus ou moins rapides ou qu'ils évoluent sur des branches plus ou moins longues de l'arbre. Ainsi, pour calculer la vraisemblance de l'arbre T, on associe un facteur multiplicatif λ à chaque classe de sites. Lorsque l'on considère qu'un site s est dans cette classe, on multiplie la longueur de toutes les branches de T par λ, et on calcule la vraisemblance de s sur l'arbre λT ainsi obtenu. Évidemment, on ne sait pas à quelle vitesse évolue un site particulier. La probabilité associée à un site s pour une vitesse λ est donc le produit de la probabilité que s évolue à cette vitesse par la probabilité qu'en évoluant à cette vitesse il donne les n nucléotides présents au site s des feuilles de T. La probabilité d'un site correspond alors à la somme des probabilités de ce site pour les différentes vitesses possibles (théorème des probabilités totales). Si l'on note π_λ la probabilité qu'un site évolue à une vitesse correspondant au facteur multiplicatif λ, et que

l'on note Λ l'ensemble des λ possibles, alors la vraisemblance $L(\Lambda, T)$ d'un arbre T pour cette modélisation des vitesses est :

$$L(\Lambda, T) = \prod_{s=1}^{l} \sum_{\lambda \in \Lambda} \left(\pi_{\lambda} \sum_{b \in \{A,C,G,T\}} \pi_b L\left(S_a^s = b; \lambda T\right) \right) \quad \textbf{(23)}$$

Pour simplifier, on ne considère, dans la suite, qu'une catégorie de site.

2.4.3 Vraisemblance d'un arbre non valué

Une fois que le modèle d'évolution et les valeurs de ses paramètres sont déterminés, la vraisemblance d'un arbre ne dépend plus que de la longueur de ses branches. On peut donc les ajuster de manière à maximiser la vraisemblance de l'arbre. Les méthodes de maximum de vraisemblance cherchent à retrouver l'arbre pour lequel la vraisemblance (une fois ses longueurs de branches ajustées) est maximale.

Cependant, même lorsque les autres paramètres sont fixés, l'ajustement des longueurs des branches d'un arbre est un problème difficile (Chor et al. 2000). Même si l'on cherche à ajuster une seule longueur de branche en considérant que les autres sont fixes, l'ajustement de cette longueur requiert l'utilisation d'une méthode d'optimisation numérique (Felsenstein 1981 ; Olsen et al. 1994). De plus, les ajustements ne sont pas indépendants les uns des autres, les branches ne peuvent donc pas être optimisées indépendamment les unes des autres. L'algorithme d'ajustement qui est généralement utilisé consiste à faire un parcours d'arbre au cours duquel chacune des branches rencontrées est ajustée en considérant que les autres branches sont fixes. Comme l'optimisation d'une branche remet en cause les longueurs des branches qui ont été optimisées auparavant, l'algorithme fait plusieurs parcours de l'arbre. A chacun de ces parcours toutes les branches sont ré-ajustées. L'algorithme s'arrête lorsque le processus converge ou que le nombre de passages dans l'arbre devient trop grand.

Dans un premier temps, nous décrivons la manière dont on peut ajuster la longueur d'une branche lorsque l'on suppose que tous les autres paramètres sont fixés. Dans un second temps, nous expliquons pourquoi on ne peut pas utiliser directement le double parcours récursif pour optimiser efficacement l'ensemble des longueurs de branches. Nous présentons ensuite une variante du double parcours récursif qui permet d'ajuster efficacement l'ensemble des longueurs de branches, et nous illustrons le fonctionnement de ce parcours sur un exemple.

2.4.3.1 Ajustement d'une longueur de branche

Pour ajuster la longueur δ_{ij} d'une branche (i, j) de l'arbre T, on suppose que S_i est la séquence ancestrale. Ainsi $\delta_{ai} = \delta_{ii} = 0$, et l'on obtient à partir des équations (20) et (21) page 51 :

$$L(T) = \prod_{s=1}^{l} \sum_{b \in \{A,C,G,T\}} \left[\pi_b L(S_i^s = b; T_i) \sum_{c \in \{A,C,G,T\}} P_{bc}(\delta_{ij}) L(S_j^s = c; T_j) \right] \quad \textbf{(24)}$$

Si l'on suppose que les autres longueurs de branches sont fixées, on peut utiliser les équations (21) et (22) pour calculer les valeurs numériques correspondant aux termes $L(S_x^s = b; T_x)$ avec x valant i ou j. Chaque terme $L(S_x^s = b; T_x)$ correspond à la probabilité d'observer les nucléotides présents au site s des feuilles de T_x lorsque le nucléotide présent au site s de S_x est le nucléotide b. L'ensemble des valeurs du type $L(S_x^s = b; T_x)$ constitue le "vecteur de vraisemblance" de T_x que nous notons $LV(T_x)$. Ce vecteur contient 4 valeurs (une par nucléotide) pour chacun des l sites. $LV(T_i)$ et $LV(T_j)$ étant complètement définis, on peut utiliser une méthode numérique d'optimisation d'une fonction à un seul paramètre pour ajuster δ_{ij} de manière à optimiser $L(T)$.

Une des méthodes les plus utilisées pour cela est la méthode de Newton-Rapthon (Press et al. 1988). Cette méthode permet de déterminer un zéro d'une fonction $f(x)$, *i.e.* une valeur de x telle que $f(x) = 0$. Elle procède de manière itérative à partir d'une valeur initiale de x. A chaque étape une nouvelle valeur de x (notée x_{i+1}) est déterminée grâce à l'équation de Newton-Rapthon $x_{i+1} = x_i - f(x_i) / f'(x_i)$, où x_i représente la valeur courante de x, $f(x_i)$ la valeur de la fonction en ce point et $f'(x_i)$ la valeur de sa dérivée. Si l'on cherche un extremum de la fonction $f(x)$, cela revient à chercher un zéro de la dérivée de cette fonction, et l'équation de Newton-Raphton devient alors $x_{i+1} = x_i - f'(x_i) / f''(x_i)$. Lorsque l'on considère qu'une seule des longueurs de branche est variable, on peut ajuster cette longueur δ_{ij} de manière à trouver une valeur extrême de $L(T)$ en modifiant successivement la valeur de δ_{ij} suivant l'équation de Newton-Raphton. Pour que cette méthode réponde exactement à notre problème d'optimisation, il est nécessaire de l'adapter pour éviter de converger vers une valeur minimale de la vraisemblance, éviter d'obtenir une valeur de δ_{ij} négative et traiter correctement les cas où la dérivée seconde est nulle. Ces modifications sont décrites de manière détaillée dans la thèse de Nicolas Galtier (1997). L'avantage de cette méthode d'optimisation est qu'elle converge rapidement ; en contre partie, elle nécessite de calculer la dérivée et la dérivée seconde de la fonction de vraisemblance. Une alternative possible permettant d'éviter de calculer ces dérivés est d'utiliser la méthode d'optimisation de Brent (Press et al. 1988, Pp. 299-302). Dans l'article (Ranwez et Gascuel 2002, Pp. 10) fourni en annexe, nous expliquons la manière dont cette méthode utilise une interpolation parabolique de $f(x)$ pour trouver rapidement un de ses extremums.

2.4.3.2 Calcul de l'ensemble des vecteurs de vraisemblance

Comme nous venons de le voir, pour ajuster la longueur d'une branche (i, j), il faut disposer des vecteurs de vraisemblance $LV(T_i)$ et $LV(T_j)$. Une manière possible de calculer ces valeurs est d'utiliser directement le calcul récursif de la vraisemblance tel qu'il est décrit dans la partie 2.4.2.1. Pour cela, il faut, pour chaque nœud de l'arbre, considérer tous

les sites possibles. Le calcul de $LV(T_i)$ et $LV(T_j)$ a donc une complexité en temps de $O(nl)$. Pour optimiser l'ensemble des branches de T, il faut effectuer un calcul similaire pour chacune de ces branches. Le calcul des vecteurs de vraisemblance nécessaires lors de ces optimisations a donc, à priori, une complexité en temps de $O(n^2 l)$. Dans le paragraphe 2.3.3 nous avons décrit la manière dont la technique du double parcours récursif pouvait être utilisée dans le cas de la parcimonie. On peut appliquer la même technique, pour obtenir l'ensemble des vecteurs de vraisemblance également en $O(nl)$. Dans ce qui suit, nous reprenons les notations et les conventions du paragraphe 2.3.3 qui sont illustrées dans la Figure 11 (page 46).

Initialement on connaît uniquement les vecteurs de vraisemblances des feuilles de l'arbre. Pour chaque feuille f de l'arbre autre que la racine, on connaît $LV(T_{f[GD]})$. Pour la feuille r utilisée comme racine de T on connaît $LV(T_{r[AD]})$. Lors du parcours postfixe, un nœud i est traité après ses fils g et d, et l'équation (21), page 51, permet de calculer $LV(T_{i[GD]})$ à partir de $LV(T_{g[GD]})$, $LV(T_{d[GD]})$ et des longueurs des branches (i, g) et (i, d).

Lors du second parcours prefixe, un nœud interne i est traité après son nœud père a. Lors du traitement du nœud i on utilise la formule de récurrence du calcul de la vraisemblance (équation (21) page 51) pour calculer $LV(T_{i[AD]})$ et $LV(T_{i[AG]})$. $LV(T_{i[AD]})$ est obtenu à partir de $LV(T_{d[GD]})$, $LV(T_{a[AX]})$ et des longueurs des branches (i, d) et (i, a) tandis que $LV(T_{i[AG]})$ est obtenu à partir de $LV(T_{g[GD]})$, $LV(T_{a[AX]})$ et des longueurs des branches (i, g) et (i, a).

Ainsi, après ces deux parcours de l'arbre T, on dispose de l'ensemble des vecteurs de vraisemblance et donc des informations nécessaires pour ajuster n'importe quelle longueur de branche de l'arbre T en $O(nl)$. Cependant, dès que l'on modifie l'une des longueurs, on modifie aussi les vecteurs de vraisemblance des sous-arbres qui contiennent cette branche. Si l'on veut que les ajustements suivants prennent en compte la modification de cette longueur, il faut mettre à jour les vecteurs de vraisemblance.

2.4.3.3 Ajustement des longueurs de branches

Il est possible d'adapter le double parcours récursif, de manière à pouvoir ajuster successivement les longueurs des différentes branches de T pour que chaque ajustement soit fait en prenant en compte les précédents. Le but de cette partie est de donner une intuition correcte de cette technique dont l'aspect algorithmique est détaillé dans (Adachi et Hasegawa 1996).

Le premier parcours de l'arbre est inchangé et permet d'obtenir l'ensemble des vecteurs de type $LV(T_{i[GD]})$. Lors du second parcours les longueurs de branches sont ajustées, de sorte que lorsque l'on arrive sur un nœud i, le vecteur $LV(T_{a[AX]})$ soit à jour par rapport aux ajustements faits dans A_i et que lorsque l'on a finit le traitement de i, le vecteur $LV(T_{i[GD]})$ soit à jour par rapport aux ajustements faits dans G_i et D_i. Ainsi, à la fin de ce parcours, l'ensemble des vecteurs de type $LV(T_{i[GD]})$ sont à jour. Cela permet d'effectuer un nouvel

ajustement des longueurs de branches en utilisant directement le second parcours récursif (le premier devient inutile puisque les vecteurs $LV(T_{i[GD]})$ sont déjà disponibles).

Lors du second parcours récursif, le traitement d'un nœud *i* se fait de la manière suivante :

1. utiliser $LV(T_{a[AX]})$ et $LV(T_{i[GD]})$ pour ajuster la longueur de la branche (*a*, *i*)
2. mettre à jour le vecteur $LV(T_{i[AD]})$ puis traiter le fils gauche de *i* (1^{er} appel récursif)
3. mettre à jour le vecteur $LV(T_{i[AG]})$ puis traiter le fils droit de *i* (2^{nd} appel récursif)
4. mettre à jour le vecteur $LV(T_{i[GD]})$.

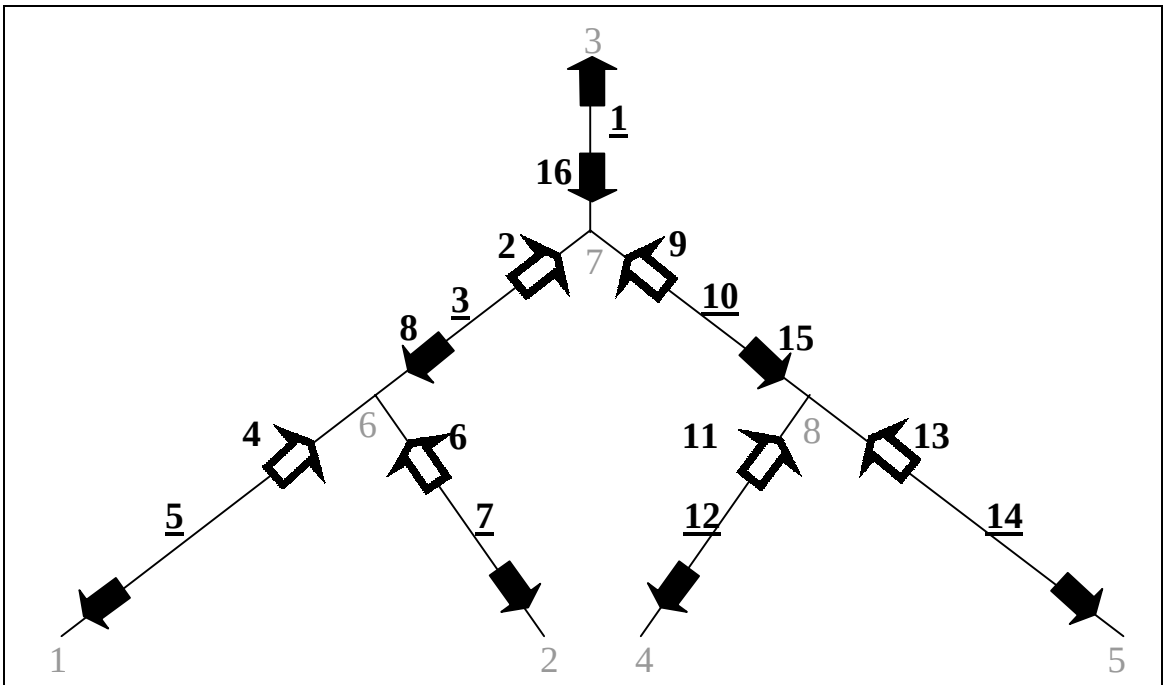


Figure 13 : ajustement des longueurs de branches

Sur cette figure, chaque flèche représente le vecteur de vraisemblance du sous-arbre vers lequel elle pointe. Les flèches entièrement noires correspondent aux vecteurs de vraisemblance disponibles au début du parcours récursif. Les chiffres en gras indiquent l'ordre dans lequel sont effectuées les mises à jours des vecteurs de vraisemblance et les ajustements des longueurs de branches. Les numéros d'étapes qui sont soulignés sont ceux qui correspondent à l'ajustement des longueurs de branches.

La Figure 13, illustre ce second parcours. Les flèches entièrement noires correspondent aux vecteurs de vraisemblance dont on dispose en début de parcours. A la fin de ce parcours,

ces vecteurs de vraisemblance ont été mis à jours de manière à prendre en compte l'ensemble les nouvelles longueurs de branches. Les chiffres en gras indiquent l'ordre dans lequel sont effectuées les mises à jours des vecteurs de vraisemblance et les ajustements des longueurs de branches. Les numéros d'étapes qui sont soulignés sont ceux qui correspondent à l'ajustement des longueurs de branches.

Ainsi, après ce parcours de l'arbre *T*, l'ensemble des longueurs de branches de *T* a été ajusté et chaque ajustement s'est fait de manière cohérente avec les ajustements précédents. De plus, l'ensemble des vecteurs de types $LV(T_{i[GD]})$ sont à jours, ce qui permet d'effectuer ensuite directement une nouvelle fois ce parcours si le processus n'a pas encore convergé.

2.4.4 Recherche de l'arbre de vraisemblance maximale.

Les méthodes de maximum de vraisemblance suivent généralement un processus d'insertion. Le double parcours récursif permet, là encore, de réduire les temps de calculs. Dans un premier temps, on calcule l'ensemble des vecteurs de vraisemblance de l'arbre *T* grâce au double parcours récursif. Pour tester rapidement l'insertion du taxon *k* sur la branche (*i*, *j*), on considère le nœud interne *a* qui est relié à *i*, *j* et *k* et on ajuste les longueurs δ_{ai}, δ_{aj} et δ_{ak} de manière à optimiser localement la vraisemblance de *T* calculée grâce à l'équation ci-dessous :

$$L(S_a^s = b;T) = \prod_{x\in\{i,j,k\}} \sum_{c\in\{A,C,G,T\}} P_{bc}(\delta_{ax}) L(S_x^s = c;T_x) \qquad \textbf{(25)}$$

On optimise ainsi la vraisemblance de l'arbre *T*, uniquement de manière locale sans remettre en cause les estimations des autres longueurs de branches. Contrairement au cas du calcul de la parcimonie, où l'on obtient la valeur exacte, on n'a ici qu'une approximation de la vraisemblance recherchée. Cette approximation peut cependant s'avérer suffisante pour choisir la branche sur laquelle on insère *k*, si l'on optimise ensuite de manière globale la vraisemblance de l'arbre obtenu et que l'on teste ses ré-arrangements de type NNI, ce qui correspond aux options par défaut du programme FASTDNAML (Olsen et al. 1994). Outre l'utilisation de cette approximation, le programme FASTDNAML et le programme DNAML, initialement proposé par (Felsenstein 1993), diffèrent essentiellement par la stratégie qu'ils utilisent pour optimiser les longueurs de branches. En effet, DNAML ajuste chaque branche suivant la technique "*EM: expectation maximization*" et chaque optimisation est faite avec une précision importante. Par contre, FASTDNAML ajuste chaque branche en utilisant la méthode de Newton-Raphton, et les optimisations sont faites de manière approchée pour ne pas perdre de temps en effectuant des ajustements très précis qui, de toutes façons seront en partie invalidés par les suivants. Ces deux modifications permettent à FASTDNAML de reconstruire une phylogénie très proche de celle proposée par DNAML (elles sont même souvent identiques) en un temps beaucoup plus court.

2.5 Conclusion

Dans ce chapitre nous avons décrit les trois grandes familles de méthodes de reconstruction phylogénétique que sont les méthodes de distances, les méthodes de parcimonie et les méthodes de maximum de vraisemblance.

Nous concluons ce chapitre en comparant les performances de ces différentes méthodes. Nous commençons par expliquer les raisons pour lesquelles il est difficile de comparer les performances des méthodes de reconstruction phylogénétique. Nous présentons ensuite les principaux atouts des méthodes de maximum de vraisemblance et les raisons qui nous ont poussés à développer de nouvelles méthodes de reconstruction phylogénétique.

2.5.1 Difficultés d'une évaluation objective

Il existe plusieurs critères pertinents pour comparer les méthodes de reconstruction phylogénétique. On peut par exemple juger ces méthodes en fonction de la taille maximale des jeux de données qu'elles sont capables de traiter, ou bien en fonction de leurs fondements et garanties théoriques. Cependant le critère primordial est évidemment la capacité qu'ont les méthodes à reconstruire correctement les phylogénies des jeux de données auxquelles les biologistes sont réellement confrontés. Or, pour ces données réelles, on ne connaît pas le modèle selon lequel les séquences ont évolué et l'on ne connaît que rarement la phylogénie réelle. Dans le cas de données réelles on peut également s'interroger sur le lien qui existe entre les performances des méthodes et les étapes qui précèdent la reconstruction phylogénétique, notamment le choix de la partie du génome qui est séquencée et l'alignement des séquences.

De plus, les performances des différentes méthodes dépendent de plusieurs paramètres liés, entre autres, à la forme de l'arbre sous-jacent à la phylogénie, à la vitesse d'évolution des séquences et aux différences qui peuvent exister entre les longueurs des différentes branches de la phylogénie. Il ne suffit donc pas d'avoir des jeux de séquences pour lesquelles on dispose de suffisamment d'information pour comparer les méthodes, il faut également qu'au regard des paramètres qui influencent les performances des méthodes, ces jeux de données soient "représentatifs" de ceux auxquels les biologistes sont réellement confrontés. Evidemment, suivant les sujets sur lesquels ils travaillent, les biologistes ne sont pas confrontés aux même types de jeux de données, et la notion de "représentativité" paraît très subjective. Il faut aussi prendre en compte le fait que les paramètres qui influencent les performances d'une méthode de reconstruction phylogénétique ne sont pas toujours clairement identifiés et qu'ils sont souvent liés les uns aux autres.

Pour toutes ces raisons, il n'existe actuellement pas de "jeux de tests" consensuels, permettant simplement de tester une méthode de reconstruction phylogénétique. Le protocole généralement utilisé consiste à comparer les méthodes sur des jeux de données obtenus par simulation informatique. Si une méthode obtient des résultats satisfaisants sur des jeux de données simulées, elle est généralement mise à la disposition des biologistes

via l'Internet. Parmi toutes les méthodes disponibles, il s'effectue alors une sélection de fait, suivant qu'elles sont ou non plébiscitées par les biologistes.

2.5.2 Performance du maximum de vraisemblance

Bien qu'il soit difficile de comparer les différentes méthodes de reconstruction phylogénétique, il semble possible d'affirmer aujourd'hui que la capacité du maximum de vraisemblance à reconstruire une "bonne phylogénie" est nettement supérieure à celle des méthodes de parcimonie et à celle des méthodes de distances.

La possibilité de définir explicitement un modèle d'évolution, est un atout non seulement pour reconstruire de "bonnes phylogénies" mais également pour mieux comprendre le processus d'évolution des séquences. En effet, la méthode du maximum de vraisemblance fournit non seulement une phylogénie vraisemblable des séquences, mais également des valeurs vraisemblables pour les différents paramètres du modèle d'évolution choisi. On sait par exemple, que le taux de nucléotides G-C d'une séquence est relié à sa thermophylie (*i.e.* sa capacité à supporter de forte chaleur). En utilisant un modèle d'évolution qui prend en compte ce taux de G-C, Galtier, Tourasse et Gouy (1999) ont remis en cause l'hypothèse selon laquelle les espèces actuelles descendraient d'une espèce thermophyle.

Par ailleurs, on peut espérer qu'il est encore possible d'améliorer les modèles d'évolution existant, ce qui pourrait augmenter les performances des méthodes de vraisemblance. Comme nous l'avons déjà souligné, les données nucléotidiques dont on dispose ne permettent généralement pas d'ajuster correctement un très grand nombre de paramètres. Toute la difficulté consiste donc à modéliser les phénomènes essentiels du processus d'évolution des séquences en utilisant un minimum de paramètres. En ce sens, l'utilisation de la loi gamma, qui permet avec un seul paramètre de représenter le fait que la vitesse d'évolution n'est pas la même pour tout les sites, est un cas idéal. Les travaux de Yang, Goldman et al. (1994) indiquent que l'utilisation de la loi gamma permet d'améliorer sensiblement les performances du maximum de vraisemblance. Evidemment, il est difficile d'utiliser des séquences obtenues par simulation informatique pour comparer les performances de méthodes de maximum de vraisemblance utilisant des modèles différents. Il est donc encore plus difficile de mesurer l'apport d'un nouveau modèle d'évolution, que de comparer les performances de deux méthodes de reconstruction phylogénétique.

2.5.3 Besoin de méthodes intermédiaires

Comme nous venons de le voir dans le paragraphe précédent, il y a plusieurs bonnes raisons pour lesquelles il est souhaitable d'utiliser des méthodes de reconstruction phylogénétique qui intègrent explicitement un modèle d'évolution. Pour les méthodes de maximum de vraisemblance, cette intégration se fait naturellement, mais ces méthodes ne permettent pas de traiter des jeux de données contenant plus de quelques dizaines de séquences. Les méthodes de distances permettent également de prendre en compte un

modèle d'évolution (lors du calcul de la matrice de distances) et permettent de traiter des jeux de données contenant plusieurs milliers de séquences.

Cependant, la capacité des méthodes de distances à reconstruire un arbre ayant une topologie correcte est nettement inférieure à celle des méthodes de maximum de vraisemblance. Les méthodes de distances permettent d'obtenir de manière rapide une histoire évolutive raisonnable d'un ensemble de séquences, alors que les méthodes de maximum de vraisemblance permettent d'obtenir une histoire évolutive beaucoup plus fiable, mais nécessitent un temps de calcul nettement plus important. Elles ne peuvent donc être appliquées que sur des jeux de données contenant relativement peu de séquences. Il est donc capital de disposer de méthodes intermédiaires permettant d'obtenir des résultats plus fiables que ceux des méthodes de distances classiques, en un temps qui reste raisonnable, même pour de grands jeux de données. Dans cette thèse nous étudions deux approches différentes pour essayer de construire des méthodes dont les performances sont intermédiaires entre celles des méthodes de distances et celles des méthodes de maximum de vraisemblance.

Une approche possible pour reconstruire rapidement un arbre ayant une vraisemblance élevée est celle proposée par (Strimmer et Von Haeseler 1996 ; Strimmer, Goldman et Von Haeseler 1997). Ces travaux présentent des méthodes qui cherchent à reconstruire rapidement un arbre ayant une vraisemblance élevée en combinant des arbres obtenus par maximum de vraisemblance sur des sous-ensembles ne contenant que 4 taxons. Les performances de ces méthodes semblaient très satisfaisantes, et le programme Quartet Puzzling qui les implémente s'est rapidement répandu dans la communauté des biologistes moléculaires. Nous avons proposé plusieurs améliorations de ces méthodes (Ranwez et Gascuel 2001a). Et nous avons montré que malgré ces améliorations les performances actuelles de ces méthodes restent décevantes et que, contrairement à ce que pouvaient laisser espérer les premiers résultats obtenus par (Strimmer, Goldman et Von Haeseler 1997), l'utilisation de quadruplets en reconstruction phylogénétique est fondamentalement discutable (Ranwez et Gascuel 2001b). Ces résultats sont décrits dans le chapitre "Améliorations et limites des méthodes de quadruplets", pages 61 à 83, en s'appuyant sur les articles (Ranwez et Gascuel 2001a ; Ranwez et Gascuel 2001b) qui sont fournis en annexe.

Une autre approche possible consiste à améliorer les performances des méthodes de distances. Pour toutes les méthodes de distances, la qualité de l'estimation des distances entre deux taxons (ou deux groupes de taxons) joue un rôle capital. Nous proposons d'améliorer ces estimations en optimisant localement la vraisemblance de triplets de taxons (ou groupes de taxons). Cette approche est décrite dans le chapitre "Méthodes de distances et maximum de vraisemblance", pages 85 à 101, en s'appuyant sur l'article (Ranwez et Gascuel 2002) qui est également fourni en annexe. Elle s'adapte aux méthodes de distances

les plus couramment utilisées et permet d'augmenter sensiblement la fiabilité des phylogénies qu'elles reconstruisent tout en conservant des temps de calcul raisonnable.

Chapitre 3 Améliorations et limites des méthodes de quadruplets

Comme nous l'avons vu, la méthode de maximum de vraisemblance est actuellement la méthode de référence en reconstruction phylogénétique. Cette méthode statistique offre des garanties théoriques solides et reconstruit des phylogénies fiables. Plusieurs améliorations de la version originale (Felsenstein 1981), ont été proposées, pour accélérer cette méthode. L'amélioration la plus significative

est certainement celle, décrite dans le chapitre précédent (§ 2.4.4), de FASTDNAML (Olsen et al. 1994). Malgré tout, ces méthodes de maximum de vraisemblance restent coûteuses en temps de calcul et ne peuvent traiter que des jeux de données de taille faible. Dès que le nombre *n* de taxons traités devient important, il est beaucoup plus rapide d'utiliser le maximum de vraisemblance pour résoudre tous les problèmes portant sur 4 de ces *n* taxons, que de l'utiliser pour résoudre directement le problème sur *n* taxons. En combinant les résultats obtenus par maximum de vraisemblance sur 4 taxons, les méthodes de quadruplets essayent de profiter de la force du maximum de vraisemblance en un temps de calcul raisonnable.

La méthode de quadruplets proposée par (Strimmer et Von Haeseler 1996) semblait particulièrement prometteuse. En effet, lors des premiers tests effectués par ses auteurs, les arbres reconstruits par Quartet Puzzling (QP) étaient presque aussi fiables que ceux reconstruits par DNAML. De plus, ses auteurs ont fait un travail important de développement pour permettre d'utiliser QP avec de très nombreux modèles d'évolution et assurer la compatibilité de QP avec les programmes existants. Pour toutes ces raisons, QP est encore actuellement la méthode de quadruplets la plus utilisée.

Cependant, nos travaux ont permis de mettre en évidence certaines faiblesses de cette méthode. En particulier, QP tend à reconstruire des arbres ayant une topologie particulière. Pour palier ces faiblesses, nous avons proposé plusieurs modifications qui améliorent les performances et les propriétés théoriques de QP ; la méthode ainsi obtenue est appelée Weight Optimization (WO). Grâce à ces améliorations, et à l'augmentation de la puissance des machines, nous avons pu tester de manière plus intensive les performances de ces méthodes de quadruplets et les comparer avec celles obtenues par des méthodes de distances et de maximum de vraisemblance. Ces tests nous ont permis de montrer que ces méthodes de quadruplets sont paradoxalement moins fiables que les méthodes de distances tout en nécessitant un temps de calcul bien plus élevé.

Ce chapitre présente nos travaux sur les méthodes de quadruplet en s'appuyant sur l'article (Ranwez et Gascuel 2001a) qui constitue l'annexe 1 de cette thèse, et sur l'article (Ranwez et Gascuel 2001b) qui constitue l'annexe 2. Dans un premier temps, nous décrivons les principales méthodes de quadruplets existantes. Dans un second temps, nous détaillons plus particulièrement l'algorithme QP (Strimmer et Von Haeseler 1996), et nous mettons en évidence plusieurs faiblesses de cet algorithme. Puis, nous décrivons l'algorithme WO (Ranwez et Gascuel 2001a) en soulignant les différences existantes entre WO et QP. Nous comparons ensuite ces méthodes de quadruplets avec les autres méthodes de reconstruction phylogénétique et nous analysons les limites des méthodes de quadruplets.

3.1 Méthodes de quadruplets

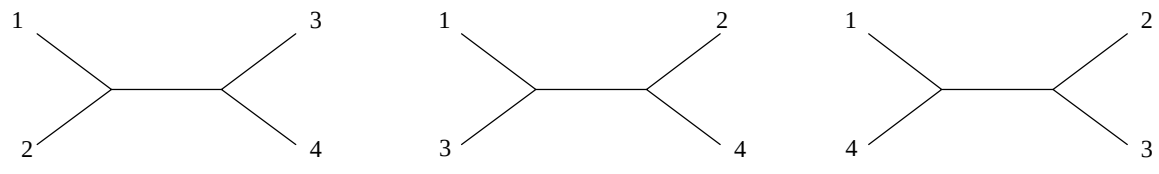
Cette partie décrit les principales méthodes de quadruplets. Ces dernières résolvent le problème initial consistant à inférer la phylogénie d'un ensemble de *n* taxons en combinant

les phylogénies obtenues pour chaque sous-groupe de quatre taxons. Ces groupes de quatre taxons sont appelés quadruplets, et la phylogénie d'un quadruplet est un 4-arbre.

Nous commençons par présenter les avantages pratiques des méthodes de quadruplets. Puis nous détaillons la manière dont le calcul de la vraisemblance se simplifie pour un 4-arbre. Nous présentons ensuite différentes méthodes existantes pour combiner les 4-arbres obtenus en une phylogénie globale. Ces méthodes sont regroupées en fonctions du type d'information qu'elles prennent en compte pour chaque quadruplet.

3.1.1 Avantages des méthodes de quadruplets

Les phylogénies contenant deux ou trois taxons ont toutes la même topologie ; elles ne contiennent que des bipartitions triviales. En utilisant des phylogénies portant sur deux ou trois taxons, on ne dispose d'informations que sur les distances qui les séparent et non sur la manière dont ces taxons se regroupent. Il est donc nécessaire de considérer au moins quatre taxons pour disposer d'informations topologiques. Pour chaque quadruplet {1,2,3,4}, il existe trois 4-arbres différents, et chaque 4-arbre contient une seule bipartition non-triviale qui est induite par la seule arête interne du 4-arbre. On note, 12|34 le 4-arbre dont l'arête interne sépare les taxons 1 et 2 des taxons 3 et 4. La Figure 14 résume ces différentes remarques.



a) 12|34 b) 13|24 c) 14|23

Figure 14 : les trois 4-arbres possibles pour résoudre le quadruplet {1,2,3,4}.

Le fait de chercher à résoudre uniquement des problèmes ne portant que sur quatre taxons permet de disposer d'informations topologiques et comporte plusieurs avantages.

Premièrement, le nombre de topologies différentes pour une phylogénie ayant n feuilles augmente de manière exponentielle en fonction de n. Il est donc, en général, impossible d'examiner toutes les topologies possibles pour le problème initial. Par contre il n'y a que trois 4-arbres possibles pour un quadruplet. Pour un ensemble de n taxons, il est donc assez rapide de considérer tous les 4-arbres possibles, il y en a $3 \times C_4^n$ ($O(n^4)$).

Deuxièmement, certains principes de reconstruction phylogénétique ne peuvent pas être utilisés pour résoudre des problèmes portant sur un grand nombre de taxons, parce qu'ils ont une complexité en temps de calcul ou en espace-mémoire qui est exponentielle. C'est

le cas par exemple du maximum de vraisemblance (Felsenstein 1981) et de la méthode des conjugués d'Hadamar (Hendy, Penny et Steel 1994).

Troisièmement, plus le nombre de séquences étudiées est grand, plus il est probable qu'il y ait des événements d'insertion ou de délétion et donc plus il est probable que le nombre de sites restants après l'alignement soit faible. Dans ce cas, il est possible que les sites restant après l'alignement global des *n* séquences étudiées soient trop peu nombreux pour permettre d'inférer leur phylogénie. Les méthodes de quadruplets permettent en théorie de contourner ce problème. Supposons que l'alignement global de *n* séquences contienne un site s impliquant un événement de types suppression sur la séquence S_1, alors ce site devra être supprimé pour pouvoir reconstruire directement la phylogénie de ces *n* séquences. Par contre, si l'on utilise une approche de quadruplets, ce site *s* pourra être utilisé pour résoudre les quadruplets qui ne contiennent pas S_1. Pour cela, on peut effectuer un alignement en ne considérant que les quatre séquences propres à chaque quadruplet. On risque alors d'obtenir des alignements peu fiables et deux séquences peuvent être alignées différemment pour des quadruplets différents. Il est donc préférable d'effectuer un seul alignement global des *n* séquences puis, de déterminer pour chaque quadruplet les sites qui sont conservés (*i.e.* les sites, qui dans l'alignement global, ne contiennent aucun événement d'insertion ou de suppression pour les quatre séquences considérées). On dispose ainsi de séquences plus longues pour résoudre les quadruplets que pour résoudre directement le problème global. Ce dernier argument est souvent mis en avant pour l'utilisation des méthodes de quadruplets. On peut noter que le même argument pourrait être utilisé pour les méthodes de distances, cependant, à notre connaissance, aucun programme n'utilise ce principe et aucune étude n'a été faite pour mesurer l'impact d'une telle approche sur les performances des méthodes de distances ou de quadruplets.

3.1.2 Vraisemblance d'un 4-arbre

La première étape des méthodes de quadruplets consiste à effectuer la reconstruction phylogénétique d'un jeu de données contenant quatre séquences. Cette étape peut s'effectuer à partir des différentes méthodes de reconstruction phylogénétique décrites dans le chapitre précédent. Il semble cependant préférable d'effectuer cette étape en utilisant le maximum de vraisemblance puisque c'est actuellement la méthode la plus fiable. Dans ce cas simple, il est inutile d'utiliser une heuristique pour rechercher l'arbre de vraisemblance maximale. Il suffit d'évaluer la vraisemblance des trois 4-arbres possibles et de choisir celui ayant la vraisemblance maximale. De plus le calcul de la vraisemblance d'un 4-arbre peut se faire de manière rapide en utilisant explicitement le fait que de nombreux sites ont la même vraisemblance.

Dans un jeu de données, chaque site est associé à un motif qui correspond aux valeurs prises par ce site dans les différentes séquences. Pour une phylogénie donnée, la vraisemblance d'un site dépend directement du motif qui lui est associé, et deux sites ayant le même motif ont la même vraisemblance. Lorsque l'on cherche à déterminer la

vraisemblance d'un arbre *T*, il est intéressant de détecter les sites ayant les mêmes motifs. En effet, l'optimisation des longueurs des branches de *T* nécessite de calculer la vraisemblance de *T* pour différentes longueurs de branches. Durant chacun de ces calculs, lorsque plusieurs sites ont le même motif, on peut calculer la vraisemblance d'un seul de ces sites et utiliser directement ce résultat pour les autres sites ayant le même motif.

Quand on considère un jeu de données ne contenant que quatre séquences, il existe relativement peu de motifs possibles. Chaque site peut prendre quatre valeurs différentes dans chacune des quatre séquences. Il y a donc $4^4 = 256$ motifs possibles. Il faut également noter que des sites ayant des motifs différents peuvent avoir une influence identique sur le calcul de la vraisemblance. Pour accélérer le calcul de la vraisemblance des 4-arbres, on peut donc regrouper les sites ayant la même vraisemblance pour tous les arbres valués, et définir ainsi des classes d'équivalence de sites. Par exemple le modèle de Jukes et Cantor ne fait aucune différence entre les différents nucléotides et les différents types de substitutions. La seule information pertinente est donc de savoir si les nucléotides sont ou non identiques. Pour ce modèle on a donc les 15 classes de sites qui sont résumées dans le tableau ci-dessous :

nombres de nucléotides	1	2		3		4
motifs possibles	XXXX	XXXY	XXYY	XXYZ	XYZX	XYZT
		XXYX	XYXY	XYXZ	YXZX	
		XYXX	YXXY	YXXZ	YZXX	
		YXXX				

Tableau 1 : Les différents motifs possibles pour le modèle de Jukes et Cantor

Dans le cas du modèle à trois paramètres de Kimura (K3ST), lorsque l'on considère des quadruplets, on peut regrouper les sites en 64 classes distinctes (Hendy, Penny et Steel 1994). La formule classique définit la vraisemblance totale d'un arbre *T* comme étant le produit des vraisemblances de chacun des *l* sites (équation (20) page 51). En notant C_1,C_2 … C_f (avec $f \leq 64$ pour le modèle K3ST) les classes effectivement représentées dans un jeu de données de quatre séquences, et en notant s_i un site de la classe C_i et n_i le nombre de sites présents dans la classe C_i, on peut ré-écrire la vraisemblance de l'arbre *T* comme étant

$$L(T) = \prod_{i=i}^{f} \left[\sum_{b \in \{A,C,G,T\}} \pi_b L(S_a^{S_i} = b; T) \right]^{n_i} \qquad \textbf{(26)}$$

L'optimisation de la vraisemblance des trois 4-arbres résolvant un même quadruplet, se fait donc en général en deux étapes. Dans un premier temps, on détermine pour ce quadruplet

le nombre de sites contenus dans chacune des classes possibles. Le nombre de classes possibles est limité à 256 et dépend du modèle d'évolution choisi. Dans un second temps, on utilise la formule (26) (page 65) pour optimiser la vraisemblance des trois 4-arbres correspondant à ce quadruplet. La longueur des séquences étudiées n'intervient donc que dans la première étape. Lors de l'optimisation de la vraisemblance d'un 4-arbre T, le temps requis pour chaque estimation de $L(T)$ est proportionnel au nombre de classes de sites et non à la longueur des séquences. L'utilisation de classes de sites permet ainsi d'accélérer l'optimisation de la vraisemblance d'un 4-arbre. Cette amélioration est d'autant plus importante que la différence entre la longueur des séquences et le nombre de classes possibles (256 pour le modèle GTR) est grande.

3.1.3 Combiner les 4-arbres

Une fois que l'on a résolu l'ensemble des quadruplets, on cherche à reconstruire la phylogénie qui correspond le mieux à l'ensemble Q des solutions obtenues. Dans le cas le plus simple, cet ensemble contient un et un seul 4-arbre pour chacun des quadruplets du jeu de données initial, on dit alors que cet ensemble est complet. Les 4-arbres de cet ensemble ne sont pas indépendants, certaines méthodes essayent donc de vérifier la cohérence de cet ensemble et de corriger certains 4-arbres qui sont visiblement en contradiction avec le reste de l'ensemble Q. Cette approche simple ne permet pas de distinguer le cas où un 4-arbre est une solution presque certaine des cas où deux (voire trois) des 4-arbres sont des solutions possibles entre lesquelles on hésite.

Une première manière d'intégrer ce type d'information consiste à ne prendre en compte que les 4-arbres que l'on estime fiables. On obtient ainsi un ensemble Q incomplet qui peut ne pas contenir de solution pour certains quadruplets difficiles à résoudre. Une autre approche possible consiste à permettre que Q contienne plusieurs 4-arbres pour un même quadruplet. Ainsi, pour chaque quadruplet, Q contient un 4-arbre qui résout ce quadruplet de manière fiable ou l'ensemble des 4-arbres entre lesquels on hésite. Ces deux approches peuvent être vues comme des cas particuliers de l'approche générale dans laquelle, pour chaque quadruplet, Q contient les trois 4-arbres possibles ainsi qu'une pondération pour chacun de ces 4-arbres indiquant la fiabilité de cette résolution.

3.1.3.1 Ensemble complet de 4-arbres

Une phylogénie T induit, au plus, un 4-arbre pour chaque quadruplet. L'arête interne de chacun de ces 4-arbres correspond à un unique chemin de T, et l'on note Q_T cet ensemble de 4-arbres. Si cette phylogénie T est complètement résolue, elle induit exactement un 4-arbre par quadruplet (Figure 15 page 67). Sinon, elle possède des nœuds internes ayant un degré supérieur ou égal à quatre ; il existe donc des quadruplets obtenus en prenant un taxon dans quatre sous-arbres distincts reliés à un même nœud. Ces quadruplets ne sont pas résolus par cette phylogénie puisque, pour ces quadruplets, l'arête interne du 4-arbre induit se réduit à un seul nœud. On dit qu'un ensemble Q de 4-arbres est arboré si et seulement il

existe une phylogénie T telle que $Q_T = Q$. Une manière possible d'inférer une phylogénie à partir d'un ensemble Q de 4-arbres consiste à chercher la phylogénie T* correspondant au plus grand sous-ensemble arboré contenu dans Q. Cet ensemble arboré maximal est unique (Bandelt et Dress 1986). La phylogénie correspondante à cet ensemble peut être reconstruite en $O(n^4)$ en utilisant l'algorithme IQ^* (Berry et Gascuel 2000) sur lequel nous reviendrons dans la partie 3.1.3.2. On souhaite parfois augmenter la résolution de T^* en y intégrant des arêtes supplémentaires vérifiant des contraintes moins fortes que celles initialement présentes dans T^*.

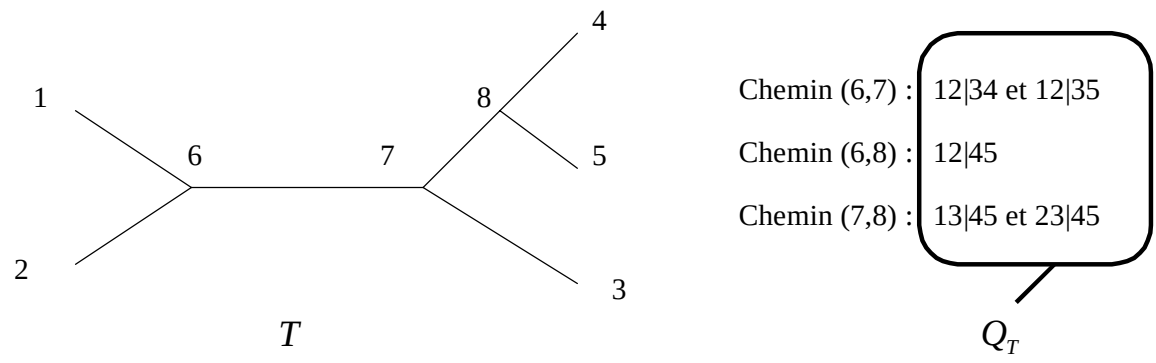


Figure 15 : une phylogénie T induit un ensemble de 4-arbres noté Q_T.

Une approche possible consiste à construire l'arbre dont les bipartitions respectent le critère local proposé dans (Berry et al. 1999). Une bipartition séparant les taxons en deux sous-groupes notés A et B induit des 4-arbres de type $a_1a_2|b_1b_2$ avec $a_1 \in A$, $a_2 \in A$, $b_1 \in B$ et $b_2 \in B$. L'approche locale de Quartet Cleaning (Berry et al. 1999) accepte une telle bipartition si le nombre de 4-arbres qu'elle induit, ne se trouvant pas dans l'ensemble Q, est inférieur à $(|A|-1)(|B|-1)/4$. Les bipartitions respectant ce critère sont compatibles en une phylogénie qui est un raffinement de la phylogénie T^*.

Une autre manière d'obtenir une phylogénie plus résolue que T^*est d'effectuer un pré-traitement sur l'ensemble Q, de manière à proposer une autre résolution pour les quadruplets dont la solution initialement proposée est en contradiction évidente avec les autres 4-arbres de l'ensemble Q. Cette idée fut introduite par (Jiang 1998) et reprise par (Berry et al. 1999). En effet, quand l'algorithme local de Quartet Cleaning conserve les bipartitions $A|B$ qui induisent un nombre de 4-arbres ne se trouvant pas dans l'ensemble Q inférieur à $(|A|-1)(|B|-1)/4$, cela revient finalement à modifier la résolution de ces 4-arbres dans l'ensemble Q et à construire ensuite l'arbre T^* pour ce nouvel ensemble de 4-arbres. Cette idée est reprise par (Willson 1999) qui propose une approche différente pour corriger les 4-arbres. Dans cette approche, un entier est associé à chaque 4-arbre pour

mesurer le soutien que ce 4-arbre reçoit des 4-arbres présents dans l'ensemble *Q*. Cet ensemble est ensuite modifié de manière à contenir pour chaque quadruplet celui des trois 4-arbres qui a le soutien maximum. Un 5-arbre *T* induit cinq 4-arbres distincts, on dit que *T* est soutenu par l'ensemble *Q*, si $Q_T \cap Q \geq 4$. Si *T* est un 5-arbre soutenu par *Q*, soit $Q_T \cap Q = 5$, dans ce cas *T* soutient chacun des 4-arbres qu'il induit, soit $Q_T \cap Q = 4$, et *T* soutient uniquement le 4-arbre de Q_T qui n'est pas dans *Q*. Le soutien d'un 4-arbre correspond au nombre de 5-arbres qui le soutiennent. Ces deux approches de *quartet cleaning* ont l'inconvénient d'avoir une complexité en temps de calcul en $O(n^5)$, alors que la majorité des méthodes de quadruplets a une complexité en $O(n^4)$.

Si l'on souhaite obtenir une phylogénie complètement résolue, il faut chercher parmi les phylogénies complètement résolues celle qui induit le plus possible de 4-arbres présents dans l'ensemble *Q*. On cherche donc la phylogénie *T* telle que l'intersection de Q_T et de *Q* soit maximale. Trouver la phylogénie *T* qui optimise ce critère est un problème NP-difficile (Steel 1992). Il est donc nécessaire de recourir à des heuristiques pour chercher la phylogénie qui satisfait un maximum de 4-arbres de l'ensemble *Q*. On peut pour cela utiliser l'algorithme AddTree (Sattath et Tversky 1977). Dans cette approche, la distance entre deux taxons *a* et *b* correspond au nombre de 4-arbres de type *ax|by* présents dans l'ensemble *Q*. On peut également utiliser des algorithmes tels que Quartet Puzzling (Strimmer et Von Haeseler 1996) ou Weight Optimization (Ranwez et Gascuel 2001b) qui sont détaillés dans la suite de ce chapitre.

3.1.3.2 Ensemble incomplet de 4-arbres

L'approche précédente oblige à proposer un 4-arbre pour chaque quadruplet, même lorsque la méthode utilisée pour le résoudre indique clairement qu'il n'est pas possible d'affirmer quelle est la bonne topologie pour ce quadruplet. Une alternative possible consiste à utiliser un ensemble incomplet de 4-arbres dont sont exclus les 4-arbres les moins fiables.

Comme dans le cas d'un ensemble complet, on peut rechercher la phylogénie T^* correspondant au plus grand sous-ensemble arboré contenu dans *Q*. Les algorithmes qui utilisent cette approche peuvent bien sûr être utilisés dans le cas particulier où *Q* est complet. En particulier, la phylogénie T^*, peut être reconstruite en $O(n^4)$ en utilisant l'algorithme IQ^* (Berry et Gascuel 2000). Cet algorithme exploite le principe du double parcours récursif décrit dans le chapitre précédent (§ 2.3.3) et dans (Ranwez et Gascuel 2001, Pp. 91-93). L'algorithme ADDQUAD (Berry 1997, p. 166) permet d'augmenter la résolution de T^* en y intégrant des arêtes supplémentaires vérifiant des contraintes moins strictes. On considère initialement la phylogénie T^*. A chaque étape, deux sous-arbres connectés à un nœud irrésolu sont agglomérés, créant ainsi une nouvelle branche qui induit de nouveaux 4-arbres. L'agglomération est choisie en fonction d'un critère local qui favorise les agglomérations contredisant un minimum de 4-arbres présents dans Q. Lorsque plusieurs agglomérations sont équivalentes au sens de ce critère, on retient celle qui induit le plus de 4-arbres présents dans *Q*. On continue à réaliser des agglomérations tant que cela

diminue la valeur d'un critère global mesurant l'adéquation entre la phylogénie T obtenue et l'ensemble Q. Ce critère est défini à partir de l'ensemble Q_T et de l'ensemble Q. Il correspond au nombre de 4-arbres présents dans un seul de ces deux ensembles.

Une autre approche pour traiter des ensembles incomplets consiste à utiliser la propriété *d'arbre-consistance* qui est moins stricte que la propriété d'arboricité. On dit qu'un ensemble Q de 4-arbres est arbre-consistant si, et seulement si, il existe une phylogénie T telle que $Q \subseteq Q_T$. Une manière possible d'inférer une phylogénie à partir d'un ensemble (éventuellement incomplet) Q de 4-arbres consiste à chercher la phylogénie T^* correspondant au plus grand sous-ensemble arbre-consistant qui est inclus dans Q. Il est à noter que cet ensemble maximal n'est pas nécessairement unique et que, dans le cas général, trouver cet ensemble est un problème NP-complet. En fait, déterminer si un ensemble est ou non arbre-consistant, est déjà un problème NP-complet (Steel 1992).

Une approche possible pour déterminer si un ensemble Q est arbre-consistant, et trouver la phylogénie correspondante, consiste à essayer de se ramener au cas de l'arboricité en déduisant les solutions des quadruplets non-résolus à partir des 4-arbres présents dans l'ensemble Q. En effet, si une phylogénie induit certains ensembles de 4-arbres on peut en déduire qu'elle induit également d'autres 4-arbres et donc compléter l'ensemble Q. Par exemple, si une phylogénie contient 12|34 et 13|45 elle contient forcément 12|35, 12|45 et 23|45. En effet, si l'on suit un processus d'insertion à partir de la phylogénie irrésolue contenant les taxons 1, 2 et 3, pour satisfaire 12|34 il faut insérer 4 sur la branche reliée au nœud 3, puis pour satisfaire 13|45 il faut insérer le nœud 5 sur la branche de 12|34 qui est relié au nœud 4. La phylogénie ainsi obtenue correspond à la phylogénie T de la Figure 15 (page 67) dont l'ensemble Q_T contient également 12|35, 12|45 et 23|45. On peut ainsi définir des règles d'inférences permettant de compléter l'ensemble Q. L'ordre d'une règle correspond au nombre de 4-arbres nécessaires pour en inférer de nouveaux, dans l'exemple ci-dessus on utilise donc des règles d'ordre deux (ou règles dyadiques) puisqu'elles s'appuient sur la présence de deux 4-arbres (12|34 et 13|45). Une fois que l'on a déduit de nouveaux 4-arbres, on peut les utiliser pour en inférer d'autres. Cependant, même si on applique plusieurs fois ces règles, cela ne garantit pas que l'on infère tous les 4-arbres qui peuvent être déduits à partir de l'ensemble Q. En effet, (Bryant et Steel 1995) ont montré que pour tout entier r, il existe des inférences d'ordre r qui ne peuvent pas être obtenues en appliquant plusieurs fois des inférences d'ordre inférieur.

Cependant, Erdös et al. (1997b) ont exhibé des cas particuliers utiles, où l'application des règles dyadiques permet d'inférer suffisamment de 4-arbres pour obtenir un ensemble complet. Ils définissent pour cela la notion de 4-arbre local. Un 4-arbre est local pour la phylogénie T lorsque l'arête interne de ce 4-arbre correspond à un chemin de T réduit à une seule branche. Si l'on dispose d'un ensemble Q de 4-arbres qui correspondent aux 4-arbres locaux d'une phylogénie, alors l'application des règles dyadiques suffit à compléter l'ensemble Q (Erdös et al. 1997a). Ils montrent ensuite qu'il suffit que Q contienne un 4-

arbre local pour chaque branche de *T*, et proposent donc de choisir un 4-arbre qui corresponde à une solution fiable d'un quadruplet. Pour cela, ils estiment la difficulté a priori d'un quadruplet à partir de la dissimilarité maximale qui existe entre deux des quatre séquences (i.e. le nombre de sites où les deux séquences diffèrent). Puisqu'on ne sait pas a priori quels sont les 4-arbres locaux, ils utilisent l'approche incrémentale suivante :

- pour chaque niveau de difficulté *k*, ne conserver dans *Q* que les 4-arbres de difficulté inférieure à *k* et compléter *Q* grâce aux règles dyadiques,
    - si l'ensemble ainsi obtenu est arboré, renvoyer la phylogénie correspondante,
    - sinon passer au niveau de difficulté suivant.

Même dans le pire des cas, cette méthode de "short quartet" a une complexité en temps qui reste proche de $O(n^4)$. Ses temps de calculs sont donc comparables à ceux des méthodes en $O(n^4)$. De plus, en s'appuyant sur un modèle simple d'évolution des séquences, Erdös et al. (1997a) montrent que leur approche garantit, en théorie, que l'arbre reconstruit est correct dès que les séquences considérées sont suffisamment longues. NJ offre également cette garantie théorique, mais pour des séquences plus longues.

3.1.3.3 4-arbres valués

Dans le cas valué, l'ensemble *Q* contient tous les 4-arbres possibles et une valeur numérique est associée à chacun d'entre-eux. Le couple (*q*, *w*), composé d'un 4-arbre *q* et de sa pondération *w*, est appelé *w*4-arbre. La pondération des *w*4-arbres est faite de manière à associer à chaque 4-arbre un poids *w* d'autant plus grand qu'il est probable que ce 4-arbre représente la topologie correcte. La somme des poids associés aux 4-arbres induits par une phylogénie *T*, constitue une mesure naturelle de l'adéquation entre *T* et *Q*. Dans le cas d'un ensemble de 4-arbres valués, on cherche donc la phylogénie *T* qui maximise le critère *W* défini par :

$$W(T) = \sum_{q \in Q_T \text{ et } (q,w) \in Q} w \qquad \textbf{(27)}$$

Ce cas d'un ensemble de 4-arbres valués généralise les cas précédents. On retrouve les cas précédents en utilisant uniquement les poids 0, 1 et $-\infty$ (en fait, $-3 \times C_4^n$ est suffisant). Le critère *W* associé au *n*-arbre complètement irrésolu vaut 0. De plus, pour tout *n*-arbre *T* qui induit au moins un 4-arbre pondéré par $-\infty$, on a $W(T) < 0$. Suivant ces pondérations, optimiser *W* revient donc à chercher l'arbre qui induit un maximum de 4-arbre pondéré par 1 sans induire de 4-arbre pondéré par $-\infty$. On note Q_1 l'ensemble des 4-arbres de *Q* ayant le poids 1. Si pour chaque quadruplet un et un seul 4-arbre est pondéré par la valeur 1 et les autres sont pondérés par $-\infty$, alors Q_1 est un ensemble complet. Dans ce cas, optimiser *W* revient à chercher la phylogénie *T* correspondant au plus grand sous-ensemble arboré de Q_1. Si l'on modifie ce système de pondération en autorisant que pour certains quadruplets

les trois 4-arbres aient une pondération valant 0, alors Q_1 est un ensemble pouvant être incomplet. Dans ce cas optimiser *W* revient à chercher la phylogénie *T* correspondant au plus grand sous-ensemble arbre-consistant de Q_1.

Dans le cas général, trouver la phylogénie *T* qui optimise le critère *W* est un problème NP-complet (Steel 1992). Il est donc nécessaire de recourir à des heuristiques pour trouver une phylogénie qui optimise correctement *W*. Les méthodes de quadruplets basées sur un ensemble de *w*4-arbres, nécessitent donc de définir un système de pondération des 4-arbres ainsi qu'une heuristique pour optimiser le critère *W*.

(Willson 1999) propose d'inférer les 4-arbres suivant une approche de parcimonie. Pour chaque quadruplet, il calcule donc les valeurs v_1, v_2, et v_3 des parcimonies des 4-arbres q_1, q_2 et q_3 associés à ce quadruplet, et associe au 4-arbre q_i la pondération :

$$w_i = \min\left(\{v_1, v_2, v_3\} - \{v_i\}\right) - v_i .$$

Par exemple, si les 4-arbres *ab*|*cd*, *ac*|*bd* et *ad*|*bc* ont des valeurs de parcimonie valant respectivement 20, 14 et 25, l'ensemble *Q* contiendra (*ab*|*cd*, -6), (*ac*|*bd*, 6) et (*ad*|*bc*, -11). Ainsi, pour chaque quadruplet, le 4-arbre le plus parcimonieux est le seul qui soit pondéré positivement, et un poids fortement positif indique qu'il existe une différence importante entre la parcimonie de ce 4-arbre et celles des deux autres. Une phylogénie est ensuite construite à partir de cet ensemble de *w*4-arbres en suivant un processus d'insertion visant à optimiser le critère *W*.

Pour les raisons évoquées en conclusion du premier chapitre, il semble préférable d'utiliser une méthode de maximum de vraisemblance pour résoudre les quadruplets. C'est l'approche adoptée par quartet puzzling (Strimmer, Goldman et Von Haeseler 1997), et c'est certainement une des raisons principales qui explique l'intérêt qu'à suscité cette méthode.

3.2 Quartet Puzzling (QP)

Ce paragraphe détaille l'algorithme QP introduit par (Strimmer et Von Haeseler 1996 ; Strimmer, Goldman et Von Haeseler 1997). Dans un premier temps, nous présentons la manière dont les 4-arbres sont pondérés, puis nous décrivons l'heuristique utilisée par QP pour inférer une phylogénie à partir de ces 4-arbres valués. Cette heuristique suit un processus d'insertions, et l'arbre reconstruit dépend très fortement de l'ordre dans lequel les taxons sont insérés. C'est pour cette raison que QP reconstruit plusieurs arbres suivant des ordres d'insertions différents et qu'il propose comme résultat final l'arbre consensus des arbres reconstruits pour les différents ordres.

3.2.1 Pondération des 4-arbres

La première version de QP utilise un ensemble complet de 4-arbres (Strimmer et Von Haeseler 1996). Pour chaque quadruplet, seul le 4-arbre le plus vraisemblable est conservé.

Une version ultérieure de cet algorithme (Strimmer, Goldman et Von Haeseler 1997) utilise des 4-arbres qui sont pondérés de la manière suivante. Pour chaque quadruplet, il existe trois 4-arbres q_1, q_2 et q_3. On note leur vraisemblance respective l_1, l_2 et l_3. La probabilité p_i que la topologie correcte soit celle de q_i peut être estimée par l'équation suivante, grâce au théorème de Bayes :

$$p_i = \frac{l_i}{l_1 + l_2 + l_3}$$

A partir de ces trois probabilités, Strimmer, Goldman et al.(1997) proposent trois systèmes de pondération des 4-arbres q_1, q_2 et q_3. Suivant le système adopté, les 4-arbres sont pondérés de manière continue, discrète ou binaire. Dans le cas continu, $w_i = p_i$. Dans le cas binaire, le 4-arbre le plus probable reçoit un poids de 1 et les deux autres un poids de 0. Dans le cas discret, les poids sont des approximations discrètes, au sens des moindres carrés, des p_i. En supposant que $p_1 \geq p_2 \geq p_3$, les poids (w_1, w_2, w_3) sont estimés par (1, 0, 0), (1/2, 1/2, 0) ou (1/3, 1/3, 1/3) en mesurant l'écart aux moindres carrés entre (p_1, p_2, p_3) et cette nouvelle représentation.

3.2.2 Construction de phylogénies à partir des *w*4-arbres

Lors de l'étape suivante, ces *w*4-arbres sont utilisés pour inférer un ensemble de *n*-arbres qui sont des solutions possibles pour la phylogénie des *n* taxons étudiés. Chacun de ces *n*-arbres est construit suivant un processus d'insertion de la manière suivante. L'ordre dans lequel les taxons sont insérés est tiré aléatoirement. Supposons, pour simplifier nos notations, qu'il s'agisse de l'ordre (1, 2, 3,…,*n*). Parmi les trois 4-arbres qui résolvent le quadruplet {1,2,3,4}, celui de poids maximal est choisi comme point de départ du processus d'insertion. Lors de l'insertion du taxon *i*, la phylogénie courante T_i contient les taxons de l'ensemble $S_i = \{1,2,3,\ldots,i-1\}$. La branche sur laquelle le taxon *i* est inséré est déterminée de la manière suivante. Une pénalité valant initialement 0 est associée à chaque branche de T_i. Tous les 4-arbres de type (*xi*|*yz, w*) avec *x*, *y*, et *z* appartenant à l'ensemble S_i sont ensuite pris en compte pour mettre à jour ces pénalités. Ces *w*4-arbres sont les seuls qui donnent directement une information sur la position de *i* dans T_i. On dira qu'ils sont *pertinents* pour l'insertion du taxon *i*. On sait par exemple que si le taxon *i* est inséré sur le chemin reliant *y* à *z*, alors le 4-arbre *xi*|*yz* ne sera pas induit par la phylogénie T_{i+1}. Pour prendre en compte l'information d'un *w*4-arbre (*xi*|*yz, w*), l'algorithme QP ajoute donc *w* à la pénalité de chacune des branches qui se trouve sur le chemin reliant *y* à *z*. Une fois que tous les 4-arbres pertinents pour l'insertion du taxon *i* ont été pris en compte, QP insère le taxon *i* sur la branche ayant la plus faible pénalité. La Figure 1 de l'article (Ranwez et Gascuel 2001b, p. 1105) illustre la manière dont QP utilise les *w*4-arbres pertinents pour sélectionner la branche du 4-arbre 12|34 sur laquelle il insère le taxon 5.

Ce processus d'insertion est légèrement modifié en fonction du système qui est utilisé pour pondérer les 4-arbres. Dans le cas d'une pondération binaire, seuls les *w*4-arbres de type

(*xi*|*yz*, 1) sont pris en compte, car les poids nuls ne modifient la pénalité d'aucune branche. Dans le cas discret, pour chaque quadruplet un 4-arbre est tiré de manière équiprobable parmi ceux ayant un poids non nul. Le processus d'insertion est ensuite le même que dans le cas binaire. Par exemple, si pour un quadruplet donné les trois pondérations sont (1/2, 1/2, 0), alors certains *n*-arbres (la moitié en moyenne) seront reconstruits en supposant que la première topologie est la bonne pour ce quadruplet, et d'autres le seront en supposant que c'est la deuxième topologie qui est correcte.

3.2.3 Consensus

Le *n*-arbre reconstruit par QP dépend de l'ordre (aléatoire) dans lequel les taxons sont insérés. Dans le cas discret, ce *n*-arbre dépend également des choix qui sont faits (aléatoirement) pour chaque quadruplet. Pour s'affranchir de ce problème, et s'assurer que la phylogénie proposée n'est pas uniquement due à ces choix aléatoires, QP définit plusieurs ordres aléatoires et reconstruit les phylogénies correspondantes. Il renvoie ensuite le consensus de ces phylogénies comme résultat final. Un arbre consensus *T* pour un ensemble d'arbres *S*, est l'arbre qui au sens d'une certaine mesure représente le mieux les arbres de l'ensemble *S*. Il existe plusieurs définitions de l'arbre consensus, qui varient en fonction de la mesure adoptée. Le consensus utilisé dans QP, est le consensus majoritaire de Margush et McMorris (1981). Dans ce cas, l'arbre consensus contient uniquement les branches (les bipartitions) qui apparaissent dans plus de la moitié des arbres de l'ensemble *S*. Il existe toujours un arbre *T* "égal" à l'ensemble de ces bipartitions. Cet arbre minimise :

$$\sum_{T_S \in S} d(T, T_S) \qquad \textbf{(28)}$$

et correspond donc au barycentre de l'ensemble *S* pour la distance de Robinson & Fould.

L'arbre ainsi obtenu est rarement complètement résolu ce qui complique la comparaison de QP avec les méthodes les plus courantes de reconstruction phylogénétique qui infèrent des arbres complètement résolus. Pour cette raison, les tests visant à évaluer QP (Strimmer et Von Haeseler 1996 ; Strimmer, Goldman et Von Haeseler 1997) sont faits en utilisant l'arbre consensus tel qu'il est codé dans le programme CONSENSE du package PHYLIP (communication personnelle). Ce programme utilise une heuristique gloutonne pour chercher l'arbre *T* complètement résolu qui minimise l'équation (28). Pour cela, il construit de manière incrémentale un ensemble *B*, de bipartitions compatibles en un arbre. Initialement l'ensemble *B* est vide. Les bipartitions présentes dans au moins un arbre de l'ensemble *S* sont ordonnées de manière décroissante en fonction du nombre d'arbres de *S* qui les contiennent. Elles sont ensuite considérées successivement suivant cet ordre. Lors du traitement d'une bipartition, si celle-ci est compatible avec les bipartitions déjà présentes dans *B*, alors elle est ajoutée à cet ensemble. L'arbre consensus renvoyé est l'arbre correspondant à l'ensemble *B* ainsi construit. Dans la suite, on appelle cet arbre le consensus glouton de *S*. Toutes les bipartitions présentes dans plus de 50% des arbres de *S*

sont compatibles en un arbre. Elles sont donc toutes présentes dans *S*, et le consensus glouton est donc un raffinement du consensus majoritaire. En théorie le consensus glouton n'est pas forcément un arbre complètement résolu. En pratique nous avons toujours obtenu un arbre complètement résolu lorsque nous avons utilisé ce consensus sur les 1000 arbres reconstruits par QP.

3.3 Faiblesses de Quartet Puzzling

Nos travaux ont mis en évidence certaines faiblesses de l'algorithmes QP (Ranwez et Gascuel 2001a). Les différents problèmes que pose l'utilisation de QP sont résumés dans ce paragraphe et ont servi de base de réflexion pour proposer un algorithme de quadruplets inspiré de QP mais corrigeant ses défauts. Cet algorithme possède de meilleures propriétés théoriques que QP et obtient de meilleurs résultats sur des jeux de tests utilisés plusieurs fois par la communauté scientifique pour valider des méthodes de reconstruction phylogénétique.

3.3.1 Un critère d'insertion perfectible

Lorsqu'un 4-arbre *xi*|*yz* n'est pas induit par un arbre *T*, on dit que *T* et *xi*|*yz* sont incompatibles. Pour déterminer la branche sur laquelle insérer le taxon *i*, QP prend en compte le *w*4-arbre (*xi*|*yz*, *w*) en ajoutant une pénalité *w* à l'ensemble des branches situées sur le chemin reliant *y* à *z*. Certes, si le taxon *i* est inséré sur une de ces branches, l'arbre T_{i+1} est incompatible avec le 4-arbre *xi*|*yz*. Cependant, il existe d'autres branches sur lesquelles l'insertion de *i* produirait un arbre incompatible avec *xi*|*yz* et qui ne sont pas pénalisées par QP. Considérons par exemple que l'arbre courant est le 4-arbre 12|34 sur lequel on cherche à insérer le taxon 5. Pour prendre en compte le *w*4-arbre (45|23, *w*), QP ajoute la pénalité *w* aux branches du chemin reliant le taxon 2 au taxon 3, mais il ne pénalise pas la branche qui est reliée au taxon 1. Pourtant, si l'on insère le taxon 5 sur cette branche, on obtient le 5-arbre *T* de la Figure 16 qui est, lui aussi, incompatible avec 45|23 (c'est le 4-arbre 25|34 qui est présent dans l'ensemble Q_T).

On dit d'un quadruplet qu'il est *bien orienté* si celui des trois 4-arbres dont la topologie est correcte est aussi celui qui possède la vraisemblance la plus grande. Une propriété importante pour une méthode de quadruplets qui pondère les 4-arbres à partir de leurs vraisemblances est de garantir qu'elle reconstruit la bonne phylogénie chaque fois que tous les quadruplets sont bien orientés.

Nous avons montré que QP possède cette propriété uniquement dans le cas binaire. Dans le cas discret, le problème vient de l'étape de discrétisation est n'est pas spécifique à QP. En effet, il est possible que tous les quadruplets soient bien orientés et que tous les 4-arbres soient pondérés par 1/3. Dans ce cas, QP peut reconstruire n'importe quelle phylogénie en fonction des tirages aléatoires qu'il effectue pour choisir la topologie qu'il conserve pour chaque quadruplet. Dans le cas continu, si tous les quadruplets sont bien orientés, alors pour chaque quadruplet le 4-arbre de poids maximal est celui dont la topologie est correcte. Cependant, même dans ce cas, nous avons trouvé un exemple où, bien que tous les quadruplets soient bien orientés, QP ne reconstruit pas la phylogénie correcte. Cet exemple est détaillé dans la Figure 1 de l'article (Ranwez et Gascuel 2001b, p. 1105). Cela est dû au fait que pour chaque *w*4-arbre QP ne pénalise qu'un sous-ensemble des branches qui devraient l'être.

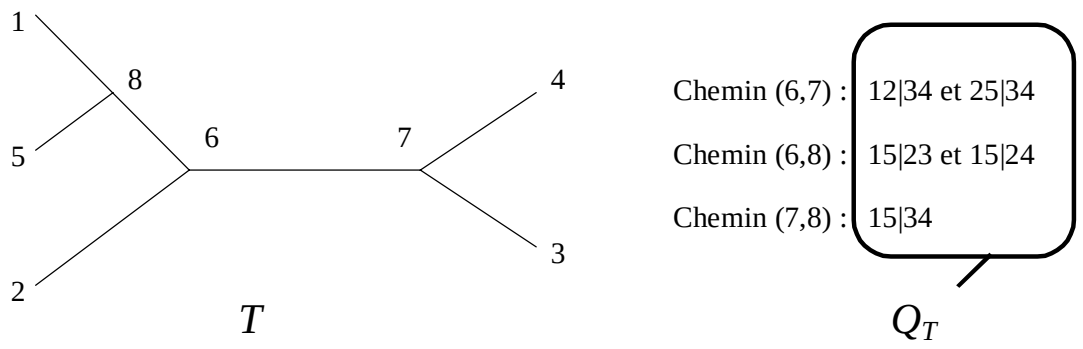


Figure 16 : limite du critère d'insertion de QP.

L'arête du 4-arbre 12|34 reliée au taxon 1 n'est pas pénalisée par QP lorsqu'il prend en compte le *w*4-arbre (45|23, *w*). Pourtant, en insérant le taxon 5 sur cette arête de 12|34, on obtient l'arbre *T* qui est incompatible avec 45|23 (ce 4-arbre n'est pas inclus dans l'ensemble Q_T).

3.3.2 Un biais topologique important

Nous avons également montré que les arbres reconstruits par QP ont un biais topologique important. Nous appelons *x*-bipartition une bipartition qui sépare les taxons étudiés en deux sous-ensembles tels que le plus petit de ces deux sous-ensembles contient exactement *x* taxons. Nous avons montré de façon expérimentale que QP tend à reconstruire des arbres où les 2-bipartitions sont sur-représentées, et que ce biais est dû au fait que l'arbre final proposé par QP est l'arbre consensus d'un ensemble d'arbres complètement résolus. Pour cela, nous avons utilisé la version continue de QP pour reconstruire 1000 arbres à partir d'un jeu de séquences contenant 12 séquences identiques. Ce jeu de séquences ne contenait donc aucun signal phylogénétique. Le nombre moyen de 2-bipartitions présentes dans les

1000 arbres reconstruits par QP était de 4, alors que l'arbre consensus de ces 1000 arbres (obtenu suivant l'algorithme glouton décrit plus haut) en contenait 6 (le maximum possible pour un 12-arbre).

Ce phénomène s'explique assez simplement. Soit S l'ensemble des 1000 n-arbres reconstruits par QP. Pour une valeur $x \geq 2$ donnée, il existe C_x^n x-bipartitions différentes possibles. En l'absence de signal phylogénétique, toutes ces bipartitions apparaissent de manière équiprobable dans S. Si l'on note N_x le nombre de x-bipartitions effectivement présentes dans S, le nombre d'apparitions attendues d'une x-bipartition particulière est N_x / C_x^n. En s'appuyant sur ces remarques, on peut facilement prouver que dans un tel ensemble S et pour une valeur $x > 2$ donnée, le nombre d'apparitions attendues d'une 2-bipartition particulière est supérieur au nombre d'apparitions attendues d'une x-bipartition particulière.

Comme $(C_2^n < C_x^n)$, il suffit en fait de prouver que $N_2 \geq N_x$. Pour cela, on montre que chaque arbre T de S contient au moins autant de 2-bipartitions que de x-bipartitions. Cela est lié au fait que les arbres de S (inférés par QP) sont tous complètement résolus. On distingue deux cas :

1. Soit $x = n/2$, alors T possède une unique x-bipartition. Cette x-bipartition correspond à une branche (n_1, n_2) de T qui divise l'arbre T en deux sous-arbres notés T_1 et T_2 qui ont respectivement n_1 et n_2 pour racine. Chacun de ces deux sous-arbres est complètement résolu et contient donc au moins une paire externe. Ces deux paires externes correspondent à deux 2-bipartitions de T. Dans le cas où $x = n/2$, T possède donc une unique x-bipartition et au moins deux 2-bipartitions.
2. Soit $x < n/2$, dans ce cas, on note m le nombre de x-bipartitions de l'arbre T. Chacune de ces x-bipartitions correspond à une branche de T qui divise cet arbre en deux sous-arbres T_1 et T_2 dont un seul possède exactement x feuilles. L'arbre T contient donc m sous-arbres T_i ayant exactement x feuilles. Ces sous-arbres n'ont aucune branche et aucun nœud en commun. De plus, tout ces sous-arbres sont complètement résolus. Ils contiennent donc tous au moins une paire externe qui correspond à une 2-bipartition de T. Ces m 2-bipartitions de T sont toutes distinctes car les différents sous-arbres T_i n'ont aucune feuille en commun. Ainsi, dans le cas où $x < n/2$, si T possède m x-bipartitions, il possède également au moins m 2-bipartitions.

Ainsi, pour un ensemble S de n-arbres, complètement résolus et reconstruits sans signal phylogénétique, la probabilité a priori d'avoir une 2-bipartition particulière est supérieure à celle d'avoir une x-bipartition particulière. Si l'on construit de manière gloutonne l'arbre consensus de S, les 2-bipartitions seront donc en général les premières conservées. Cela ne prouve rien dans le cas où il y a effectivement un signal phylogénétique dans les données ou dans le cas de l'utilisation du consensus majoritaire. On peut cependant penser que même dans ces cas, le fait que les arbres de S sont complètement résolus influence la

topologie de l'arbre consensus que QP propose comme résultat final. Cela permet de comprendre, de manière intuitive, la raison pour laquelle QP tend à inférer des *n*-arbres possédant un nombre important de 2-bipartitions.

3.3.3 Une complexité élevée

Tel qu'il est décrit par ses auteurs, l'algorithme utilisé par QP pour reconstruire un arbre à partir de l'ensemble des *w*4-arbres a une complexité en $O(n^5)$. La complexité de QP est donc plus élevée que celle de la plupart des méthodes de quadruplets. Cependant, cette complexité en $O(n^5)$ peut paraître secondaire puisque dans le cas de QP, l'étape de pondération des 4-arbres a une complexité encore plus élevée ($O(n^4 l)$).

Néanmoins, l'heuristique de reconstruction d'arbre utilisée par QP peut s'appliquer avec d'autres modes de pondération moins coûteux en temps de calcul. De plus, même dans le cas d'une pondération par maximum de vraisemblance, il est important d'optimiser cette étape de reconstruction car elle est faite de nombreuses fois ce qui augmente considérablement les temps de calcul. Par défaut, la version de QP qui est librement diffusée sur le Web effectue 1000 fois cette étape de construction. Par ailleurs, les auteurs de QP indiquent que plus le nombre de taxons est élevé, plus il faut répéter cette étape de construction d'arbres.

Une méthode de quadruplets pondérés requiert a priori un espace mémoire minimum en $O(n^4)$, lui permettant de stocker l'ensemble des pondérations des 4-arbres. Ce qui peut s'avérer problématique lorsque l'on souhaite traiter de grands jeux de données. Pour un ensemble de n taxons, il existe $3 \times C_4^n$ 4-arbres distincts. Si pour chacun de ces 4-arbres on mémorise une pondération codée sur 64 bits, alors pour $n = 150$ il faut près de 460Mb de mémoire, et pour $n = 200$ il faut près de 1500Mb. Cependant, au lieu de stocker ces pondérations, il est possible de modifier l'algorithme pour qu'il les calcule chaque fois que cela est nécessaire. Dans ce cas, on réduit considérablement la place mémoire nécessaire mais, dès que les poids des *w*4-arbres sont utilisés plus d'une fois, on augmente les temps de calculs. QP utilise 1000 fois le poids de chaque 4-arbre (une fois pour chaque ordre d'insertion), il n'est donc pas raisonnable d'optimiser la vraisemblance de chaque 4-arbre un si grand nombre de fois, et la complexité en espace de QP reste donc bien en $O(n^4)$.

3.4 Weight Optimization

Au vu de l'article (Strimmer, Goldman et Von Haeseler 1997), il nous a semblé que les performances de QP étaient prometteuses, et que son approche originale méritait d'être étudiée plus en détail. Nous avons donc cherché à corriger les faiblesses de la version originale. Ces diverses corrections nous ont amenés à proposer une méthode qui est finalement assez éloignée de QP et que nous avons appelée Weight Optimization (WO). Dans cette partie, nous présentons les améliorations que WO apporte à QP et nous présentons les principes qui nous ont permis de réaliser ces améliorations. Les détails algorithmiques ainsi que les preuves de complexités de cet algorithme sont donnés dans

l'article (Ranwez et Gascuel 2001a) qui constitue l'annexe 1 de cette thèse. Les simulations effectuées pour comparer cette approche à différentes méthodes de reconstruction phylogénétique sont décrites et leurs résultats analysés dans l'article (Ranwez et Gascuel 2001b) qui constitue l'annexe 2 de cette thèse.

3.4.1 Un nouveau critère d'insertion

WO est un algorithme glouton permettant d'optimiser le critère *W* (équation (27) page 70). Il utilise la version binaire ou la version continue de la pondération des 4-arbres proposée par (Strimmer, Goldman et Von Haeseler 1997). Comme QP, il suit un processus d'insertion, mais le critère d'insertion est différent. Au lieu d'attribuer une pénalité aux branches, il leur attribue un bonus ce qui permet de clarifier le lien qui existe entre la valuation des branches et le critère *W*. Au moment d'insérer le taxon *i*, WO choisit la branche permettant d'obtenir l'arbre T_{i+1} qui est le meilleur possible au sens du critère *W*. A chaque étape, WO optimise donc de manière locale le critère *W*. Pour cela, WO considère l'ensemble des *w*4-arbres de type (*xi*|*yz*, *w*) et ajoute le bonus *w* à toutes les branches où l'insertion de *i* génère un arbre qui induit *xi*|*yz*. Ce qui revient à pénaliser l'ensemble des branches d'un sous-arbre de l'arbre courant et non un seul chemin comme dans QP (Ranwez et Gascuel 2001a, p. 88). Une fois que tous les *w*4-arbres pertinents ont été pris en compte, le bonus de chaque branche *b* relativement à l'insertion du taxon *i* est égal à l'augmentation du critère *W* qui résulterait de cette insertion. La Figure 1 de l'article (Ranwez et Gascuel 2001b, p. 1105) illustre la manière dont WO utilise les *w*4-arbres pertinents pour sélectionner la branche du 4-arbre 12|34 sur laquelle il insère le taxon 5.

Le critère d'insertion de WO a l'avantage d'être directement lié au critère *W* que l'on souhaite optimiser. Ce lien direct permet de garantir que lorsque tous les quadruplets sont bien orientés, WO reconstruit la phylogénie correcte dans le cas binaire (tout comme QP), mais également dans le cas continu (ce qui n'était pas vrai pour QP).

3.4.2 Un ordre d'insertion défini dynamiquement

La phylogénie reconstruite par WO est initialement constituée de trois taxons sélectionnés aléatoirement. Les taxons restants sont ensuite insérés les uns après les autres sur *T*. Cependant, contrairement à QP qui utilise un ordre d'insertions défini aléatoirement, WO procède de manière à ce que les insertions les plus fiables soient faites en premier. Cette idée est également utilisée par (Willson 1999). Elle repose sur le fait que chaque insertion augmente le nombre de *w*4-arbres pertinents pour les insertions suivantes. Cet ordonnancement des insertions permet donc de disposer d'un maximum d'informations au moment d'effectuer les insertions les plus délicates. En notant *M* la branche ayant le plus fort bonus pour l'insertion de *i* et *m* la branche ayant le bonus immédiatement inférieur, nous définissons la fiabilité $F(i)$ de l'insertion du taxon *i* par

$$F(i) = \frac{\delta W(M,i) - \delta W(m,i)}{\delta W(M,i) + \delta W(m,i)} \qquad \textbf{(29)}$$

En ordonnant ainsi les insertions, il n'est plus nécessaire de reconstruire plusieurs centaines d'arbres suivant des ordres d'insertions différents comme c'est le cas avec QP. Nous n'avons donc pas besoin de déterminer le consensus d'un ensemble d'arbres et nous évitons ainsi le biais topologique important dont souffre QP. Pour comparer la qualité des arbres reconstruits par QP et par WO, nous avons repris le protocole de test proposé par (Kumar 1996) et repris par (Gascuel 1997) pour valider des méthodes de reconstruction phylogénétique. Ces tests montrent clairement que les arbres reconstruits par WO sont nettement meilleurs que ceux reconstruits par QP. En fait, les seuls cas où QP obtient de bons résultats correspondent aux cas où le biais topologique de QP devient un atout du fait que la phylogénie correcte possède un très grand nombre de 2-bipartitions. Cela explique les très bons résultats obtenus par QP lors des premiers tests, pour lesquels toutes les phylogénies correctes contenaient le maximum de 2-bipartitions possibles.

3.4.3 Une complexité optimale

Nous avons prouvé que la complexité en temps de calcul de WO est $O(n^4)$ (Ranwez et Gascuel 2001a). Pour un ensemble de n taxons, il existe $O(n^4)$ quadruplets différents. Une méthode qui prend en compte chacun de ces quadruplets a donc une complexité minimale de $O(n^4)$ et, en ce sens, la complexité de WO est optimale (Ranwez et Gascuel 2001a, Pp. 89-90). L'idée principale qui permet de conserver cette complexité optimale dans WO est de factoriser la propagation des bonus. Pour prendre en compte un w4-arbre pertinent, il faut ajouter un bonus à toutes les branches d'un sous-arbre de l'arbre courant. Au lieu de considérer indépendamment chaque w4-arbre, on regroupe le traitement de tous ceux qui doivent ajouter un bonus aux branches d'un même sous-arbre T_s. On peut alors calculer la somme de ces bonus, et ajouter en une seule fois la somme de ces bonus à chacune des branches de T_s. Ainsi, il suffit de parcourir une seule fois T_s pour prendre en compte de nombreux w4-arbres pertinents. De plus, au lieu de parcourir successivement les différents sous-arbres de l'arbre courant pour y ajouter leurs bonus globaux respectifs, il est possible d'effectuer simultanément l'ensemble de ces ajouts en parcourant seulement deux fois l'arbre courant. On utilise pour cela le double parcours récursif, cela ne change pas la complexité, mais permet de réduire le temps de calcul (Ranwez et Gascuel 2001a, Pp. 91-94). De plus WO construit un seul arbre alors que QP en construit 1000, ce qui permet à WO d'être nettement plus rapide que QP (Ranwez et Gascuel 2001b, p. 1112).

Comme WO ne construit qu'un seul arbre et que pour cela il ne consulte qu'une fois le poids de chaque w4-arbre, il n'est pas nécessaire de stocker ces valeurs. Le poids de chaque w4-arbre peut être calculé au moment de le prendre en compte sans ralentir l'algorithme. Cela nécessite de stocker les n séquences de l nucléotides, mais réduit néanmoins considérablement l'espace mémoire nécessaire à WO. L'algorithme WO ainsi modifié a une complexité en espace qui est seulement de $O(nl + n^2)$.

En utilisant des principes algorithmiques similaires à ceux utilisés par WO, nous avons également pu réduire la complexité de l'algorithme QP à $O(n^4)$. Cependant, au vu des

nombreux avantages que présente WO par rapport à QP, cette réduction de la complexité de QP semble désormais d'un faible intérêt.

3.5 Discussion

Nous avons effectué de nombreuses simulations informatiques pour évaluer la fiabilité des arbres reconstruits par WO et QP ainsi que les temps de calculs de ces deux algorithmes. Ces tests montrent que les performances de WO sont nettement supérieures à celles de QP. Ils montrent également que les performances de ces deux méthodes restent inférieures à celles obtenues avec une méthode de distances. Nous avons donc réalisé des tests complémentaires afin de comprendre d'où proviennent les faiblesses de WO. Cela nous a permis de mettre en évidence certaines difficultés qui semblent inhérentes aux méthodes de quadruplets. L'article (Ranwez et Gascuel 2001b) détaille les simulations et les analyses sur lesquelles s'appuient les résultats qui sont résumés dans cette partie.

3.5.1 Des performances décevantes en phylogénie

Nous avons comparé la fiabilité des arbres reconstruits par WO et QP à celles obtenues par DNAPARS (Felsenstein 1993) qui est une méthode de parcimonie, par BIONJ (Gascuel 1997) qui est une méthode de distances et par FASTDNAML (Olsen et al. 1994) qui est une méthode de maximum de vraisemblance. La matrice de distances utilisée par BIONJ est obtenue à partir des séquences nucléotidiques en utilisant le programme DNADIST (Felsenstein 1993). Afin d'obtenir des tests suffisamment impartiaux et représentatifs des problèmes biologiques réels, nous avons repris le protocole utilisé par (Kumar 1996) et (Gascuel 1997) dans un cadre similaire.

Les résultats de ces tests sont fournis dans les Tableaux 1 à 6 de l'Annexe 2 (Ranwez et Gascuel 2001b). Ces résultats indiquent clairement que les performances de QP sont nettement moins bonnes que celles de WO, excepté pour un cas extrême où le biais topologique de QP devient un atout. Malgré l'amélioration sensible que WO représente par rapport à QP, ces tests remettent directement en cause l'intérêt de ce type d'approche. En effet, les arbres reconstruits par WO sont nettement moins fiables que ceux inférés par FASTDNAML. Ils sont moins fiables que ceux obtenus avec DNAPARS sauf pour des taux d'évolution extrêmement élevés. Ils sont également moins fiables que ceux obtenus avec BIONJ, et cela pour toutes les conditions d'évolution que nous avons testées. Cela peut sembler paradoxal, puisque les méthodes de quadruplets utilisent de manière plus intensive l'approche par maximum de vraisemblance que les méthodes de distances. On peut cependant noter qu'elles n'utilisent pas l'ensemble des informations fournies par cette analyse. En particulier, elles ne prennent pas en compte les longueurs des branches des 4-arbres obtenus lors de l'optimisation de la vraisemblance. De plus, l'information utilisée par les méthodes de quadruplets est fortement redondante, et par conséquent, elle n'est peut être pas aussi riche qu'il y parait. D'une certaine manière, on peut estimer que les méthodes de distances disposent de plus d'informations que les méthodes de quadruplets. En effet, à

partir d'une matrice de distances évolutives on peut inférer une solution à chacun des 4-arbres. En revanche, un ensemble de 4-arbres, pondérés ou non, ne permet pas de déduire une matrice de distances évolutives.

Les temps de calcul que nous avons mesurés montrent également l'apport de WO par rapport à QP. Néanmoins, même WO est plus lent qu'une méthode de distances telle que BIONJ. Par exemple pour traiter un jeu de 25 séquences de 1896 nucléotides sur un PC ayant un processeur cadencé à 466Mhz avec 256MB de RAM il faut : moins de 3 secondes pour DNAPARS et pour DNADIST+BIONJ, 53 secondes pour WO, 101 pour QP et 318 pour FASTDNAML.

Ainsi, et de manière paradoxale, WO (et a fortiori QP) est une méthode nettement moins fiable que BIONJ tout en étant plus lente.

3.5.2 Analyse des faiblesses possibles de WO

Afin de comprendre les raisons des résultats décevants obtenus par WO, nous avons réalisé des tests complémentaires. Ces tests montrent que lorsque WO ne reconstruit pas l'arbre correct, il reconstruit généralement un arbre meilleur que l'arbre vrai au sens du critère *W*. Cela montre que les résultats décevants de WO ne sont pas liés à l'heuristique qu'utilise WO pour optimiser le critère *W*, mais reflètent les limites du critère *W*.

Nous avons proposé un critère $P(T)$ alternatif à $W(T)$, et une heuristique PO permettant de l'optimiser (Ranwez et Gascuel 2001a). Ce critère attribue un poids à chacun des chemins de l'arbre *T*, le critère $P(T)$ correspond alors à la somme des poids de ces chemins. Nous avons montré de manière expérimentale que ce critère est au moins aussi pertinent que le critère $W(T)$. Néanmoins, le gain qu'apporte ce critère est insuffisant pour rivaliser avec les méthodes de distances.

Une autre possibilité pour augmenter les performances des méthodes de quadruplets est d'essayer d'améliorer le système de pondération des 4-arbres qu'ils utilisent. Cependant, la pondération actuelle proposée par (Strimmer, Goldman et Von Haeseler 1997) est raisonnable et il semble probable que, là encore, le gain possible soit très limité.

On peut donc finalement se demander, si ce n'est pas l'approche même des méthodes de quadruplets qui doit être remise en cause.

3.5.3 Limites des méthodes de quadruplets

Comme nous le soulignons dans notre article (Ranwez et Gascuel 2001b, p. 1113), il suffit qu'un très faible pourcentage de quadruplets soit mal résolu pour qu'aucune méthode de quadruplets ne puisse retrouver l'arbre correct. Si l'on considère un ensemble de n taxons, il existe des arbres qui ne diffèrent que sur la résolution de $n-3$ des $O(n^4)$ quadruplets. Nous présentons le reste de notre analyse dans le cas discret, mais elle se transpose facilement au cas continu en s'appuyant sur la notion de quadruplets bien orientés. Dans le cas binaire, si l'on note T_1 l'arbre vrai et T_2 un arbre obtenu à partir de T_1 en déplaçant une

de ses feuilles, il est possible, en changeant seulement la résolution de $(n-3)$ quadruplets de transformer, Q_{T_1} en Q_{T_2} (Ranwez et Gascuel 2001b, p. 1115). Dans ce cas, il suffit que ces $(n-3)$ quadruplets soient mal résolus pour que toutes les méthodes de quadruplets reconstruisent T_2 au lieu de T_1. Le taux de quadruplets mal résolus, suffisant pour qu'aucune méthode de quadruplet ne reconstruise la phylogénie correcte, est de l'ordre de $1/n^3$. Ce taux tend donc très rapidement vers 0 lorsque *n* augmente.

En pratique, nous avons observé des cas où plus de 90% des quadruplets sont bien résolus et où la phylogénie correcte n'est pas reconstruite. En fait, la majeure partie des erreurs semble se concentrer sur les 4-arbres locaux. Rappelons qu'un 4-arbre est dit local pour la phylogénie *T* lorsque l'arête interne de ce 4-arbre correspond à un chemin de *T* réduit à une seule branche. Les 4-arbres locaux sont donc particulièrement soumis au phénomène d'attraction des longues branches ce qui explique pourquoi ils sont difficiles à inférer. Or, pour pouvoir reconstruire la phylogénie correcte à partir d'une méthode de quadruplets, il est particulièrement important que les 4-arbres locaux soient bien inférés (Erdös et al. 1997b).

La difficulté d'inférer correctement la phylogénie de quadruplets en utilisant une approche de parcimonie à été étudiée par (Philippe et Douzery 1994) qui concluent "*Reconstructing history with only four taxa is rather a game of chance*". Les travaux de (Strimmer et Von Haeseler 1996) ont donné l'espoir que l'utilisation du maximum de vraisemblance permettrait de contourner cette difficulté. Mais, plusieurs travaux récents ont sérieusement remis en cause cet espoir. En particulier, Adachi et Hasegawa (1999) concluent "*As Philipe and Douzery showed, it is now clear that an argument based on a quartet analysis of a single gene is very dangerous*". Nos travaux confirment leur analyse et permettent de comprendre les raisons qui ont conduit à surestimer les performances de QP ainsi que les limites intrinsèques de ce type de méthode de quadruplets.

3.6 Conclusion, autres applications

Les méthodes de quadruplets semblaient constituer une approche prometteuse pour obtenir des méthodes de reconstruction phylogénétique capables de traiter de grands jeux de données tout en étant plus fiables que les méthodes de distances. L'algorithme QP semblait particulièrement efficace. Nos travaux ont permis de mettre en évidence plusieurs faiblesses de cette approche et d'y apporter des solutions. L'algorithme WO issu de ces améliorations possède de meilleures propriétés théoriques que QP, et nos simulations nous ont permis de vérifier que les arbres inférés par WO sont également plus fiables que ceux inférés par QP.

Nous avons montré que ce type de méthode de quadruplets n'est actuellement pas une alternative intéressante aux méthodes de distances. Elles ont une complexité théorique plus élevée que les méthodes de distances, sont donc plus lentes et ne peuvent pas traiter des jeux de données aussi grands que ces dernières. De plus, les arbres qu'elles reconstruisent

sont moins fiables que ceux obtenus avec les méthodes de distances. Il semble que ces résultats décevants reflètent les limites inhérentes aux méthodes de quadruplets.

Une des difficultés majeures des méthodes de quadruplets semble être d'arriver à prendre en compte correctement le fait qu'il existe plusieurs topologies possibles pour un même quadruplet. Or ce problème ne se pose plus lorsque l'on considère seulement deux ou trois taxons puisque, dans ce cas, il n'existe qu'une seule topologie possible. Le cas de deux taxons correspond au cas des méthodes de distances classiques. Dans le chapitre suivant, nous montrons comment les distances obtenues en résolvant des sous-problèmes portant sur trois taxons permettent d'améliorer la fiabilité des arbres inférés par les méthodes de distances. De plus, cette amélioration se fait en conservant des temps de calculs faibles qui permettent de traiter de très grands jeux de données.

Cependant, les méthodes de quadruplets peuvent certainement servir de base pour résoudre de manière efficace d'autres types de problèmes. On peut par exemple définir une distance topologique entre deux phylogénies basée sur les quadruplets. Cette distance correspond simplement au nombre de 4-arbres qui sont présents dans une seule des deux phylogénies. Cette distance fournit une mesure plus fine que la distance de Robinson et Fould et l'on peut la calculer en temps quasi linéaire. En effet, Brodal et al. (2001) décrivent un algorithme en $O(n\log^2 n)$ pour calculer cette distance. Si l'on considère un ensemble d'arbres, on peut définir son consensus comme étant le barycentre au sens de cette distance. On peut espérer que le consensus ainsi défini ait un biais topologique moins important que celui que nous avons observé avec un consensus basé sur une distance de bipartition. En particulier, si l'on pondère chaque 4-arbre par le nombre d'arbres de S qui l'induisent, rechercher la phylogénie T complètement résolue qui est le barycentre de S, revient à optimiser le critère $W(T)$. On peut également penser qu'une approche par quadruplets permettrait de combiner efficacement des phylogénies portant sur des ensembles partiellement disjoints de taxons. Des travaux utilisant ce type d'approche sont en cours au sein de notre équipe.

Chapitre 4 Méthodes de distances et maximum de vraisemblance

Nous avons vu dans le chapitre précédent, qu'actuellement, les méthodes de quadruplets ne permettent pas de reconstruire de grandes phylogénies suivant le principe du maximum de vraisemblance. On considère généralement que les méthodes de distances constituent une famille de méthodes de reconstruction phylogénétique distincte de l'approche par maximum de vraisemblance. Cependant, lorsque l'on étudie un ensemble de séquences nucléotidiques, les distances évolutives qui constituent la matrice des distances initiales sont estimées au sens du maximum de vraisemblance. Dans ce cas, on peut donc légitimement considérer que les méthodes de distances constituent des heuristiques permettant de reconstruire une phylogénie suivant le principe de maximum de vraisemblance (Swofford et al. 1996, p. 446). Dans ce chapitre

nous décrivons NJ+TRIPLEML, une nouvelle méthode de reconstruction phylogénétique qui combine de manière encore plus forte les méthodes de distances et le principe du maximum de vraisemblance. NJ+TRIPLEML utilise un processus analogue pour estimer les distances initiales et pour réduire la matrice de distances utilisée par NJ (et ses variantes). Dans les deux cas, les distances sont estimées à partir d'une optimisation locale de la vraisemblance basée sur des triplets de taxons (ou de groupes de taxons).

NJ+TRIPLEML n'est pas la seule méthode qui combine les méthodes de distances et les méthodes de maximum de vraisemblance. Les résultats obtenus par NJML (Ota et Li 2000) ont déjà montré l'intérêt d'une telle combinaison. Cette méthode utilise un arbre reconstruit par NJ pour restreindre l'espace dans lequel sera ensuite effectuée la recherche d'un arbre de vraisemblance élevée. NJML est un enchaînement astucieux de NJ est d'une méthode de maximum de vraisemblance, alors que NJ+TRIPLEML est une méthode hybride qui intègre l'utilisation du maximum de vraisemblance au cœur de NJ. Nous pensons que NJML et NJ+TRIPLEML sont deux approches différentes qui sont plus complémentaires que concurrentes. En effet, il semble assez facile de les combiner, la manière la plus simple étant de remplacer dans NJML l'utilisation de NJ par celle de NJ+TRIPLEML.

Nous commençons par détailler les propriétés des distances évolutives sur lesquelles repose notre approche. Puis, en nous appuyant sur l'article (Ranwez et Gascuel 2002) (fourni en annexe 3), nous décrivons l'algorithme NJ+TRIPLEML et ses principales qualités. Nous évoquons ensuite les possibilités que nous explorons actuellement pour améliorer et généraliser cette approche.

4.1 Propriétés des distances évolutives

Cette partie décrit les propriétés importantes des distances évolutives. Nous verrons dans la suite de ce chapitre comment ces propriétés sont prises en compte dans TRIPLEML. Nous commençons par rappeler que le passage d'un ensemble de séquences de nucléotides à une matrice de distances évolutives se fait au prix d'une perte d'information. En nous appuyant sur le modèle de Jukes et Cantor, nous soulignons ensuite le lien existant entre les formules analytiques permettant d'estimer les distances évolutives et l'approche par maximum de vraisemblance. Puis nous montrons, sur ce modèle simple, que la fiabilité de l'estimation d'une distance évolutive diminue très rapidement lorsque cette distance augmente.

4.1.1 Perte d'information lors du passage aux distances évolutives

Le calcul de la distance qui sépare deux séquences S_i et S_j ne prend pas en compte les autres séquences du jeu de données. Cette distance est simplement déduite du nombre d'apparitions des seize motifs possibles dans le jeu de séquences réduit à l'union de S_i et S_j. Il est donc possible que différents jeux de données aient des matrices de distances identiques (Penny 1982). Les méthodes de distances disposent donc de moins d'informations que les méthodes de caractères qui, comme la parcimonie ou le maximum de vraisemblance, utilisent directement les séquences moléculaires. Cependant, même pour

des méthodes de caractères, il existe de nombreux jeux de données pour lesquels l'arbre reconstruit est le même (Olsen 1987). Cette perte d'information lors du passage à la matrice de distances évolutives ne remet donc pas directement en cause l'approche des méthodes de distances. Néanmoins, cette perte d'information est certainement une des causes principales de l'écart entre les performances des méthodes de distances et celles des méthodes de maximum de vraisemblance (Swofford et al. 1996, p. 446). En effet, cette compression (avec perte) de l'information, permet aux méthodes de distances d'être très rapides mais limite la quantité d'information dont elles disposent et donc la fiabilité des phylogénies qu'elles reconstruisent.

4.1.2 Distances évolutives et maximum de vraisemblance

Nous avons vu dans le second chapitre (§ 2.2.1) qu'il existe, pour certains modèles, des formules analytiques permettant d'estimer la distance évolutive δ qui sépare deux séquences S_1 et S_2 à partir des différences observées entre ces séquences.

Dans le cas général, cette distance peut être estimée en optimisant la vraisemblance du 2-arbre T dont les deux feuilles sont respectivement associées à S_1 et S_2. La vraisemblance de T s'exprime alors de manière très simple à partir des équations (22) et (24) (pages 51 et 53). En supposant que S_1 est la séquence ancestrale on obtient :

$$L(T) = \prod_{s=1}^{l} \pi_{S_1^s} P_{S_1^s S_2^s}(\delta)$$

Comme dans le cas des 4-arbres, il existe très peu de motifs différents. Pour deux séquences d'ADN, il n'existe que seize motifs possibles. Ce qui permet de calculer rapidement la vraisemblance de T pour différentes distances. En notant n_{XY} le nombre de sites s pour lesquels $S_1^s = X$ et $S_2^s = Y$, *i.e.* le nombre d'occurrences du motif XY, on obtient la formule suivante :

$$L(T) = \prod_{(X,Y)\in\{ACGT\}^2} \left[\pi_X P_{XY}(\delta)\right]^{n_{XY}}$$

Suivant le modèle utilisé, des motifs différents peuvent avoir une influence identique sur le calcul de la vraisemblance. Par exemple, dans le cas du modèle de Jukes et Cantor, il suffit de distinguer deux types de sites : ceux où les nucléotides sont les mêmes pour S_1 et S_2, et ceux où il s'agit de deux nucléotides distincts. En notant k le nombre de sites s tels que $S_1^s = S_2^s$, et p la probabilité d'un tel site (équations (10) page 25), la vraisemblance de T, s'écrit simplement :

$$L(T) = p^k(1-p)^{n-k} \text{ avec } \begin{cases} p = \frac{3}{4} - \frac{3}{4}e^{-8\alpha t} \\ (1-p) = \frac{1}{4} + \frac{3}{4}e^{-8\alpha t} \end{cases}$$

Puisque les séquences S_1 et S_2 sont connues, les valeurs de k et n sont fixées, et $L(T)$ dépend donc uniquement de αt. De plus, pour le modèle de Jukes et Cantor, la distance évolutive séparant S_1 et S_2, vaut $6\alpha t$ (équation (9), page 25). On peut donc déduire l'estimation optimale de cette distance au sens du maximum de vraisemblance à partir de la valeur de αt qui maximise $L(T)$.

La fonction logarithmique étant strictement croissante, il est équivalent de maximiser $L(T)$ ou de maximiser :

$$\ln(L(T)) = k\ln\left(\frac{3}{4} - \frac{3}{4}\exp(-8\alpha t)\right) + (n-k)\ln\left(\frac{1}{4} + \frac{3}{4}\exp(-8\alpha t)\right)$$

L'optimum de cette fonction est atteint en un point où sa dérivée par rapport à αt s'annule. La dérivée de $\ln(f(x))$ par rapport à x valant $f'(x)/f(x)$, on a donc :

$$\begin{aligned} \frac{\partial \ln(L(T))}{\partial t} &= k\frac{6\alpha t\exp(-8\alpha t)}{\frac{3}{4} - \frac{3}{4}\exp(-8\alpha t)} + (n-k)\frac{-6\alpha t\exp(-8\alpha t)}{\frac{1}{4} + \frac{3}{4}\exp(-8\alpha t)} \\ &= (24\alpha t\exp(-8\alpha t))\left(\frac{k}{3-3\exp(-8\alpha t)} + \frac{(k-n)}{1+3\exp(-8\alpha t)}\right) \end{aligned}$$

Les extremums de $\ln(L(T))$ correspondent donc aux valeurs de αt qui annulent l'un des deux termes de ce produit. Le premier terme s'annule pour $\alpha t = 0$. En ce point, la vraisemblance vaut 0 pour toutes les valeurs de $k \neq 0$. Pour ces valeurs de k, $\alpha t = 0$ correspond donc à une valeur minimale. L'étude de la dérivée seconde de $\ln(L(T))$ montre que le second terme s'annule pour une valeur maximale de $\ln(L(T))$. C'est donc le cas où le second terme s'annule qui nous intéresse, c'est-à-dire le cas où :

$$\begin{aligned} &k(1+3\exp(-8\alpha t)) + (k-n)(3-3\exp(-8\alpha t)) = 0 \\ &\Rightarrow 4k - 3n + 3n\exp(-8\alpha t) = 0 \end{aligned}$$

soit :

$$-8\alpha t = \ln\left(1 - \frac{4k}{3n}\right) \quad \textbf{(30)}$$

On rappelle que pour le modèle de Jukes et Cantor, $\delta = 6\alpha t$. En utilisant l'équation (30), on obtient donc l'estimation de δ au sens du maximum de vraisemblance, et l'on retrouve la formule analytique classique :

$$\delta_{12} = \begin{cases} -\frac{3}{4}\ln(1-\frac{4k}{3n}) & \text{si } (k/n < 3/4) \\ \infty & \text{sinon} \end{cases} \quad \textbf{(31)}$$

Pour les modèles d'évolution plus complexes, les formules analytiques se retrouvent de manière similaire en cherchant les points où la dérivée de $\ln(L(T))$ s'annule. Lorsque le modèle comporte plusieurs paramètres, il faut que la dérivée s'annule par rapport à ces différents paramètres, on obtient ainsi un système d'équations dont l'une des solutions fournit la formule recherchée. Par exemple, pour le modèle à deux paramètres de Kimura, il faut étudier les zéros de $\partial \ln(L(T)) / \partial \alpha t$ et de $\partial \ln(L(T)) / \partial \beta t$. Il est important de noter, que dans cette approche, les paramètres αt et βt sont estimés au sens du maximum de vraisemblance et que leur rapport ne peut pas être fixé a priori.

Il existe, pour plusieurs modèles d'évolution, des formules analytiques qui fournissent une distance optimale au sens du maximum de vraisemblance. Ces formules déduisent implicitement les paramètres du modèle d'évolution à partir des deux séquences étudiées. Chacune des distances contenues dans la matrice de distances correspond donc à une optimisation de la distance pour un modèle d'évolution commun, mais dont les paramètres peuvent prendre des valeurs différentes. On ne peut donc pas utiliser ces formules analytiques lorsque, comme le conseillent (Swofford et al. 1996, p. 458), on utilise les mêmes paramètres pour estimer l'ensembles des distances. Dans ce cas, il est nécessaire d'utiliser une méthode numérique d'optimisation, telle que celles décrites dans le second chapitre (§2.4.3.1). Ces méthodes d'optimisation permettent, lorsque les valeurs des paramètres du modèle sont fixées, de chercher efficacement la distance optimale au sens du maximum de vraisemblance. Comme nous l'avons vu dans la partie 2.4.1, on peut choisir le modèle le plus approprié et fixer les valeurs de ses paramètres en s'appuyant sur la phylogénie reconstruite par NJ à partir des distances estimées par les formules analytiques. Une fois ces valeurs déterminées, il est possible de ré-estimer les distances évolutives en fixant les paramètres du modèle. Dans la partie 4.2.4, nous détaillons la manière dont les distances peuvent être estimées, au sens du maximum de vraisemblance, lorsque l'on fixe le ratio transition/transvertion du modèle à deux paramètres de Kimura.

4.1.3 Variance des estimateurs de distances évolutives

Lorsque l'on cherche à estimer par δ la distance d qui sépare S_1 et S_2, la fiabilité de l'estimateur δ se dégrade lorsque la distance d augmente. Plus précisément, la variance de l'estimateur δ est une fonction exponentielle de d. Par exemple, la variance σ^2 de l'estimateur de Jukes et Cantor (équation (31) page 89) est approchée, en utilisant la "*delta method*" (De Groot 1989, p. 449) par :

$$\sigma^2 = \frac{p(1-p)}{n\left(1-\frac{4}{3}p\right)^2} \text{ avec } \begin{cases} p = \frac{3}{4} - \frac{3}{4}e^{-\frac{4d}{3}} \\ \left(1-\frac{4}{3}p\right) = e^{-\frac{4d}{3}} \end{cases}$$

soit :

$$\begin{aligned} \sigma^2 &= \frac{\frac{3}{4}\left(1-e^{-\frac{4d}{3}}\right)\times\frac{1}{4}\left(1+3e^{-\frac{4d}{3}}\right)}{le^{-16\alpha t}} \\ &= \frac{3}{16l}\left(1+2e^{-\frac{4d}{3}}-3e^{-\frac{8d}{3}}\right)e^{\frac{8d}{3}} \end{aligned}$$

soit :

$$\sigma^2 = \frac{3}{16l}\left(\exp\left(\frac{8d}{3}\right)+2\exp\left(\frac{4d}{3}\right)-3\right) \quad \textbf{(32)}$$

Cette formule de la variance, que l'on trouve par exemple dans (Nei et Jin 1989), est spécifique au modèle de Jukes et Cantor. Cependant, comme le soulignent Rzhetsky et Nei (1995), lorsque l'on utilise des modèles ayant des paramètres supplémentaires, les estimateurs des distances évolutives ont des variances encore plus fortes. L'étude du modèle de Jukes et Cantor nous suffit donc pour montrer que la fiabilité de l'estimateur δ se dégrade très rapidement lorsque la distance *d* augmente.

Pour des distances évolutives inférieures à 0,1, la formule (32) et celles obtenues pour les autres modèles sont très proches (Nei et Jin 1989). En particulier, au voisinage de 0, tous les modèles deviennent équivalents et leur variance peut être estimée en utilisant le développement limité de (32) (Gascuel 1997). Rappelons que pour des valeurs de *x*, proches de 0, $e^x \approx 1+x$. Pour des distances faibles, le développement limité au premier ordre de cette formule donne donc :

$$\begin{aligned} \sigma^2 &\approx \frac{3}{16l}\left(\left(1+\frac{8d}{3}\right)+2\times\left(1+\frac{4d}{3}\right)-3\right) \\ &\approx \frac{d}{l} \end{aligned} \quad \textbf{(33)}$$

ce qui, comme on pouvais s'y attendre, correspond à la variance d'un phénomène de Poisson.

Plusieurs méthodes de distances prennent en compte la variance des estimateurs des distances évolutives. Quel que soit le modèle d'évolution utilisé pour estimer les distances

évolutives, les variances de ces estimateurs sont obtenues par WEIGHBOR (Bruno, Socci et Halpern 2000) en s'appuyant sur la formule (32) (page 90), tandis que BIONJ (Gascuel 1997) et FITCH (Fitch et Margoliash 1967 ; Felsenstein 1993) s'appuient sur la formule (33) (page 90).

4.2 TripleML

Dans les méthodes de distances qui utilisent un processus agglomératif, telles que NJ, BIONJ et Weighbor, il y a deux types d'estimations de distances. Le premier concerne l'estimation des distances initiales. Le second concerne l'estimation des distances qui séparent un nouveau sous-arbre des sous-arbres existants, lors de l'étape de réduction de la matrice. En nous appuyant sur les propriétés des distances évolutives, nous proposons une nouvelle manière d'estimer ces deux types de distances. Cette approche, que nous appelons TRIPLEML, améliore l'estimation des distances évolutives et donc la qualité des phylogénies reconstruites par les méthodes de distances. Avec des méthodes agglomératives telles que NJ, BIONJ et WEIGHBOR, TRIPLEML peut être utilisé à la fois pour calculer la matrice de distances initiales et pour estimer les nouvelles distances après chaque agglomération.

Nous commençons par présenter la manière dont TRIPLEML estime les distances initiales séparant deux séquences nucléotidiques. Cette estimation se fait en utilisant une troisième séquence, nous précisons donc ensuite la manière dont cette troisième séquence est sélectionnée. Puis, nous expliquons comment nous généralisons cette approche afin d'estimer de manière analogue les distances entre deux sous-arbres.

4.2.1 Estimation des distances initiales

L'estimation de la distance δ_{ij}, qui sépare deux taxons contemporains *i* et *j*, est généralement obtenue en optimisant la vraisemblance de "l'arbre" qui contient ces deux taxons (§ 4.1.2). Au lieu d'estimer la distance qui sépare le taxon *i* du taxon *j* en utilisant le 2-arbre correspondant, nous proposons d'estimer cette distance à partir d'un 3-arbre. Deux feuilles de cet arbre sont les taxons étudiés et la troisième, que l'on note *k*, est choisie de manière à prendre en compte les problèmes liés à l'estimation des longues branches (§ 4.1.3). Les longueurs des trois branches de cet arbre sont fixées de manière à maximiser sa vraisemblance, et sont ensuite utilisées pour affiner l'estimation de δ_{ij}. Ce 3-arbre est obtenu en reliant les taxons *i*, *j* et *k* à un ancêtre commun *a* grâce à trois branches dont les longueurs sont respectivement δ_{ai}, δ_{aj} et δ_{ak}. La distance δ_{ij} est alors estimée par $\delta_{ij} = \delta_{ai} + \delta_{aj}$. La qualité de cette estimation dépend du troisième taxon que l'on a choisi. Utiliser tous les 3-arbres contenant *i* et *j* est trop coûteux en temps de calcul. Il faut donc choisir a priori un troisième taxon permettant d'obtenir un bon estimateur δ_{ij}. C'est pourquoi, dans notre approche, les distances initiales entre deux taxons *i* et *j* sont estimées en deux étapes : on commence par estimer ces distances de manière classique en utilisant le maximum de vraisemblance sur le 2-arbre correspondant. Puis, ces premières

estimations sont utilisées pour sélectionner pour chaque paire $\{i, j\}$ un troisième taxon permettant d'améliorer l'estimation de δ_{ij}. La distance δ_{ij} est alors ré-estimée en utilisant le maximum de vraisemblance sur le 3-arbre contenant ces trois taxons.

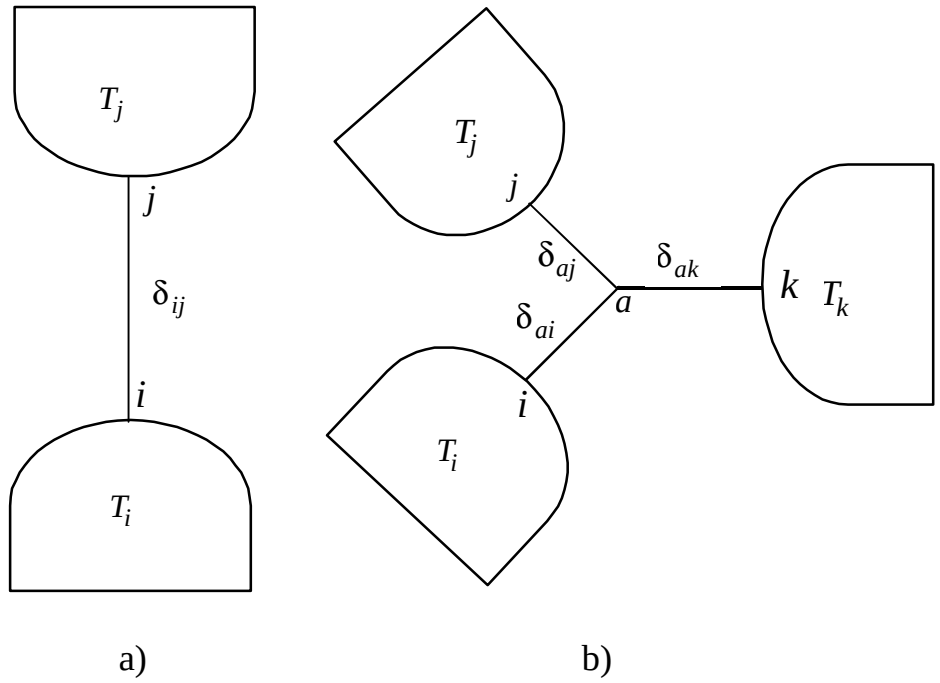
4.2.2 Sélection du troisième taxon

Utiliser un troisième taxon *k* permet de disposer de plus d'informations pour estimer la distance δ_{ij} qui sépare les taxons *i* et *j*. Ce phénomène est souligné par Swofford et al. (1996). Pour obtenir une phylogénie plus fiable, ils conseillent d'utiliser des taxons supplémentaires de manière à couper les longues branches à peu près en leur milieu, de reconstruire la phylogénie de cet ensemble plus large de taxons, puis d'enlever les taxons supplémentaires de la phylogénie ainsi obtenue. En s'appuyant sur cette remarque, nous cherchons un taxon *k* qui coupe la branche (*i*, *j*) en un point proche de son milieu, ce que nous mesurons par $(\delta_{ik} - \delta_{jk})^2$ en utilisant les premières estimations des distances. Lorsque l'on coupe ainsi la branche (*i*, *j*), on crée un nœud interne *a* et une nouvelle branche (*a*, *k*). Si l'on coupe une longue branche en utilisant une autre longue branche, il est peu probable que l'estimation de δ_{ij} soit réellement améliorée. On souhaite donc aussi que le taxon *k* soit proche de *i* et de *j*, ce que nous mesurons par $\delta_{ik}\delta_{jk}$. Ces deux mesures sont du même ordre, puisqu'elles correspondent toutes les deux à un produit de distances. Pour estimer δ_{ij}, nous cherchons donc le taxon *k* qui minimise $(\delta_{ik} - \delta_{jk})^2 + \delta_{ik}\delta_{jk}$. Il est facile de voir que ce critère est minimal lorsque $\delta_{ai} = \delta_{aj}$ et $\delta_{ak} = 0$, c'est-à-dire dans le cas idéal où il existe un taxon équidistant de *i* et *j* dont la séquence est connue. D'autres critères sont possibles, nous en avons testé de nombreux, mais nous n'en avons trouvé aucun qui permette d'obtenir de meilleurs résultats que ceux obtenus avec le critère, simple, donné ci-dessus.

4.2.3 Réduction de la matrice de distances

Après chaque agglomération, de nouvelles distances δ_{ij} sont estimées. NJ, BioNJ et WEIGHBOR estiment ces distances en utilisant une moyenne pondérée de distances. Dans TRIPLEML, ces nouvelles distances ne sont pas déduites de la matrice de distances courante (comme dans NJ, BIONJ, ou WEIGBOR) mais directement à partir des séquences nucléotidiques. Cela permet de réduire la perte d'information due à l'utilisation d'une matrice de distances (§ 4.1.1).

Une agglomération induit un nouveau sous-arbre, et l'on cherche à estimer les distances qui séparent ce nouveau sous-arbre de ceux déjà existants. La distance qui sépare un sous-arbre T_i, ayant le noeud *i* pour racine, et le sous-arbre T_j, ayant *j* pour racine, peut être estimée à partir de la phylogénie $T_i \cup T_j$ obtenue à partir de T_i et de T_j en ajoutant la branche (*i*, *j*) (Figure 17.a).



a) b)

Figure 17 : estimation de la distance δ_{ij} séparant T_i de T_j

a) Une première estimation de δ_{ij} peut être obtenue à partir de la phylogénie $T_i \cup T_j$.
b) Cette estimation peut ensuite être améliorée en utilisant un troisième sous-arbre T_k et la phylogénie $T_i \cup T_j \cup T_k$, on a alors $\delta_{ij} = \delta_{ai} + \delta_{aj}$.

Les longueurs des branches de $T_i \cup T_j$ pourraient être ajustées de manière à maximiser sa vraisemblance. On obtiendrait ainsi, non seulement l'estimation de δ_{ij} que l'on cherche, mais aussi de nouvelles estimations des longueurs de branches de T_i et de T_j. Cependant, pour conserver un temps de calcul raisonnable, nous avons choisi de ne pas remettre en question les estimations des longueurs des branches de T_i et de T_j qui ont été obtenues lors des étapes précédentes. Au lieu d'optimiser globalement la vraisemblance de $T_i \cup T_j$, nous optimisons localement cette vraisemblance uniquement par rapport à δ_{ij}.

Une meilleure estimation de δ_{ij} peut être obtenue en utilisant un troisième sous-arbre T_k dont la racine est le nœud *k*. Dans ce cas on considère la phylogénie $T_i \cup T_j \cup T_k$ obtenue à

partir de T_i, T_j et T_k en reliant les trois nœuds *i*, *j* et *k* à un nouveau nœud *a* grâce à trois branches de longueurs respectives δ_{ai}, δ_{aj} et δ_{ak} (Figure 17.b). Ces trois longueurs sont ajustées de manière à maximiser la vraisemblance de $T_i \cup T_j \cup T_k$ et la distance δ_{ij} est ré-estimée par $\delta_{ij} = \delta_{ai} + \delta_{aj}$.

Ainsi, lors de la création par agglomération d'un nouveau sous-arbre T_i, les distances δ_{ij} le séparant des autres sous-arbres T_j sont estimées en deux étapes. Dans un premier temps, ces distances sont estimées en optimisant localement la vraisemblance $T_i \cup T_j$. Ces premières estimations sont ensuite utilisées pour sélectionner pour chaque paire $\{T_i, T_j\}$ un troisième sous-arbre T_k permettant d'améliorer ces premières estimations. Le sous-arbre T_k est sélectionné en utilisant le même critère que pour choisir le taxon *k* lors de l'estimation des distances initiales, et la distance δ_{ij} est alors ré-estimée en optimisant localement la vraisemblance de $T_i \cup T_j \cup T_k$. On voit donc que l'estimation de la distance entre deux taxons contemporains est un cas particulier de l'estimation de la distance entre deux sous-arbres.

4.2.4 Optimisation locale de la vraisemblance

La première estimation de la distance δ_{ij} qui sépare les sous-arbres T_i et T_j (éventuellement réduits à une seule feuille) se fait en optimisant localement la vraisemblance de $T = T_i \cup T_j$ (Figure 17.a, page 93). En supposant que S_i est la séquence ancestrale, cette vraisemblance est définie par :

$$L(T) = \prod_{s=1}^{l} \sum_{b \in \{A,C,G,T\}} \left[\pi_b L(S_i^s = b; T_i) \sum_{c \in \{A,C,G,T\}} P_{bc}(\delta_{ij}) L(S_j^s = c; T_j) \right] \qquad \textbf{(34)}$$

La seconde estimation se fait ensuite en optimisant localement la vraisemblance de $T = T_i \cup T_j \cup T_k$ (Figure 17.b, page 93). En supposant que S_a est la séquence ancestrale, cette vraisemblance vaut :

$$L(T) = \prod_{s=1}^{l} \sum_{b \in \{A,C,G,T\}} \left[\pi_b \prod_{x \in \{i,j,k\}} \sum_{c \in \{A,C,G,T\}} P_{bc}(\delta_{ax}) L(S_x^s = c; T_x) \right] \qquad \textbf{(35)}$$

Rappelons que l'on appelle vecteur de vraisemblance de T_x l'ensemble des valeurs de type $L(S_x^s = c; T_x)$ et que l'on note ce vecteur $LV(T_x)$. Pour pouvoir calculer les vraisemblances définies par les deux équations ci-dessus, il est nécessaire de connaître les vecteurs de vraisemblance de T_i, T_j, et T_k. Dans les méthodes d'agglomération, chaque sous-arbre est initialement constitué d'un seul taxon, et son vecteur de vraisemblance est donc complètement défini. Après chaque agglomération, les longueurs δ_{1i} et δ_{2i} sont estimées (équation (15), page 38), puis le vecteur de vraisemblance du nouveau sous-arbre T_i est calculé en utilisant $LV(T_1)$, $LV(T_2)$ et les longueurs δ_{1i} et δ_{2i}. Après cette agglomération, $LV(T_1)$ et $LV(T_2)$ deviennent inutiles, l'espace mémoire nécessaire pour stocker les vecteurs

de vraisemblance est donc du même ordre ($O(nl)$) que celui nécessaire pour stocker les séquences étudiées.

Les techniques d'optimisation numériques que nous utilisons pour optimiser localement la vraisemblance de $T_i \cup T_j$ et de $T_i \cup T_j \cup T_k$ s'appuient sur la méthode de Brent et sont décrites de manière détaillées dans l'annexe 3 (Ranwez et Gascuel 2002).

4.3 Apports de TripleML

Ce mode d'estimation des distances peut être utilisé avec n'importe quelle variante de NJ. Pour cela, il suffit d'utiliser TRIPLEML pour estimer les distances présentes dans la matrice de distances, au lieu d'utiliser les estimations de la méthode de distances en question. La seule différence entre NJ et BIONJ est précisément la manière dont ces distances sont estimées. L'utilisation de TRIPLEML avec NJ ou BIONJ produit donc le même algorithme que nous appelons NJ+TRIPLEML. La méthode obtenue en combinant TRIPLEML avec WEIGHBOR produit un algorithme que nous appelons WEIGHBOR+TRIPLEML. Il est également possible d'utiliser notre approche uniquement pour estimer la matrice des distances initiales. Les méthodes obtenues en utilisant cette matrice de distances avec NJ, BIONJ et WEIGBOR sont respectivement appelées : NJ+3DIST, BIONJ+3DIST et WEIGHBOR+3dist.

Dans cette partie, nous résumons les trois principaux avantages liés à l'utilisation de TRIPLEML. Premièrement, nous avons effectué de nombreuses simulations informatiques qui nous ont permis de vérifier que l'utilisation de TRIPLEML augmente sensiblement la fiabilité des arbres reconstruits. Deuxièmement, lorsque l'on utilise une méthode de distances agglomérative telle que NJ, le fait d'utiliser TRIPLEML pour estimer les distances évolutives n'augmente pas la complexité globale de cette méthode. Troisièmement, des simulations nous ont permis de vérifier que l'augmentation des temps de calcul engendrée par l'utilisation de TRIPLEML est suffisamment faible pour que de très grands jeux de données puissent être traités.

L'utilisation de TRIPLEML permet donc d'augmenter la fiabilité des arbres reconstruits par les méthodes de distances tout en conservant des temps de calcul raisonnables.

4.3.1 Amélioration de la fiabilité des arbres reconstruits

Nous avons comparé la fiabilité des arbres reconstruits par NJ, BIONJ et WEIGHBOR à celle des arbres reconstruits en utilisant ces méthodes avec 3DIST ou avec TRIPLEML. Nous avons également comparé ces différentes méthodes à FASTDNAML. Pour toutes ces comparaisons, nous avons utilisé les mêmes jeux de tests que ceux qui nous ont servis lors de l'étude des méthodes de quadruplets. De plus, afin d'obtenir une vision plus synthétique des performances de ces différentes méthodes et de les tester pour des conditions d'évolution plus variées, nous avons également utilisé des jeux de données obtenus à partir de 5000 arbres générés aléatoirement. La partie "data sets" de l'article (Ranwez et Gascuel

2002), fournit en annexe 3, détaille la manière dont nous avons généré ces jeux de tests. Les Tableaux 1 et 2 de cette annexe fournissent les résultats obtenus sur ces différents jeux de tests. Les résultats consignés dans ces tableaux sont étudiés en détail dans la partie "*topological accuracy*" de ce même article. Nous reprenons ici les conclusions de ces analyses.

Les tests sur les 5000 arbres aléatoires confirment que les résultats obtenus par BIONJ et WEIGHBOR sont meilleurs que ceux obtenus par NJ. Pour les trois méthodes, l'utilisation des distances calculées par 3DIST permet de réduire la proportion de branches mal inférées. On note par exemple, que les performances de NJ+3DIST sont équivalentes à celle de BIONJ, et que celles de BIONJ+3DIST sont équivalentes à celle de WEIGHBOR. Comme nous le détaillerons dans la partie suivante, ces améliorations sont obtenues avec une très faible augmentation des temps de calcul. L'utilisation de la version complète de TRIPLEML permet d'augmenter de manière plus importante les performances de ces méthodes de distances. L'utilisation de TRIPLEML combiné avec NJ ou avec WEIGHBOR fournit des méthodes de reconstruction phylogénétique dont les performances se situent à mi-chemin entre celles de NJ et celles de FASTDNAML.

Les résultats obtenus sur les jeux de données basés sur des arbres modèles "typiques", permettent de mettre en évidence les conditions qui influencent la fiabilité des arbres reconstruits par les différentes méthodes testées. En particulier, ces tests soulignent le fait que pour des jeux de données respectant l'hypothèse de l'horloge moléculaire, les arbres reconstruits par BIONJ et WEIGHBOR sont à peine plus fiables que ceux reconstruits par NJ. L'écart entre ces deux méthodes et NJ est surtout sensible lorsque l'écart à l'horloge moléculaire est très important. Par contre, l'écart entre NJ+TRIPLEML et NJ est important même lorsque l'hypothèse de l'horloge moléculaire est respectée.

4.3.2 Une complexité globalement inchangée

Pour utiliser une méthode de distances, il est nécessaire d'estimer les distances évolutives séparant chaque paire de séquences contemporaines. Nous considérons donc dans cette partie que la complexité globale d'une méthode de distances inclut à la fois le calcul des distances initiales et la reconstruction d'une phylogénie à partir de ces distances. Contrairement aux méthodes de distances classiques, l'estimation des distances initiales fait partie intégrante de l'algorithme NJ+TRIPLEML. Cet algorithme est détaillé dans la Figure 2 de l'article (Ranwez et Gascuel 2002), qui constitue l'annexe 3 de cette thèse.

Dans le cas de NJ, le calcul des distances initiales est en $O(n^2l)$, et la reconstruction à partir de cette matrice est en $O(n^3)$. La complexité totale en temps est donc en $O(n^2l+n^3)$. La partie "*time complexity analysis*" de l'article (Ranwez et Gascuel 2002) montre que pour NJ+TRIPLEML, le calcul des distances initiales est en $O(n^2l+n^3)$, tout comme l'étape de reconstruction. Ainsi, bien que l'utilisation de TRIPLEML augmente la complexité en temps de chaque étape, la complexité totale de NJ et de NJ+TRIPLEML est la même et vaut $O(n^2l+n^3)$.

Dans la partie 4.2.4, nous avons vu que l'espace mémoire nécessaire à TRIPLEML pour stocker les vecteurs de vraisemblance est du même ordre ($O(nl)$) que celui requis pour stocker les séquences étudiées lors du calcul de la matrice de distances initiales. L'espace mémoire global requis par NJ+TRIPLEML est donc du même ordre que celui requis par NJ.

4.3.3 Des temps de calculs raisonnables

Afin d'avoir un ordre de grandeur des temps de calcul des différentes méthodes étudiées, nous les avons testées sur des jeux de données de taille variable. Les temps de calcul nécessaires à FASTDNAML pour traiter ces jeux de données ainsi que ceux nécessaires à différentes méthodes de distances utilisant, ou non, 3DIST et TRIPLEML sont résumés dans le tableau 3 de (Ranwez et Gascuel 2002). Ces temps de calcul sont en partie spécifiques aux jeux de données testés. Ils doivent donc être uniquement considérés comme des indicateurs de la taille des jeux de données qu'une méthode est capable de traiter.

Ces simulations montrent clairement que l'utilisation de 3DIST n'augmente que très faiblement les temps de calcul. Même si l'utilisation de TRIPLEML augmente de manière plus significative les temps de calculs, NJ+TRIPLEML et WEIGHBOR+TRIPLEML sont cependant capables de traiter de grands jeux de données (plusieurs centaines de séquences) en un temps raisonnable. En effet, malgré les écarts de temps de calcul qui existent entre les différentes méthodes de distances, toutes sont visiblement capables de traiter des jeux de données beaucoup plus importants que ceux testés. En revanche, FASTDNAML semble près de sa limite puisque pour traiter le plus grand de nos jeux de données, il met déjà plus de six heures alors que les méthodes de distances nécessitent toutes moins de 3 minutes.

4.4 Discussion, perspectives

Nos travaux montrent qu'en améliorant l'estimation des distances évolutives séparant deux sous-arbres (éventuellement réduits à un seul taxon), on améliore significativement les performances des méthodes de distances. En particulier, l'utilisation d'un troisième sous-arbre, sélectionné avec soin, permet d'améliorer l'estimation des distances évolutives et donc les performances des méthodes de distances. Nous étudions actuellement plusieurs variantes de TRIPLEML afin d'améliorer encore ces estimations et donc probablement les performances de NJ+TRIPLEML. Une piste possible est d'utiliser une approche en deux étapes. Dans un premier temps un arbre serait reconstruit par NJ. Cela permettrait, ensuite, d'utiliser NJ+TRIPLEML en choisissant, pour chaque distance, le troisième sous-arbre également parmi ceux présents dans l'arbre reconstruit initialement par NJ. Une autre piste possible consiste à détecter les distances les plus difficiles à estimer. Cela pourrait permettre d'améliorer l'estimation de ces distances en s'appuyant sur une optimisation locale de quadruplets. Nous pensons cependant que l'apport de TRIPLEML est déjà important, et il nous semble donc souhaitable d'étendre cette approche de manière à pouvoir prendre en compte d'autre modèles d'évolutions.

Dans un premier temps, nous expliquons la manière dont TRIPLEML peut facilement être adapté pour prendre en compte les variations de vitesses d'évolutions pouvant exister entre différents sites ou pour traiter des séquences d'acides aminés. Nous présentons ensuite les deux pistes que nous étudions actuellement afin d'améliorer l'estimation des distances évolutives.

4.4.1 D'autres modèles d'évolutions

TRIPLEML peut utiliser n'importe quel modèle d'évolution pour lequel il est possible d'optimiser localement la vraisemblance des arbres $T = T_i \cup T_j$ et $T = T_i \cup T_j \cup T_k$. Dans TRIPLEML, ces optimisations sont faites en utilisant la méthode de Brent. Il n'est donc pas nécessaire de savoir calculer la dérivé ou la dérivé seconde de $L(T)$. La seule condition pour pouvoir utiliser un modèle particulier dans TRIPLEML est donc de savoir calculer récursivement la vraisemblance d'un arbre T pour ce modèle, ce qui permet d'adapter les équations (34) et (35) au modèle considéré.

Nous avons vu, dans le chapitre 2 (§ 2.4.2.2), que l'on peut utiliser une version discrète de la loi gamma pour modéliser la distribution des vitesses d'évolution parmi les sites et que le paramètre de cette loi peut être estimé en même temps que les autres paramètres du modèle d'évolution GTR. Dans ce cas, si l'on note π_λ la probabilité qu'un site évolue à une vitesse correspondant au facteur multiplicatif λ, et que l'on note Δ l'ensemble des λ possibles, alors la vraisemblance $L(\Delta, T)$ d'un arbre $T = T_i \cup T_j$ pour cette modélisation des vitesses est :

$$L(\Delta,T) = \prod_{s=1}^{l} \sum_{\lambda\in\Delta} \left(\pi_\lambda \sum_{b\in\{A,C,G,T\}} \left[\pi_b L(S_i^s = b;\lambda T_i) \sum_{c\in\{A,C,G,T\}} P_{bc}(\lambda\delta_{ij}) L(S_j^s = c;\lambda T_j) \right] \right) \quad \textbf{(36)}$$

et la vraisemblance $L(\Delta, T)$ d'un arbre $T = T_i \cup T_j \cup T_k$ est :

$$L(\Delta,T) = \prod_{s=1}^{l} \sum_{\lambda\in\Delta} \left(\pi_\lambda \sum_{b\in\{A,C,G,T\}} \left[\pi_b \prod_{x\in\{i,j,k\}} \sum_{c\in\{A,C,G,T\}} P_{bc}(\lambda\delta_{ax}) L(S_x^s = c;\lambda T_x) \right] \right) \quad \textbf{(37)}$$

La seule modification notable pour prendre en compte la variabilité des vitesses d'évolution entre sites est donc l'augmentation du nombre de vecteurs de vraisemblance. En effet, au lieu d'avoir un vecteur de vraisemblance par sous-arbres, on a désormais un vecteur de vraisemblance par sous-arbre et par vitesse d'évolution. Les complexités (en temps et en espace) de NJ+TRIPLEML dépendent alors également du nombre de vitesses d'évolution possibles, mais en général ce nombre est faible (<10).

On peut aussi facilement étendre TRIPLEML pour prendre en compte des séquences d'acides aminés. En s'appuyant sur les taux de substitution instantanés issus des mesures empiriques de (Dayhoff, Schwartz et Orcutt 1978) (matrice PAM), Kishino, Miyata et Hasegawa (1990) ont développé un modèle d'évolution pour les séquences d'acides aminés.

Ce modèle, dit de Dayhoff, est analogue au modèle GTR utilisé pour les séquences d'ADN. Les paramètres du modèle de Dayhoff sont fixés à partir des mesures empiriques obtenues sur de grands jeux de données et la vraisemblance d'un arbre ne dépend que des longueurs de ses branches. On peut ainsi facilement étendre les formules (36) et (37) (page 98) au cas des acides aminés, il suffit simplement de faire varier *b* et *c* non plus sur l'ensemble des 4 nucléotides possibles, mais sur l'ensemble des vingt acide aminés possibles.

4.4.2 Elargir le choix du troisième sous-arbre

Pour l'estimation de certaines distances initiales, il pourrait parfois être plus intéressant d'estimer cette distance en utilisant un sous-arbre plutôt qu'un troisième taxon. De manière générale, dans TRIPLEML, l'estimation de la distance séparant deux sous-arbres (éventuellement réduits à un seul taxon) se fait en utilisant un troisième sous-arbre choisi parmi ceux qui, à l'étape courante, ont déjà été construits par agglomération. Dans certains cas, il pourrait être intéressant d'estimer ces distances en utilisant d'autres sous-arbres.

Une manière possible pour que NJ+TRIPLEML dispose d'un ensemble plus large de sous-arbres lors de l'estimation des distances, consiste à adopter une approche en deux temps. Dans un premier temps, on reconstruit une première phylogénie T_1 en utilisant NJ, puis l'on reconstruit une phylogénie T en utilisant une variante de NJ+TRIPLEML qui, pour chaque estimation, choisit le troisième sous-arbre non seulement parmi ceux qu'elle à déjà reconstruits par agglomération mais également parmi les sous-arbres de T_1. Pour cela, il faut disposer des vecteurs de vraisemblance de l'ensemble des sous-arbres de T_1. Comme nous l'avons vu au chapitre 2 (§ 2.4.3.2), l'ensemble de ces vecteurs peut se calculer en $O(nl)$ grâce au double parcours récursif. L'espace mémoire nécessaire pour les stocker est proportionnel à $O(nl)$, et le seul changement dans NJ+TRIPLEML est que le troisième sous-arbre est choisi parmi un ensemble plus grand, mais dont la taille reste proportionnelle à $O(n)$. A priori, cette modification ne change donc pas la complexité de NJ+TRIPLEML.

Cependant, la difficulté liée à cette approche, vient du fait qu'il n'est pas raisonnable d'utiliser n'importe quel sous-arbre T_k de T_1 pour estimer la distance δ_{ij} séparant deux sous-arbres T_i et T_j. Par exemple, les séquences associées aux feuilles de T_k ne doivent pas être présentes aux feuilles de T_i ou de T_j. Cette contrainte est relativement simple, mais il faut certainement que T_k vérifie d'autres propriétés plus complexes, qui ne sont pas uniquement liées à T_i et T_j. Par exemple, si dans les sous-arbres construits par NJ+TRIPLEML, il existe un sous-arbre ayant les mêmes feuilles que T_k mais une topologie différente, est-il raisonnable d'utiliser T_k pour estimer δ_{ij} ? De manière plus générale, il semble important de vérifier que le sous-arbre T_k est *cohérent* avec les sous-arbres obtenus par les agglomérations de NJ+TRIPLEML. Nous essayons actuellement de définir correctement cette notion de cohérence, afin de pouvoir garantir, que les troisièmes sous-arbres utilisés sont cohérents avec les agglomérations déjà effectuées sans augmenter la complexité de NJ+TRIPLEML.

4.4.3 Au-delà des triplets

Lorsque l'on cherche à estimer la distance δ_{ij} séparant deux sous-arbres T_i et T_j (éventuellement réduits à un seul taxon), il est possible, si cette distance est très grande, que l'utilisation d'un troisième sous-arbre soit insuffisante pour obtenir des branches de longueur assez faible pour être estimée précisément. Inversement, pour les distances très faibles, l'utilisation d'un troisième sous-arbre augmente les temps de calcul sans réellement améliorer la fiabilité de l'estimation. Il serait donc intéressant de généraliser l'approche de NJ+TRIPLEML de manière à pouvoir estimer le nombre de sous-arbres qui sont nécessaires pour estimer correctement une distance particulière. Cette idée est séduisante, mais elle pose plusieurs problèmes sur lesquels nous réfléchissons actuellement.

Le premier problème est d'arriver à définir, lors de chaque estimation, les sous-arbres qui doivent être pris en compte. En effet, le nombre optimal de sous-arbres devant être pris en compte ne dépend pas uniquement de la distance que l'on cherche à estimer. Par exemple, si deux taxons constituent une paire externe, il est inutile d'essayer de couper la branche reliant ces taxons en plusieurs points. Il est donc probablement inutile d'utiliser plusieurs sous-arbres pour estimer cette distance. De plus, si l'on souhaite utiliser deux sous-arbres T_k et T_l pour estimer la distance séparant T_i et T_j, on va sélectionner T_k et T_l de manière à ce que la branche (i, j) soit scindée en trois branches ayant approximativement la même longueur. Par contre, si l'on utilise un seul sous-arbre pour estimer cette distance, on souhaite couper la branche (i, j) en son milieu, le sous-arbre retenu dans ce cas ne sera donc certainement ni T_k ni T_l. Cela pose deux types de difficultés. D'une part, il semble inévitable de définir des critères différents suivant le nombre de sous-arbres que l'on veut sélectionner. Les critères étant différents, ils seront certainement difficiles à comparer. D'autre part, même si l'on connaît les meilleurs ensembles de x sous-arbres permettant d'estimer δ_{ij}, cela ne semble pas donner d'information sur les ensembles de $(x+1)$ sous-arbres permettant d'estimer au mieux δ_{ij}. Il faut donc, à priori, essayer toutes les valeurs de x possibles pour pouvoir trouver le meilleur ensemble de sous-arbres qui permette d'estimer une distance particulière. Une approche plus raisonnable consiste à limiter les valeurs possibles de x à 0, 1 ou 2, mais même ainsi, nous n'avons pas encore trouvé de solution satisfaisante pour choisir les sous-arbres à prendre en compte.

Le second problème est d'arriver à mettre cette approche en place sans augmenter la complexité de NJ+TRIPLEML ou du moins en conservant des temps de calcul raisonnables. En effet, même en se limitant au cas où l'on considère au plus deux autres sous-arbres ($x \leq 2$), on augmente la complexité de NJ+TRIPLEML. Il suffit pour s'en rendre compte de considérer le calcul des distances initiales. Pour chacune de ces $O(n^2)$ distances, il faut sélectionner les deux autres taxons qui permettent la meilleure estimation de cette distance. Le calcul des distances initiales a donc, a priori, une complexité de $O(n^2 l + n^4)$. Comme pour les méthodes de quadruplets, cette complexité risque de limiter une telle approche à des jeux de données de taille beaucoup plus faible que ceux que peuvent traiter les autres méthodes de distances. Il est donc important que l'utilisation d'un quatrième taxon reste

exceptionnelle et que l'on puisse juger de sa nécessité, même de manière approximative, sans avoir à considérer l'ensemble des quadruplets possibles.

4.5 Conclusion

Ce chapitre présente TRIPLEML, une méthode qui permet d'obtenir de meilleures estimations des distances évolutives. Cette nouvelle approche utilise un processus analogue pour estimer les distances initiales et pour réduire la matrice de distances utilisée par NJ et ses variantes. Dans les deux cas, les distances sont estimées à partir d'une optimisation locale de la vraisemblance basée sur les triplets de taxons (ou de groupes de taxons). La combinaison de TRIPLEML avec NJ ou WEIGHBOR fournit des méthodes rapides dont la capacité à reconstruire la bonne phylogénie est bien meilleure que celle des méthodes de distances usuelles. Les méthodes ainsi obtenues ont des performances intermédiaires entre celles de NJ utilisé seul et celles de FASTDNAML.

Nous avons également présenté une variante de TRIPLEML, que nous appelons 3DIST, et qui n'utilise notre mode d'estimation des distances que pour calculer les distances initiales. Nos tests indiquent que 3DIST augmente de manière sensible les performances des méthodes de distances. De plus l'utilisation de 3DIST n'augmente quasiment pas les temps de calcul et ne nécessite pas de modifier la méthode de distances avec laquelle on la combine. L'augmentation des performances est moins spectaculaire qu'avec TRIPLEML, mais 3DIST est mieux adapté lorsque l'on traite de très grands jeux de données, contenant plusieurs milliers de séquences.

Ces travaux montrent clairement l'intérêt de combiner les méthodes de distances et les méthodes de maximum de vraisemblance pour proposer de nouvelles méthodes fiables capables de traiter de grands jeux de données. Il existe certainement de nombreuses autres manières de les combiner efficacement, ce qui ouvre de nouvelles perspectives de recherches.

Conclusion

Dans cette thèse, nous avons étudié deux types de méthodes visant à reconstruire de manière efficace de grandes phylogénies suivant le principe du maximum de vraisemblance : les méthodes de quadruplets (chapitre 3) et les méthodes de distances (chapitre 4).

Ces dernières années, les méthodes de quadruplets ont fait l'objet de très nombreux travaux. Ces méthodes semblaient notamment constituer une approche prometteuse pour obtenir des méthodes de reconstruction phylogénétique capables de traiter de grands jeux de données tout en étant plus fiables que les méthodes de distances. Dans ce cadre, la méthode de quadruplets QP proposée par (Strimmer et Von Haeseler 1996) semblait particulièrement prometteuse, elle semble d'ailleurs, encore actuellement, être la méthode de quadruplets la plus utilisée.

Nos travaux ont permis de mettre en évidence plusieurs faiblesses de cette approche et d'y apporter des solutions. L'algorithme WO (Ranwez et Gascuel 2001a ; Ranwez et Gascuel 2001b) issu de ces améliorations possède de meilleures propriétés théoriques que QP, et nos simulations nous ont permis de vérifier que les arbres inférés par WO sont également plus fiables que ceux inférés par QP.

Cependant, de manière paradoxale, l'apport le plus important de nos travaux, sur les méthodes de quadruplets, est certainement d'avoir montré que ce type de méthode n'est actuellement pas une alternative intéressante aux méthodes de distances (Ranwez et Gascuel 2001b). En effet, elles ont une complexité théorique plus élevée que les méthodes de distances et ne peuvent donc pas traiter des jeux de données aussi grands que ces dernières. De plus, les arbres qu'elles reconstruisent sont moins fiables que ceux obtenus avec les méthodes de distances. Il semble que ces résultats décevants reflètent les limites inhérentes aux méthodes de quadruplets.

Même si les méthodes de quadruplets ne sont pas adaptées pour reconstruire de grandes phylogénies suivant le principe de maximum de vraisemblance, elles ont de nombreuses autres applications. On peut les utiliser, par exemple, pour calculer la distance topologique séparant deux phylogénies, pour définir un arbre consensus ou pour inférer des *super-arbres* à partir de phylogénies portant sur des ensembles disjoints de taxons. Notre algorithme WO est une heuristique efficace pour proposer une phylogénie optimale au sens d'un critère défini à partir des pondérations des 4-arbres. Ainsi, simplement en adaptant la pondération des 4-arbres, WO devient une heuristique permettant d'obtenir un super-arbre ou un arbre consensus. Un travail important reste donc à faire pour mesurer la pertinence

de ce type d'approche (*i.e.* ce type de consensus induit-il un biais topologique?), et l'efficacité de WO dans un tel contexte.

Mises à par les méthodes de quadruplets, les méthodes de distances sont les seules méthodes permettant de reconstruire de grandes phylogénies suivant le principe du maximum de vraisemblance et intégrant un modèle explicite de l'évolution des séquences. En effet, lorsque l'on étudie un ensemble de séquences nucléotidiques, les distances évolutives, utilisées comme données initiales par ces méthodes, sont estimées au sens du maximum de vraisemblance. Les méthodes de distances étant très rapides, et les méthodes de maximum de vraisemblance étant très fiables, il semble naturel d'essayer de combiner ces deux approches de manière à obtenir des méthodes intermédiaires.

Dans cette optique, la méthode NJML (Ota et Li 2000) utilise un arbre reconstruit par NJ pour restreindre l'espace dans lequel est ensuite effectuée la recherche d'un arbre de vraisemblance élevée. Cette approche cherche ainsi à accélérer les méthodes de maximum de vraisemblance en s'appuyant sur une méthode de distances. Nous utilisons l'approche duale qui vise à utiliser de manière plus intensive le maximum de vraisemblance pour augmenter la fiabilité des arbres reconstruits par les méthodes de distances. Dans toutes les variantes de la méthode de distances NJ (Saitou et Nei 1987), il y a deux types d'estimations de distances. L'un concerne l'estimation des distances séparant deux taxons (distances initiales) et, l'autre l'estimation de distances séparants deux groupes de taxons (réduction de la matrice de distances). TRIPLEML estime de manière analogue ces deux types de distances à partir d'une approche de maximum de vraisemblance sur des triplets de séquences (ou de groupes de séquences). L'utilisation de TRIPLEML permet d'augmenter significativement la fiabilité des arbres reconstruits par les différentes variantes de NJ tout en conservant des temps de calcul suffisamment faibles pour pouvoir inférer rapidement de grandes phylogénies. De plus cette approche se généralise facilement à d'autres types de jeux de données et à d'autres modèles d'évolution que le modèle GTR.

Les performances de NJ+TRIPLEML étant très encourageantes, nous travaillons actuellement sur plusieurs pistes visant à l'améliorer. Nous réfléchissons notamment à la manière dont les distances pourraient être estimées en utilisant non plus simplement des triplets mais des *n*-uplets. La difficulté est alors de pouvoir choisir automatiquement, pour chaque estimation d'une distance, le *n*-uplet le mieux adapté sans augmenter la complexité en temps ou en espace de notre approche. L'autre piste est d'utiliser NJ pour reconstruire un premier arbre T_1, et d'utiliser ensuite les sous-arbres de T_1 pour améliorer l'estimation des distances faite par NJ+TripleML. Cette approche semble plus facile à mettre en œuvre, la seule difficulté restante provient du fait que tous les sous-arbres de T_1 ne peuvent pas être utilisés pour estimer toutes les distances estimées dans NJ+TripleML. Il nous reste donc à définir un test permettant, sans augmenter la complexité en temps de notre approche, de déterminer les sous-arbres de T_1 qui peuvent servir à estimer une distance particulière dans NJ+TripleML.

Nos travaux (Ranwez et Gascuel 2002) et ceux de (Ota et Li 2000) montrent, sous des angles différents, que les approches mixtes, qui combinent méthodes de distances et maximum de vraisemblance, constituent une manière efficace de reconstruire de grandes phylogénies. Ces deux méthodes ouvrent une voie nouvelle, des plus prometteuses, qui reste à explorer.

Table des figures

Bibliographie

Adachi, J. et M. Hasegawa (1996). "MOLPHY version 2.3".

Adachi, J. et M. Hasegawa (1999). "Instability of quartet analyses of molecular sequance data by the maximum likelihood method: the cetacea/artiodactyla relashionships." Cladistics **5**: 164-166.

Bandelt, H.-J. et A. Dress (1986). "Reconstructing the Shape of a Tree from Observed Dissimilarity Data." Advances in Applied Mathematics **7**: 309-343.

Berge, C. (1970). Graphes et Hypergraphes. Paris, Dunod.

Berry, V. (1997). "Méthodes et algorithmes pour reconstruire les arbres de l'évolution". Montpellier, Thèse de doctorat en informatique de l'Université Montpellier II.

Berry, V. et O. Gascuel (2000). "Inferring Evolutionary Trees with Strong Combinatorial Evidence." Theoretical Computer Science **240**: 271-298.

Berry, V., P. Kearney, M. Li, T. Jiang, et al. (1999). "Quartet Cleaning: Improved Algorithms and Simulations". European Symposium on Algorithms, Prague, Springer.

Brodal, G. S., R. Fagerberg et N. S. Pedersen (2001). "Computing the Quartet Distance Between Evolutionary Trees". international symposium on algorithms and computation, Christchurch, New Zealand, springer (LNCS).

Bruno, W. J., N. D. Socci et A. L. Halpern (2000). "Weighted neighbor joining: a likelihood-based approach to distance-based phylogeny reconstruction." Molecular Biology and Evolution **17**(1): 189-197.

Bryant, D. et M. A. Steel (1995). "Extension operations on sets of leaf-labelled trees." Advances in Applied Mathematics **16**(4): 425-453.

Bulmer, M. (1991). "Use of the method of generalized least squares in reconstructing phylogenies from sequence data." Molecular Biology and Evolution **8**: 868-883.

Buneman, P. (1971). Mathematics in Archeological and Historical Sciences, Edhinburgh University Press.

Caraux, G., O. Gascuel, G. Andrieux et D. Levy (1995). "Approches informatiques de la reconstruction phylogénétique." Technique et Science Informatiques **14**(2): 113-139.

Chor, B., M. D. Hendy, B. R. Holland et D. Penny (2000). "Multiple Maxima of Likelihood in Phylogenetic Trees: an Analytical Approach." Molecular Biology and Evolution **17**(10): 1529-1541.

Cox, D. R. et H. D. Miller (1977). The Theory of Stochastic Processes, Chapman & Hall.

Darwin, C. (1859). On the origin of species, John Murray.

Dayhoff, M. O., R. M. Schwartz et B. C. Orcutt (1978). "A model of evolutionary change in proteins". Atlas of Protein Sequence and Structure. M. O. Dayhoff. Washington D.D., National Biomedical Research Foundation. **5:** 345-352.

De Groot, M. H. (1989). Probability and statistics. New York, Addidon-Wesley.

Elemento, O. et O. Gascuel (2002). "An efficient and accurate distance based algorithm to reconstruct tandem duplication trees". European Conference on Computational Biology (ECCB), Saarbrücken, Germany, à paraître dans Bioinformatique.

Erdös, P., M. A. Steel, L. A. Székely et T. Warnow (1997a). "Constructing big trees from short sequences." Lecture Notes in Computer Science **1256**: 827-837.

Erdös, P., M. A. Steel, L. A. Székely et T. Warnow (1997b). "A few logs suffice to build (almost) all trees: Part I", DIMACS.

Felsenstein, J. (1981). "Evolutionary trees from DNA sequences: a maximum likelihood approach." Journal of Molecular Evolution **17**(6): 368-376.

Felsenstein, J. (1993). "PHYLIP (phylogeny inference package) version 3.5c".

Fitch, W. M. (1971). "Toward defining the course of evolution : minimum change for a specific tree topology." Systematic Zoology **20**: 406-416.

Fitch, W. M. et E. Margoliash (1967). "Construction of phylogenetic trees." Science **155**(760): 279-284.

Foulds, L. R. et R. L. Graham (1982). "The Steiner problem in phylogeny is NP-complete." Advances in Applied Mathematics **3**: 43-49.

Galtier, N. (1997). "L'approche statistique en phylogénie moléculaire : influence des compositions en bases variables". Lyon, Thèse de doctorat en biologie de l'Université Claude Bernard.

Galtier, N., N. J. Tourasse et M. Gouy (1999). "A non-hyperthermophilic ancestor to extant life forms." Science **283**: 220-221.

Gascuel, O. (1994). "A note on Sattath and Tversky's, Saitou and Nei's, and Studier and Keppler's algorithms for inferring phylogenies from evolutionary distances." Molecular Biology and Evolution **11**(6): 961-963.

Gascuel, O. (1997). "BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data." Molecular Biology and Evolution **14**(7): 685-695.

Gillespie, J. H. (1984). "The molecular clock may be an episodic clock." Proceedings of the National Academy of Sciences of the United States of America **81**: 8009-8013.

Harvey, P. H., R. M. May et S. Nee (1996). New uses for new phylogenies. Oxford, Oxford University Press.

Hasegawa, M., H. Kishino et T. Yano (1985). "Dating of the human-ape splitting by a molecular clock of mitochondrial DNA." Journal of Molecular Evolution **22**(2): 160-174.

Hendy, M. D., D. Penny et M. A. Steel (1994). "A discrete Fourier analysis for evolutionary trees." Proceedings of the National Academy of Sciences of the United States of America **91**(8): 3339-3343.

Jiang, T. (1998). "Orchestrating Quartets: Approximation and Data Correction". 39th Annual Symposium on Foundations of Computer Science, Palo Alto, California.

Jones, D. T., W. R. Taylor et J. M. Thornton (1992). "The rapid generation of mutation data matrices from protein sequences." Computer Applications in the Biosciences **8**: 197-203.

Jukes, T. H. et C. R. Cantor (1969). "Evolution of protein molecules". Mammalian protein metabolism. H. M. Munro. New York, Academic Press.

Kelsey, C. R., K. A. Crandall et A. F. Voevodin (1999). "Different models, different trees: the geographic origin of PTLV-I." Molecular Phylogenetics and Evolution **13**(2): 336-347.

Kimura, M. (1980). "A simple method for estimating evolutionary of base substitution through comparative studies of nucleotide sequences." Journal of Molecular Evolution **16**: 111-120.

Kimura, M. (1981). "Estimation of evolutionary distances between homologous nucleotide sequences." Proceedings of the National Academy of Sciences of the United States of America **78**: 454-458.

Kishino, H. T., T. Miyata et M. Hasegawa (1990). "Maximum likelihood inference of protein phylogeny and the origin of chloroplasts." Journal of Molecular Evolution **31**: 151-160.

Kumar, S. (1996). "A stepwise algorithm for finding minimum evolution trees." Molecular Biology and Evolution **13**(4): 584-593.

Lamarck, J. B. (1809). Philosophie zoologique.

Lanave, C., G. Preparata, C. Saccone et G. Serio (1984). "A new method for calculating evolutionary substitution rates." Journal of Molecular and Applied Genetics **20**(1): 86-93.

Margush, T. et F. R. McMorris (1981). "Consensus n-trees." Bulletin of Mathematical Biology **43**: 239-244.

Mullis, K. F. et F. A. Faloona (1987). "Specific synthesis of dna in vitro via polymerase catalysed chained reaction." Methods in Enzymology **155**: 335-350.

Nei, M. et L. Jin (1989). "Variances of the Average Numbers of Nucleotide Substitutions Within and Between Populations." Molecular Biology and Evolution **6**(3): 290-300.

Olsen, G. J. (1987). "Earliest phylogenetic branchings: comparing rRNA-based evolutionary trees inferred with various techniques." Cold Spring Harbor Symposia on Quantitative Biology **52**: 825-837.

Olsen, G. J., H. Matsuda, R. Hagstrom et R. Overbeek (1994). "fastDNAmL: a tool for construction of phylogenetic trees of DNA sequences using maximum likelihood." Computer Applications in the Biosciences **10**(1): 41-48.

Ota, S. et W. H. Li (2000). "NJML: a hybrid algorithm for the neighbor-joining and maximum-likelihood methods." Molecular Biology and Evolution **17**(9): 1401-1409.

Page, R. D. M. et E. C. Holmes (1998). Molecular Evolution: A Phylogenetic Approach. London, Blackwell Science.

Penny, D. (1982). "Towards a basis for classification: the incompleteness of distance measures, incompatibility analysis and phenetic classification." Journal of Theoretical Biology **96**(2): 129-142.

Philippe, H. et E. Douzery (1994). "The pitfalls of molecular phylogeny based on four species, as illustrated by the Cetacea/Artiodactyla relationship." Journal of Mammal Evolution **2**: 133-152.

Posada, D. et K. A. Crandall (2001). "A comparison of different strategies for selecting models of DNA substitution." Systematic Biology **50**: 580-601.

Press, W. H., B. P. Flannery, S. A. Teukolsky et W. T. Vetterling (1988). Numerical Recipies in C: The Art of scientific computing, Cambridge University Press.

Ranwez, V. et O. Gascuel (2001a). "Phylogenetic Reconstruction Algorithms Based on Weighted 4-Trees." Lecture Notes in Computer Science **2066**: 337-348.

Ranwez, V. et O. Gascuel (2001b). "Quartet-based phylogenetic inference: improvements and limits." Molecular Biology and Evolution **18**(6): 1103-1116.

Ranwez, V. et O. Gascuel (2002). "Improvement of distance-based phylogenetic methods via a local maximum likelihood approach using triplets." Molecular Biology and Evolution (à paraitre).

Robinson, D. F. et L. R. Foulds (1981). "Comparison of phylogenetic trees." Mathematical Biosciences **53**: 131-147.

Rodriguez, F., J. L. Oliver, A. Marin et J. R. Medina (1990). "The general stochastic model of nucleotide substitution." Journal of Theoretical Biology **142**: 485-501.

Rzhetsky, A. et M. Nei (1995). "Tests of applicability of several substitution models for DNA sequence data." Molecular Biology and Evolution **12**(1): 131-151.

Saitou, N. et M. Nei (1987). "The neighbor-joining method: a new method for reconstructing phylogenetic trees." Molecular Biology and Evolution **4**(4): 406-425.

Sattath, S. et A. Tversky (1977). "Additive similarity trees." Psychometrika **42**: 319-345.

Smolenskii, Y. A. (1969). "A method for linear recoding of graphs." Computational Mathematics and Mathematical Physics **2**: 396-397.

Steel, M. A. (1992). "The complexity of reconstructing trees from qualitative characters and subtrees." Journal of Classification **9**: 91-116.

Strimmer, K., N. Goldman et A. Von Haeseler (1997). "Bayesian probabilities and Quartet Puzzling." Molecular Biology and Evolution **14**: 210-211.

Strimmer, K. et A. Von Haeseler (1996). "Quartet puzzling: A quartet maximum-likelihood method for reconstructing tree topologies." Molecular Biology and Evolution **13**(7): 964-969.

Studier, J. A. et K. J. Keppler (1988). "A note on the neighbor-joining algorithm of Saitou and Nei." Molecular Biology and Evolution **5**(6): 729-731.

Sullivan, J. et D. L. Swofford (1997). "Are guinea pigs rodents? The importance of adequate models in molecular phylogenies." Journal of Mammal Evolution **4**: 77-86.

Swofford, D. L. (2002). "PAUP 4.0 : Phylogenetic Analysis Using Parsimony (and Other Methods)". Sunderland, Massachusetts, Sinauer Associates.

Swofford, D. L., G. J. Olsen, P. J. Waddell et D. M. Hillis (1996). "Phylogenetic inference". Molecular Systematics. M. D. Hillis, C. Moritz et B. K. Mable. Massachusetts, Sinauer Associates.

Tajima, F. et M. Nei (1982). "Biases of the estimates of DNA divergence obtained by the restriction enzyme technique." Journal of Molecular Evolution **18**: 115-121.

Tamura, K. et M. Nei (1993). "Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees." Molecular Biology and Evolution **10**(3): 512-526.

Willson, S. J. (1999). "Building phylogenetic trees from quartets by using local inconsistency measure." Molecular Biology and Evolution **16**: 685-693.

Woese, C. R. et G. E. Fox (1977). "Phylogenetic structure of the procaryotic domain: the primary kingdoms." Proceedings of the Academy of Natural Sciences of USA **74**: 5088-5090.

Yang, Z. (1993). "Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites." Molecular Biology and Evolution **10**(6): 1396-1401.

Yang, Z. (1994). "Estimating the pattern of nucleotide substitution." Journal of Molecular Evolution **39**(1): 105-111.

Yang, Z. (1996). "Among-site rate variation and its impact on phylogenetic analyses." Trends in Ecology and Evolution **11**: 367-372.

Yang, Z., N. Goldman et A. Friday (1994). "Comparison of models for nucleotide substitution used in maximum-likelihood phylogenetic estimation." Molecular Biology and Evolution **11**(2): 316-324.

Zarestkii, K. (1965). "Construction d'un arbre sur la base d'un ensemble de distances entre ces feuilles." Uspehi Mathematicheskikh Nauk.

Zharkikh, A. et W. H. Li (1995). "Estimation of confidence in phylogeny: the complete-and-partial bootstrap technique." Molecular Phylogenetics and Evolution **4**(1): 44-63.

Zuckerkandl, E. et L. Pauling (1962). "Molecular disease, evolution, and genic heterogeneity". Horizons in Biochemistry. M. Kasha et B. Pullman. New York, Academic Press**:** 189-225.

Annexes

Annexe 1 : (Ranwez et Gascuel 2001a)

Ranwez, V. et O. Gascuel (2001a). "Phylogenetic Reconstruction Algorithms Based on Weighted 4-Trees." Lecture Notes in Computer Science **2066**: 337-348.

Annexe 2 : (Ranwez et Gascuel 2001b)

Ranwez, V. et O. Gascuel (2001b). "Quartet-based phylogenetic inference: improvements and limits." Molecular Biology and Evolution **18**(6): 1103-1116.

Annexe 3 : (Ranwez et Gascuel 2002)

Ranwez, V. et O. Gascuel (2002). "Improvement of distance-based phylogenetic methods via a local maximum likelihood approach using triplets." Molecular Biology and Evolution (à paraitre).