Ultrametri

c Least Squares : une méthode de distances rapide et précise pour estimer le taux de substitution à partir d'un ensemble de séquences hétérochrones

Nous présentons ici une méthode de distances, Ultrametric Least Squares *(ULS), qui estime le taux de substitution d'un ensemble de séquences hétérochrones, en faisant l'hypothèse d'une horloge moléculaire stricte. Cette méthode corrige la distance évolutive entre séquences, par l'adjonction d'un facteur correctif aux séquences non contemporaines. Ce facteur est proportionnel au taux de substitution à estimer, ainsi qu'à l'ancienneté de la séquence en question. Le taux de substitution est alors estimé par la minimisation d'un critère quadratique, qui mesure l'ultramétricité de la distance corrigée. Nous montrons que ce critère est parabolique par morceaux, et proposons un algorithme efficace, en* $O(n^3 \log n)$ *où* n *est le nombre de séquences, pour minimiser ce critère. Nous montrons aussi qu'il est possible de borner cette complexité et sans perte de précision par un tirage aléatoire de triplets. Notre méthode peut être étendue à l'estimation de plusieurs taux de substitution variant au cours du temps, par exemple pour prendre en compte la prise d'un traitement et sa date de début, ou par lignage (horloges moléculaires locales). ULS est confrontée sur données simulées à d'autres méthodes de distances, comme sUPGMA ou TREBLE, aux régressions linéaires* Root-to-Tip *et* Pairwise-Distance*, ainsi qu'à l'approche probabiliste développée dans le logiciel BEAST, qui est à l'heure actuelle considérée comme l'une des plus précises mais est handicapée par un temps de calcul très important. Les expériences montrent qu'ULS est plus précise ou aussi précise que les autres méthodes de distances et que BEAST, tout en étant extrêmement rapide. Nous présentons ensuite une application d'ULS sur deux jeux de données du VIH.*

Sommaire

4.1 Introduction

Les méthodes de distances définissent avec les méthodes probabilistes et les méthodes de parcimonie les trois approches principales permettant l'inférence de phylogénies moléculaires (cf. Chapitre 1). Un des principes souvent utilisé avec les méthodes de distances est celui des moindres carrés (en anglais *Least Squares*) qui compare les distances évolutives estimées entre paires de séquences, contenant des erreurs dues à l'échantillonnage et inhérentes au modèle d'évolution, aux distances patristiques (ou distances de chemin) calculées dans l'arbre estimé (Felsenstein, 1997; Bulmer, 1991; Fitch & Margoliash, 1967). Ce principe est non seulement rapide en temps de calcul, mais augmente en précision au fur et à mesure que les erreurs d'estimation dans les distances tendent à disparaître. En pratique, il est impossible d'estimer les vraies distances évolutives puisque les modèles d'évolution font des hypothèses simplificatrices, comme, par exemple, l'indépendance des sites. Pour contrer cet effet, plus marqué sur les grandes distances que sur les petites, les méthodes de moindres carrés utilisent généralement une valeur de pondération devant chaque terme de la somme, qui est inverse à la variance de l'estimateur et donc plus faible pour les grandes distances. Ainsi, les méthodes de moindres carrés pondérées *Weighted Least Squares*, WLS, généralisent la

méthode des moindres carrés ordinaires *Ordinary Least Squares*, OLS, qui n'utilise pas de pondérations, ou, ce qui revient au même, des pondérations toutes identiques.

Cette approche de pondération est peu exploitée (hormis TREBLE) par les méthodes de distances qui permettent d'estimer le taux de substitution, alors qu'elle est presque universelle pour les méthodes d'inférence phylogénétique. À notre connaissance, la méthode sUPGMA est la seule qui utilise le principe des moindres carrés, mais sans valeurs de pondération (OLS) (Rodrigo *et al*, 2007; Drummond & Rodrigo, 2000).

Nous présentons dans ce chapitre une méthode de distances, *Ultrametric Least Squares* (ULS), qui estime le taux de substitution d'un ensemble de séquences hétérochrones sous l'hypothèse d'une horloge moléculaire stricte, c'est-à-dire sous les hypothèses du modèle SRDT. Cette méthode utilise des triplets de séquences et des pondérations, comme TREBLE, mais propose un algorithme radicalement différent où on optimise un critère global dont nous montrons qu'il est parabolique par morceaux. Cette méthode est ensuite étendue aux modèles MRDT et DR (mais avec des horloges moléculaires locales). Les performances de cette méthode sont simultanément comparées avec celles des autres méthodes de distances et celles de la méthode probabiliste de référence, BEAST.

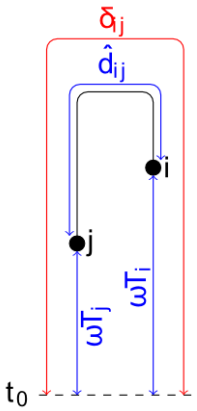
4.2 Description de la méthode

Considérons un ensemble de n séquences homologues alignées $\mathcal{S} = \{S_i, i = 1, \cdots, n\}$, associées aux souches $\mathcal{E} = \{i, i = 1, \cdots, n\}$ échantillonnées aux temps $\mathcal{T} = \{t_i, i = 1, \cdots, n\}$. Soit $d: \mathcal{S} \times \mathcal{S} \to \mathbb{R}_+$ la distance arborée entre ces souches correspondant à la vraie phylogénie (inconnue). Pour simplifier, $d(S_i, S_j)$ est noté d_{ij}. Soit $\hat{d}$ une estimation de d satisfaisant les conditions de symétrie ($\hat{d}_{ij} = \hat{d}_{ji}$), de positivité ($\hat{d}_{ij} \geq 0$ et $\hat{d}_{ii} = 0$) et de réflexivité ($\hat{d}_{ij} = 0 \Leftrightarrow i = j$). On dit que $\hat{d}$ est un indice de distance ou une dissimilarité sur $\mathcal{S}$. L'inégalité triangulaire est une condition non forcément respectée par $\hat{d}$ et non nécessaire ici. Pour tout t, $t' \in \mathcal{T}$, $t < t'$ signifie que t est plus ancien que t' ou que t' est plus récent que t. Soit $t_0 \in \mathcal{T}$ la date d'échantillonnage la plus récente, c'est-à-dire que $t_0 = \max_{i \in \mathcal{E}} t_i$. L'intervalle de temps qui sépare la date d'échantillonnage $t_i \in \mathcal{T}$ de t_0 se note $T_i = t_0 - t_i$ et s'exprime en unité de temps (généralement en jours, années ou générations). Comme $t_0 \geq t_k$, pour tout $k \in \mathcal{E}$, alors $t_0 - t_k = T_k \geq 0$ pour tout $k \in \mathcal{E}$.

Pour estimer le taux de substitution ω, relatif à l'ensemble $\mathcal{S}$, nous « corrigeons » $\hat{d}$, à l'aide de ω, en une mesure δ, fonction du taux de substitution, qui représente une distance ou dissimilarité où chaque souche est vue comme contemporaine (Figure 26). Soient deux souches i et j de $\mathcal{E}$, $i \neq j$, alors

(1) $$\delta_{ij}(\omega) = \hat{d}_{ij} + \omega \times (T_i + T_j).$$

Figure 26. Schéma représentant la définition de l'équation (1).
Soient deux taxa i et j échantillonnés aux temps t_i et t_j. La distance estimée séparant i et j est $\hat{d}_{ij}$, la distance restante pour ramener j (respectivement i) au temps contemporain t_0 est $\omega \times (t_0 - t_j) = \omega T_j$ (respectivement ωT_i). Donc la distance δ_{ij} qui voit i et j comme contemporains vaut $\hat{d}_{ij} + \omega \times (T_i + T_j)$.



Remarque 1. La mesure δ_{ij} est une fonction affine de ω ayant pour coefficient directeur $T_i + T_j$ et pour ordonnée à l'origine $\hat{d}_{ij}$. Comme $T_i \geq 0$ et $T_j \geq 0$, il en va de même pour leur somme $T_i + T_j \geq 0$. Ainsi, δ_{ij} est strictement croissante lorsque $T_i + T_j \neq 0$, c'est-à-dire lorsqu'au moins une des deux souches est échantillonnée à un temps différent de t_0 et est constante lorsque $T_i + T_j = 0$, c'est-à-dire lorsque les deux souches sont échantillonnées au temps t_0.

L'hypothèse de l'horloge moléculaire stricte stipule que l'évolution est un processus constant (à travers le temps) et uniforme (à travers les lignées). Donc, les souches identifiées comme contemporaines dans la vraie phylogénie sont à égale distance de leur ancêtre commun. Cette phylogénie peut se représenter par un dendrogramme et la distance additive associée est dite sphérique ou encore ultramétrique. Nous allons préciser cette notion, qui va nous servir à établir le critère sur lequel est basé ULS.

Définition 1. *Soit E un ensemble. Une distance $d: E \times E \to \mathbb{R}_+$ est ultramétrique si pour tout triplet i, j et k de E, la condition*

$$d_{ij} \leq max\{d_{ik}, d_{jk}\}$$

est vérifiée. Cette condition se nomme aussi la condition des trois points.

Soient i, j et k de $\mathcal{E}$, la définition de l'ultramétricité implique que deux des trois nombres d_{ij}, d_{ik} et d_{jk} sont égaux et maximaux. Il est évident que cette condition implique la condition des quatre points (pour tout x, y, z et t de $\mathcal{E}$: $d_{xy} + d_{zt} \leq max\{d_{xz} + d_{yt}, d_{xt} + d_{yz}\}$) et, par conséquent,

l'inégalité triangulaire (pour tout x, y et z de $\mathcal{E}$: $d_{xy} \leq d_{xz} + d_{zy}$). En notant respectivement par $d_{ijk}^{(1)}$ et $d_{ijk}^{(2)}$ le plus grand nombre et le deuxième plus grand nombre parmi d_{ij}, d_{ik} et d_{jk}, nous avons

$$d_{ijk}^{(1)} - d_{ijk}^{(2)} = 0 = d_{ijk}^{(2)} - d_{ijk}^{(1)}.$$

Proposition 1. *Soient un espace E et d une distance sur E. La distance d est ultramétrique si, et seulement si, le critère*

(2) $$Q(d) = \sum_{i<j<k} \left(d_{ijk}^{(1)} - d_{ijk}^{(2)}\right)^2$$

est nul.

Le critère Q utilise la méthode des moindres carrés pour tester l'ultramétricité d'une distance. L'équation (2) montre le critère sous sa forme la plus simple (OLS). Sachant que δ représente les distances entre les feuilles vues comme contemporaines et que nous faisons l'hypothèse d'un taux de substitution ω unique, alors une méthode naturelle d'estimation du taux de substitution ω consiste à minimiser le critère $Q\big(\delta(\omega)\big)$, ce qui revient à rendre δ le plus « ultramétrique possible ».

La formule (2) sous-entend que les estimations $\hat{d}_{ij}$ ont le même taux d'erreur quels que soient i et j. Une supposition fausse car plus la distance entre i et j est grande, plus l'erreur sur $\hat{d}_{ij}$ est importante. Pour prendre en compte cet effet Fitch et Margoliash (1967), Felsenstein (1997) et d'autres proposent d'ajouter une valeur de pondération à chaque terme de la somme, en suivant une approche de type WLS. Cette valeur de pondération augmente l'importance donnée aux estimations ayant un faible taux d'erreur, c'est-à-dire associées à une petite variance, et une faible importance aux estimations ayant un fort taux d'erreur, c'est-à-dire associées à une grande variance. Par ce procédé, la formule (2) devient :

(3) $$Q\big(\delta(\omega)\big) = \sum_{i<j<k} w_{ijk} \left(\delta_{ijk}^{(1)}(\omega) - \delta_{ijk}^{(2)}(\omega)\right)^2$$

où w_{ijk} est la valeur de pondération, dépendante de i, j et k, attribuée à chaque terme de la somme. Le choix de cette valeur de pondération, correspondant à l'inverse d'une variance, sera discuté à la section 4.2.3.

4.2.1 Minimisation du critère d'ultramétricité sur un triplet

Dans un premier temps, nous allons étudier le comportement du critère en ne considérant qu'un seul triplet, puis nous le généraliserons sur plusieurs triplets. De plus, avec un seul triplet la pondéra-

tion WLS n'intervient pas, simplifiant alors l'analyse. Notons par $\delta_{|ijk}(\omega)$ la distance corrigée $\delta(\omega)$ restreinte au seul triplet i, j et k de $\mathcal{E}$.

Sur un triplet, l'équation (2) se réduit à un terme

$$Q\left(\delta_{|ijk}(\omega)\right) = \left(\delta_{ijk}^{(1)}(\omega) - \delta_{ijk}^{(2)}(\omega)\right)^2. \tag{4}$$

Ce terme est dépendant des trois droites $\delta_{ij}(\omega)$, $\delta_{ik}(\omega)$ et $\delta_{jk}(\omega)$. Leurs équations sont définies en (1). Dans la suite et pour simplifier, on omettra d'indiquer ω lorsque ce ne sera pas nécessaire.

Remarque 3. Soit un intervalle $I = [a, b]$ où les droites δ_{ij}, δ_{ik} et δ_{jk} n'ont aucun point d'intersection. Alors les deux plus hautes droites correspondent aux termes $\delta_{ijk}^{(1)}(\omega)$ et $\delta_{ijk}^{(2)}(\omega)$, qui restent identiques sur I. Posons $\delta_{ijk}^{(1)}(\omega) = a_1\omega + b_1$ et $\delta_{ijk}^{(2)}(\omega) = a_2\omega + b_2$. Alors

$$Q\left(\delta_{|ijk}(\omega)\right) = \left((a_1 - a_2)\omega + b_1 - b_2\right)^2 = A^2\omega^2 + 2AB\omega + B^2,$$

avec $A = a_1 - a_2$ et $B = b_1 - b_2$. Ainsi, $Q\left(\delta_{|ijk}(\omega)\right)$ est une parabole de variable ω. Elle est positive (c'est une différence au carré) et convexe (le coefficient de son monôme de plus haut degré est positif). De plus, lorsque les droites $\delta_{ijk}^{(1)}$ et $\delta_{ijk}^{(2)}$ sont parallèles ($a_1 = a_2$), le critère est constant sur I ($A = 0$).

Regardons maintenant ce qu'il se passe autour des points d'intersection des trois droites $\delta_{ij}(\omega)$, $\delta_{ik}(\omega)$ et $\delta_{jk}(\omega)$, c'est-à-dire là où les termes $\delta_{ijk}^{(1)}$ et $\delta_{ijk}^{(2)}$ sont modifiés, correspondant alors à des droites différentes. Il convient de distinguer deux types de points d'intersection. Le premier regroupe les points d'intersection qui ne modifient pas l'expression du critère. Ce sont les points d'intersection entre les deux plus hautes droites ; ils permutent les droites représentées par les termes $\delta_{ijk}^{(1)}$ et $\delta_{ijk}^{(2)}$ entre elles, mais leur différence au carré reste identique avant et après le point d'intersection qui a la particularité d'annuler le critère. Le second regroupe les points d'intersection qui modifient l'expression du critère. Ce sont les points d'intersection entre la deuxième et la troisième plus haute droite. Ils modifient uniquement la droite représentée par le terme $\delta_{ijk}^{(2)}$, et, donc, le comportement du critère avant et après ce point change, au sens où on a toujours une parabole, mais d'équation différente.

Définition 2. *Soient trois droites quelconques A, B et C définies sur $\mathbb{R}$, et soit x_{ab} le point d'intersection entre les droites A et B. Le point x_{ab} est dit « frontière » si l'inégalité*

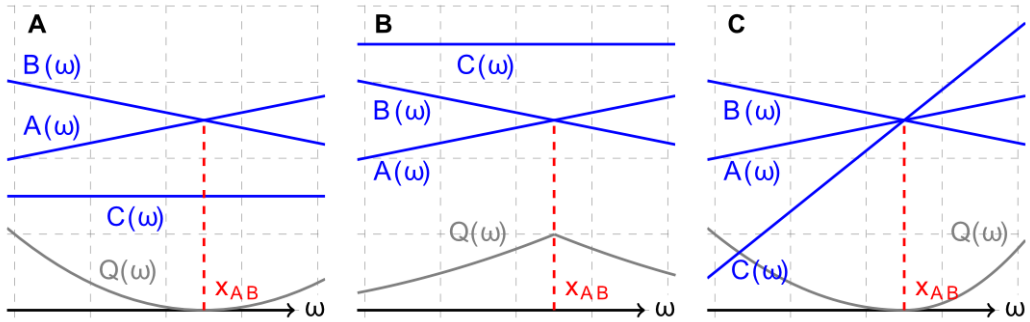
$$A(x_{AB}) = B(x_{AB}) \leq C(x_{AB})$$

est vérifiée. Il est dit « solution » si l'inégalité

$$A(x_{AB}) = B(x_{AB}) \geq C(x_{AB})$$

*est vérifiée (*Figure 27*).*

Figure 27. Différence entre point solution et point frontière.
Le graphique A montre un point solution (une seule parabole dont on atteint le minimum), le graphique B un point frontière (on change de parabole) et le graphique C montre qu'il est possible pour un point d'être à la fois frontière et solution (deux paraboles se rencontrent sur leur minimum).

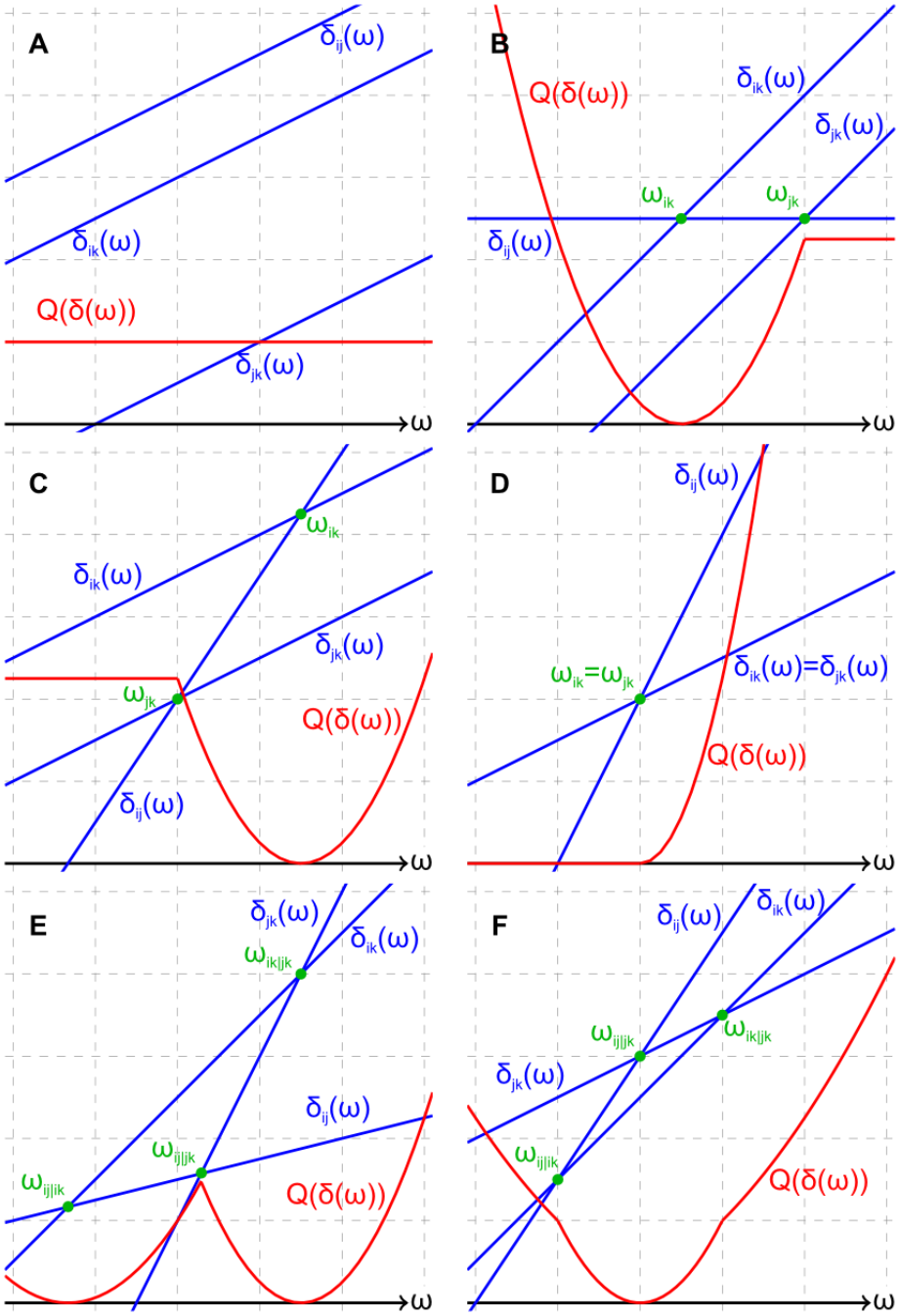


Les points d'intersection solutions entre les droites δ_{ij}, δ_{ik}et δ_{jk} sont ceux qui annulent le critère $Q(\delta_{|ijk})$. En effet, en ces points, les deux termes $\delta_{ijk}^{(1)}$ et $\delta_{ijk}^{(2)}$ sont égaux et leur différence est donc nulle. Ils correspondent à une solution optimale pour l'estimation du taux de substitution avec le critère $Q(\delta_{|ijk})$. Les autres points d'intersection, les points frontières, n'annulent pas le critère, mais changent la fonction parabolique représentant sa valeur. Ainsi, le critère est une fonction parabolique par morceaux. La définition 2 n'exclut pas le fait qu'un point d'intersection peut être à la fois frontière et solution (Figure 27C). Ce dernier cas se produit lorsque les trois droites sont concourantes. La Figure 28 montre le comportement du critère sur un triplet, ainsi que les droites δ_{ij}, δ_{ik} et δ_{jk} ayant permis d'obtenir son allure.

Remarque 4. *L'expression $Q(\delta_{|ijk})$, définie en* (4)*, est continue sur* $\mathbb{R}$.

Les différents cas de figures montrés à la Figure 28 suggèrent qu'il y a toujours au moins une solution pour le taux de substitution ω, qui rende $\delta_{|ijk}$ ultramétrique, sauf lorsque les droites δ_{ij}, δ_{ik} et δ_{jk} sont parallèles. Cependant, les minimums de la fonction ne sont pas toujours acceptables (Figure 29). En effet, comme le taux de substitution est une vitesse, les valeurs négatives ne peuvent lui être assignées ; dans ce cas, la solution optimale est parfois zéro, comme montré dans la Figure 29.

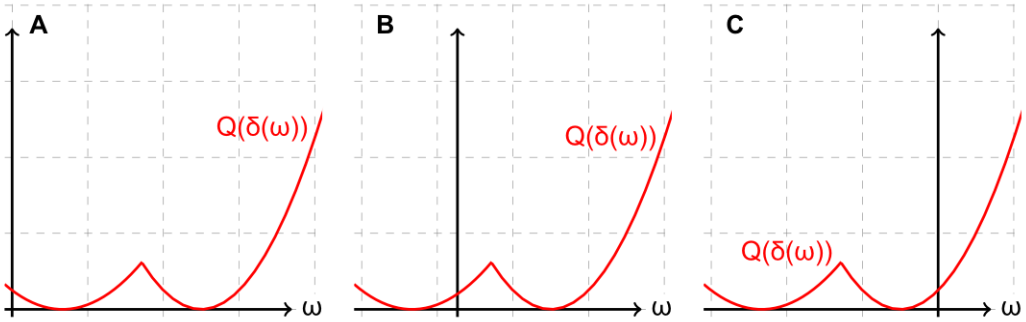
Figure 28. Quelques exemples (non exhaustifs) de l'allure du critère restreint à un triplet.
Le comportement du critère restreint à un triplet sur $\mathbb{R}$ est montré en rouge, les trois droites δ_{ij}, δ_{ik} et δ_{jk} en bleu et les points d'intersection en vert. Pour l'exemple A, les trois droites sont parallèles et le critère résultant est une droite. Les exemples B et C montrent les cas où seulement deux droites sont parallèles et l'exemple D le cas où deux droites sont confondues. Les exemples E et F se produisent dans le cas où aucune des droites n'est parallèle. L'exemple E a deux solutions tandis que l'exemple F a deux frontières.



Dans le cas d'un seul triplet, il est généralement possible (sauf Figure 29C) de trouver au moins une valeur pour le taux de substitution rendant la distance $\delta_{|ijk}(\omega)$ ultramétrique. Cette valeur est déterminée par le(s) point(s) d'intersection solution(s), qui coïncide(nt) avec un minimum du critère. La valeur de ces points peut aussi être obtenue en annulant la dérivée du critère. Lorsque nous considérerons plusieurs triplets (cf. ci-dessous), les points d'intersection solutions des différents triplets

ne correspondront plus avec un minimum du critère (sauf cas très particulier), et pour en déterminer le minimum global, il faudra dériver chaque morceau du critère, lui aussi parabolique par morceaux. Par ailleurs, dans le cas d'un seul triplet, le minimum rend toujours la distance corrigée parfaitement ultramétrique. Mais cette observation n'est plus vraie lorsque plusieurs triplets sont considérés et le résultat rendra la distance corrigée « aussi ultramétrique que possible ».

Figure 29. Solutions considérées par l'algorithme ULS suivant la positivité ou la négativité des points solutions.
Le critère restreint à un triplet a toujours au plus deux solutions qui l'annulent. Cependant, ces solutions ne sont pas toujours convenables suivant qu'elles soient positives ou négatives (graphiques A et B). Dans le cas du graphique C, aucune des solutions proposées par le triplet n'est considérée, et zéro sera présenté comme solution optimale.



4.2.2 Minimisation du critère d'ultramétricité sur plusieurs triplets

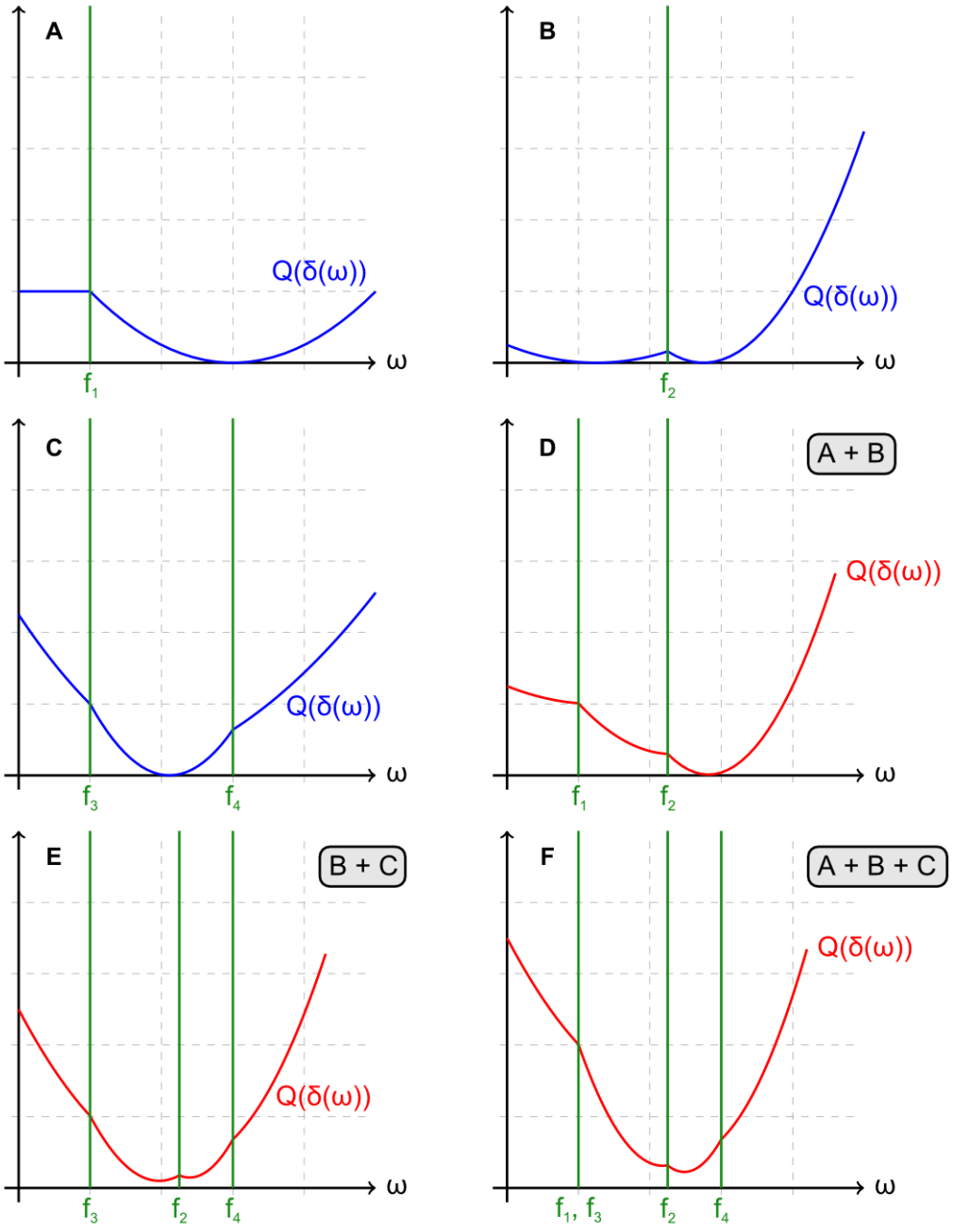
Le critère d'ultramétricité restreint à un triplet se comporte sur $\mathbb{R}$ comme une fonction parabolique par morceaux, où chaque morceau est soit une constante ou soit une parabole convexe positive. Quand il est étudié sur plusieurs triplets, il devient une somme de fonctions paraboliques par morceaux (Figure 30), c'est donc toujours une fonction parabolique par morceaux, mais elle est plus complexe. Néanmoins, l'objectif reste d'en déterminer le minimum, considéré comme la meilleure estimation possible pour le taux de substitution ω, et qui rend la distance corrigée δ aussi ultramétrique que possible.

L'étude du critère sur un triplet a montré que le minimum du critère correspond au(x) point(s) solution(s)**.** Les exemples E et F de la Figure 30 montrent que cela n'est plus systématiquement vrai. Donc, le minimum du critère ne peut être obtenu que par dérivation des morceaux de parabole du critère. Il convient ensuite de vérifier si ce minimum est inclus dans le domaine de définition du morceau considéré. Il devient alors une solution potentielle, et le minimum global correspond alors au minimum de toutes les solutions potentielles (le plus souvent uniques).

Le critère est continu en tout point de $\mathbb{R}$, puisqu'il est la somme de critères continus définis sur chaque triplet, mais il n'en est pas dérivable aux points frontières. Cependant, en ces points la propriété de dérivabilité n'est pas nécessaire. En effet, sauf dans un cas particulier (Figures 28A et 28D), extrêmement rare avec des données réelles et correspondant à un plateau, on peut montrer que les

morceaux de parabole ont une pente croissante après le passage de chaque point frontière (Figures 30D, 30E et 30F). De ce fait, il est inutile de rechercher une solution aux points frontières puisqu'elle se trouvera forcément entre ces points (à moins d'un plateau, ou lorsque le changement de paraboles correspond aussi à leur minimum, cf. Figure 27C, ces deux cas nécessitant un soin particulier, bien que très peu probables).

Figure 30. Comportement du critère sur plusieurs triplets.
Les graphiques A, B et C montrent le critère obtenu à partir d'un triplet, tandis que les graphiques D, E et F le montrent comme étant la somme de plusieurs critères définis sur un triplet. Le critère du graphique D (respectivement E) correspond à la somme des critères A et B (respectivement B et C). Le critère du graphique F correspond à la somme des trois critères A, B et C. Les graphiques E et F montrent qu'il n'est pas possible d'obtenir une valeur qui annule le critère, mais un minimum existe et il est alors choisi comme solution optimale. Le graphique F montre qu'il est possible d'avoir plusieurs points frontières identiques et E qu'il est possible d'avoir plusieurs minima locaux.



Remarque 5. Soit un intervalle $I \subset \mathbb{R}$ sur lequel il n'y a aucun point frontière. Alors les variables $\delta_{ijk}^{(1)}$ et $\delta_{ijk}^{(2)}$, de chaque terme de la somme (2), représentent toujours les mêmes droites. Pour un triplet i, j et k de $\mathcal{E}$, posons $\delta_{ijk}^{(1)}(\omega) = a_{ijk}^{(1)}\omega + b_{ijk}^{(1)}$ et $\delta_{ijk}^{(2)}(\omega) = a_{ijk}^{(2)}\omega + b_{ijk}^{(2)}$, alors

$$Q\big(\delta(\omega)\big) = \sum_{i<j<k} \left(\delta_{ijk}^{(1)}(\omega) - \delta_{ijk}^{(2)}(\omega)\right)^2 = \sum_{i<j<k} \left(\left(a_{ijk}^{(1)} - a_{ijk}^{(2)}\right)\omega + b_{ijk}^{(1)} - b_{ijk}^{(2)}\right)^2$$

d'où

$$Q\big(\delta(\omega)\big) = \omega^2 \sum_{i<j<k} \left(a_{ijk}^{(1)} - a_{ijk}^{(2)}\right)^2 + 2\omega \sum_{i<j<k} \left(a_{ijk}^{(1)} - a_{ijk}^{(2)}\right)\left(b_{ijk}^{(1)} - b_{ijk}^{(2)}\right) + \sum_{i<j<k} \left(b_{ijk}^{(1)} - b_{ijk}^{(2)}\right)^2.$$

Le minimum ω^* de cette parabole est déductible par dérivation et vaut

$$\omega^* = -\frac{\sum_{i<j<k}\left(a_{ijk}^{(1)} - a_{ijk}^{(2)}\right)\left(b_{ijk}^{(1)} - b_{ijk}^{(2)}\right)}{\sum_{i<j<k}\left(a_{ijk}^{(1)} - a_{ijk}^{(2)}\right)^2}.$$

Si $\omega^* \in I$, alors ω^* est un minimum local au critère $Q(\delta)$. Dans le cas contraire, $Q(\delta)$ n'a aucun minimum sur I, du fait de la propriété précédente.

Pour obtenir le minimum global du critère, il faut donc rechercher le minimum local de chaque morceau. La remarque 5 fournit, pour un morceau de parabole, la formule donnant son minimum local. L'algorithme va donc consister à parcourir les points frontières et à vérifier sur chaque intervalle si celui-ci contient ou non une solution potentielle.

Comme le taux de substitution est nécessairement positif ou nul, seule la définition du critère sur $\mathbb{R}_+$ a besoin d'être considérée. Mais il se peut que certains triplets aient leurs points frontières et solutions tous négatifs (Figure 29C). Ce type de triplets biaise l'estimation du taux de substitution et « pousse » celle-ci vers zéro. Pour éviter un tel biais, les triplets ayant cette caractéristique sont omis de l'analyse.

Le comportement du critère reste identique avec la version pondérée du critère. En effet, multiplier chaque terme de la somme par une constante positive ne modifie en rien ses propriétés. Cependant, cette constante influe sur l'importance à donner à tels ou tels termes. En effet, plus cette constante est proche de zéro, plus le terme correspondant, jugé non informatif, tend à ressembler à une droite horizontale passant par zéro (et a donc peu d'influence sur le critère), tandis que plus sa valeur est élevée, plus elle donne de l'impact à ce terme (jugé informatif). Ainsi, considérer des pondérations sur chaque terme de la somme n'a pas de conséquence sur les propriétés énoncées précédemment, mais uniquement sur le résultat final.

4.2.3 Détermination de la valeur de pondération optimale

Pour les méthodes standards de reconstruction phylogénétique, la valeur de pondération associée à chaque terme de la somme des moindres carrés pondérés (WLS) est $1/\hat{d}_{ij}$ (Swofford *et al*, 1996), car $\hat{d}_{ij}$ est une bonne approximation (à un facteur constant dépendant du nombre de sites) de la variance associée à $\hat{d}_{ij}$ (Felsenstein, 1984), signifiant que la confiance dans l'estimation des distances évolutives devient de plus en plus faible au fur et à mesure que $\hat{d}_{ij}$ est grand. Une variante de pondération largement utilisée, proposée par Fitch et Margoliash (1967), est $1/\hat{d}_{ij}^2$. Sur un jeu de données comprenant 43 mammifères, Sanjuán et Wróbel (2005) montrent que 1,823 est une valeur d'exposant optimal, plutôt que 2. Sur un autre jeu de données contenant des souches *env* du VIH-1 provenant de différents organes (moelle osseuse, cerveau, liquide cérébro-spinal, rein, foie, poumon, ganglion lymphatique et rate), ils montrent, cette fois-ci, que la valeur d'exposant 1,766 est la plus appropriée. Ce dernier jeu de données avait pour but de montrer le phénomène de compartimentalisation du VIH (McGrath *et al*, 2001). Dans tous les cas, ils indiquent que $1/\hat{d}_{ij}^2$ fournit des résultats similaires. Cette valeur d'exposant (2) semble être *a priori* optimale (ou n'en est pas loin) et nous la conserverons dans les simulations ci-après.

Les formules de pondération précédentes ne peuvent pas être utilisées directement pour w_{ijk} puisque chaque terme de la somme du critère dépend sur $\mathbb{R}$ des trois distances $\hat{d}_{ij}$, $\hat{d}_{ik}$ et $\hat{d}_{jk}$ (sauf cas particulier). De plus, nous avons montré que le critère est une fonction continue et, afin de conserver cette propriété remarquable, il est important que la valeur de pondération associée à chaque terme de la somme soit constante sur $\mathbb{R}$. Une mesure naturelle est alors d'utiliser l'inverse de la somme des trois distances, soit

$$w_{ijk} = \frac{1}{\hat{d}_{ij} + \hat{d}_{jk} + \hat{d}_{ik}}.$$

Cette valeur de pondération a le même comportement que celle utilisée classiquement. Néanmoins, si la somme des trois distances est très proche de zéro, une confiance presque absolue est donnée au terme correspondant et faussera inévitablement les estimations du taux de substitution. Il est d'usage de supposer que chaque mesure $\hat{d}_{ij}$ ne peut pas être plus petite que la moitié d'une substitution observée, c'est-à-dire $\hat{d}_{ij} \geq 1/(2N)$, où N est la longueur de l'alignement (Swofford *et al*, 1996). Ainsi, nous pouvons rajouter un pseudo-compte au dénominateur afin d'éviter les valeurs trop petites ou nulles, d'où

$$w_{ijk} = \frac{1}{\hat{d}_{ij} + \hat{d}_{ik} + \hat{d}_{jk} + k/N},$$

où k est choisi sur la base de simulations de manière à optimiser l'impact de w_{ijk} sur chaque terme de la somme. En suivant l'approche classique de Fitch et Margoliash (1967), on peut employer le carré de cette pondération, soit

(5) $$w_{ijk} = \frac{1}{\left(\hat{d}_{ij} + \hat{d}_{ik} + \hat{d}_{jk} + k/N\right)^2}.$$

Le produit des trois distances (au lieu de leur somme) peut aussi être considéré comme valeur de pondération, bien que l'interprétation en soit moins naturelle, soit

$$w_{ijk} = \frac{1}{\hat{d}_{ij} \times \hat{d}_{ik} \times \hat{d}_{jk} + k/N},$$

et à nouveau on peut passer au carré

(6) $$w_{ijk} = \frac{1}{\left(\hat{d}_{ij} \times \hat{d}_{ik} \times \hat{d}_{jk} + k/N\right)^2}.$$

Dessimoz *et al*. (2006) ont proposé des formules moins empiriques, bien adaptées à notre cas de figure. Ils ont calculé la variance de la différence $\left(\hat{d}_{ik} - \hat{d}_{jk}\right)$, et notre critère est justement calculé à partir de différences de distances, qui est approximée par la formule

(7) $$\sigma^2\left(\hat{d}_{ik} - \hat{d}_{jk}\right) = \frac{\hat{d}_{ij}^{1,3182}}{v_{ij}^{0,3026}} \times \frac{\left(v_{ik} + v_{jk}\right)^{1,0933}\left(v_{ik}v_{jk}\right)^{0,1181}}{\left(\hat{d}_{ik} + \hat{d}_{jk}\right)^{1,2449}},$$

où les termes v_{ij}, v_{ik} et v_{jk} sont les variances respectivement associées aux distances $\hat{d}_{ij}$, $\hat{d}_{ik}$ et $\hat{d}_{jk}$. Ces variances peuvent être calculées pour la plupart des modèles d'évolution, mais elles sont rarement données par les logiciels les plus couramment utilisés. Elles peuvent alors être approximées par $\hat{d}_{ij}/N$, $\hat{d}_{ik}/N$ et $\hat{d}_{jk}/N$ respectivement, où N est la longueur de l'alignement. Le problème avec cette variance est qu'elle considère une paire de distances, issues de $\delta_{ijk}^{(1)}$ et $\delta_{ijk}^{(2)}$, et que cette paire peut varier aux points frontières. Afin de conserver une variance constante sur $\mathbb{R}$ (cf. ci-dessus), nous considérons, pour chaque terme de la somme, la variance associée aux droites avec un point solution, et si plusieurs variances sont possibles, la plus grande est choisie. Ainsi, la valeur de pondération correspond à l'inverse de cette variance maximum plus un pseudo-compte de la forme k/N^2.

La pertinence des différentes valeurs de pondérations proposées est testée sur un jeu de données contenant 800 simulations (Figure 31) et pour lequel les séquences sont de longueur 300 paires de bases pour la figure A et de 1 000 paires de bases pour la figure B. La description complète du proto-

cole de simulation est donnée à la section 4.3.1.1. Les valeurs de pondération testées sont : sans pondération en rouge, pondération tenant compte de la variance de Dessimoz (7) pour k valant 0 et 1 en jaune et vert respectivement, pondération (5) pour k valant 0 et 1 respectivement en cyan et bleu et pondération (6) pour k valant 1 en violet. L'ordonnée indique la précision d'estimation des différentes valeurs de pondération (fonction déviation relative, cf. section 4.3.1.2). Pour l'essentiel, plus cette valeur est petite, plus les estimations sont précises. Différentes entrées sont aussi étudiées, soit avec des matrices de distances (calculées avec DNAdist), soit avec des arbres FastME (calculés à partir des matrices de distances) ou soit avec des arbres PhyML. Dans ces deux derniers cas, on commence par calculer la distance patristique avant d'appliquer ULS. Les précisions d'estimation des différentes valeurs de pondération proposées sont quasi-identiques pour des valeurs de k oscillant entre 0,5 et 2, et ne sont donc pas toutes montrées. De même, les précisions d'estimation de la valeur de pondération (6) pour $k = 0$ ne sont pas montrées car elles sont systématiquement supérieures à 0,2 pour des séquences de 300 paires de bases et supérieures à 0,1 pour des séquences de 1 000 paires de bases. Les différences en précision d'estimation sont flagrantes en passant d'alignements de 300 sites à 1 000 sites, comme on peut s'y attendre. Dans ce dernier cas, il n'y a presque plus de différence en précision d'estimation entre les différentes valeurs de pondération puisqu'au fur et à mesure que la longueur des séquences augmente les estimations des distances évolutives deviennent de plus en plus justes et la pondération devient alors plus ou moins superflue, en particulier avec des arbres FastME et PhyML. Les alignements de 300 sites semblent dont être le cas de figure idéal pour trancher entre les différentes valeurs de pondération où l'on observe, qu'en moyenne, la pondération (6) pour $k = 1$ est la plus performante, en particulier avec des matrices de distances. Donc, la valeur de pondération (6) pour $k = 1$ est conservée pour notre méthode. Remarquons que la valeur de pondération (5) et celle avec la variance de Dessimoz (7), pour $k = 0$, semblent inadaptées avec des arbres PhyML et des séquences de 300 paires de bases, mais ce problème est corrigé en utilisant un pseudo-compte ou en considérant un alignement plus grand.

4.2.4 Limites algorithmiques et solutions proposées

La mise en œuvre d'ULS nécessite l'utilisation de techniques algorithmiques avancées pour résoudre certains problèmes pratiques (essentiellement place mémoire) et d'observer les propriétés du critère (par exemple, la possibilité d'échantillonner les triplets) afin d'obtenir un programme rapide et précis.

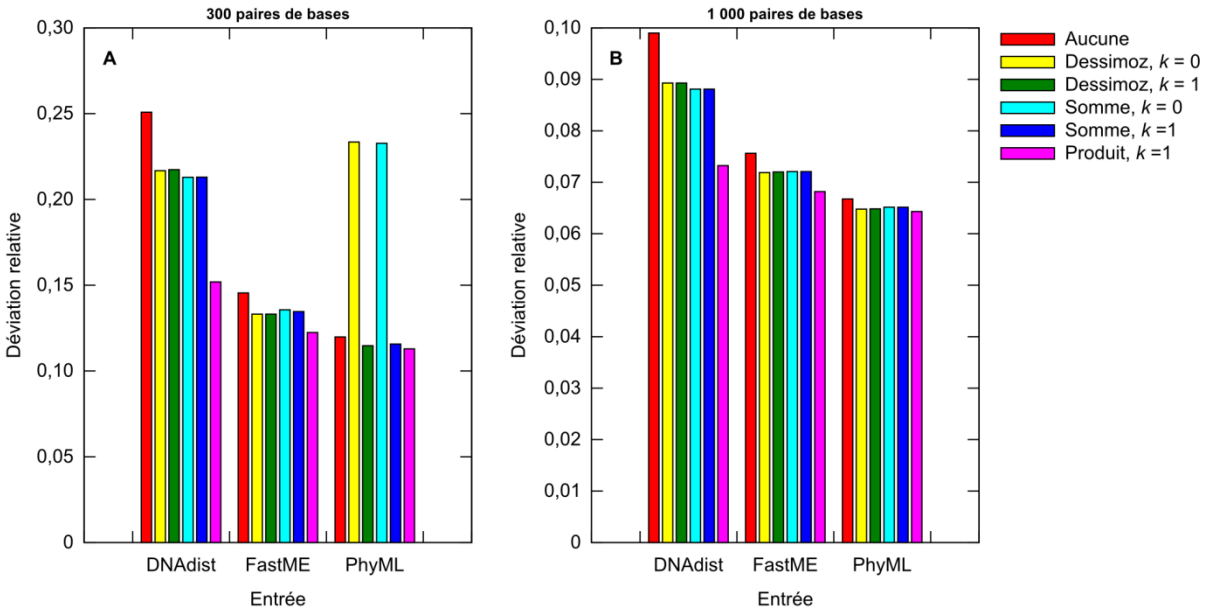
4.2.4.1 Conservation des coefficients de chaque morceau de parabole

ULS minimise un critère qui est une fonction parabolique par morceaux pour déterminer la meilleure estimation du taux de substitution. Chaque morceau du critère est un polynôme du second

degré (ou une constante qui en est un cas particulier) et peut donc être représenté par les coefficients de ses trois monômes (dans le cas d'une constante, les coefficients des monômes de degré un et deux sont nuls). Pour connaître le minimum global du critère, il faut donc connaître avec exactitude la définition numérique de chaque morceau de parabole, ainsi que leur intervalle de définition.

Initialement, les frontières sont indéterminées. Elles apparaissent progressivement et de façon aléatoire lors du parcours des triplets. Ces contraintes obligent à devoir construire les coefficients associés à chaque morceau de façon progressive mais surtout non dépendante de l'intervalle auquel il appartient (puisque cet intervalle est uniquement connu à la fin du parcours des triplets). Autrement, il serait nécessaire de parcourir l'ensemble des triplets deux fois de suite : une première fois pour connaître toutes les frontières et la seconde pour connaitre les coefficients de chaque morceau de parabole.

Figure 31. Performance en précision d'estimation des différentes valeurs de pondération étudiées.
Ces graphiques montrent en ordonnée les performances en précision d'estimation (plus les valeurs sont petites, meilleures sont les estimations ; fonction déviation relative, cf. section 4.3.1.2) pour les différentes valeurs de pondération proposées (de gauche à droite : aucune pondération en rouge, pondération avec variance de Dessimoz (7) pour $k = \{0; 1\}$ en jaune et vert respectivement, pondération (5) pour $k = \{0; 1\}$ en cyan et bleu respectivement, pondération (6) pour $k = 1$ en violet) et pour différentes entrées (matrices de distances (DNAdist), arbres FastME et PhyML). Les figures A et B correspondent aux mêmes jeux de données contenant chacun 800 simulations (cf. section 4.3.1.1), mais avec des séquences de 300 paires de bases pour la figure A et de 1 000 paires de bases pour la figure B. Les précisions d'estimation de la pondération (6) pour $k = 0$ sont toutes supérieures à 0,2 avec des séquences de 300 paires de bases et à 0,1 avec des séquences de 1 000 paires de bases. Enfin, aucune différence notable n'existe pour différentes valeurs de k oscillant entre 0,5 et 2.



Une méthode simple et efficace qui permet de construire les coefficients de chaque morceau de parabole, en s'abstenant de la connaissance de leur intervalle de validité, est d'utiliser des coefficients temporaires associés à chaque frontière. Notons par (a_u, b_u, c_u) les coefficients du morceau de parabole $m_u(\omega) = a_u\omega^2 + b_u\omega + c_u$ défini entre les frontières f_u et f_{u+1} ($f_u < f_{u+1}$). Chaque frontière f_u correspond à la borne inférieure de l'intervalle de définition du morceau m_u et à la

borne supérieure de l'intervalle de définition du morceau m_{u-1}. Elle contient les coefficients temporaires (a'_u, b'_u, c'_u) qui permettent de connaître les coefficients (a_u, b_u, c_u) du morceau de parabole m_u en les sommant aux coefficients $(a_{u-1}, b_{u-1}, c_{u-1})$ du morceau de parabole m_{u-1}, c'est-à-dire

$$(a_u, b_u, c_u) = (a_{u-1}, b_{u-1}, c_{u-1}) + (a'_u, b'_u, c'_u). \quad (8)$$

Concevoir de tels coefficients temporaires est chose aisée. En effet, les frontières associées à un triplet, ainsi que les coefficients des morceaux de parabole entre ces frontières, sont facilement calculables et constituent la base pour concevoir les coefficients temporaires. Imaginons que pour un triplet i, j et k de $\mathcal{E}$ nous avons deux frontières positives f^1_{ijk} et f^2_{ijk}, $f_0 < f^1_{ijk} < f^2_{ijk}$, avec $f_0 = 0$. Ajoutons aux coefficients temporaires de f_0 les coefficients du morceau propre à i, j et k passant par f_0 (ici défini sur $]-\infty; f^1_{ijk}]$) et retranchons-les aux coefficients temporaires de f^1_{ijk}. Ensuite, ajoutons aux coefficients temporaires de f^1_{ijk} les coefficients du morceau propre à i, j et k défini entre $[f^1_{ijk}; f^2_{ijk}]$, puis après avoir retranché ces derniers aux coefficients temporaires de f^2_{ijk}, ajoutons-leur les coefficients du morceau de parabole défini entre $[f^2_{ijk}; +\infty[$. En procédant de même pour tous les triplets, les coefficients associés à chaque frontière ont la propriété de l'équation (8) et sont calculés itérativement en parcourant l'ensemble des triplets.

Comme le taux de substitution est supposé être positif ou nul (hypothèse biologique évidente), la définition du critère sur $\mathbb{R}_-$ est inutile. Aussi, les frontières négatives ne sont pas considérées. Cependant, chaque triplet a une influence non négligeable sur la partie positive ($\mathbb{R}_+$) du critère, même si toutes ses frontières sont négatives (sauf, bien sûr, dans le cas d'une constante ; l'influence est alors identique en tout point de $\mathbb{R}$ et peut être négligée). Dans ce dernier cas, le triplet est uniquement considéré s'il a au moins une solution positive, et cela même si toutes ses frontières sont négatives. Les triplets n'ayant pas de solution(s) positive(s) ne sont jamais considérés (cf. section 4.2.1). Ainsi, $f_0 = 0$ est considéré comme la première frontière de chaque triplet. Elle contient donc l'information de tous les morceaux de parabole considérés. De ce fait, les coefficients temporaires (a'_0, b'_0, c'_0) associés à cette frontière correspondent aux vrais coefficients (a_0, b_0, c_0) associés au premier morceau de parabole m_0. Donc, les coefficients du morceau m_u, défini sur l'intervalle $[f_u; f_{u+1}]$, s'expriment par la relation

$$(a_u, b_u, c_u) = \sum_{k=0}^{u} (a'_k, b'_k, c'_k),$$

où (a'_k, b'_k, c'_k) représentent les coefficients temporaires associés à la frontière f_k, $k = \{0, \cdots, u\}$.

Il convient de noter que, lors d'un cas particulier où le critère associé à un triplet a deux solutions, une positive et l'autre négative (Figure 29B), le morceau de parabole défini sur $\mathbb{R}_+$ qui correspond à la solution négative (s'il y en a un) n'est pas considéré (puisqu'il biaisera l'estimation du taux de substitution vers zéro) et que, dans ce cas, le morceau de parabole associé à la solution positive débutera à zéro et non au point frontière.

4.2.4.2 Parcours de chaque morceau du critère et estimation des minima locaux

Pour calculer le minimum global du critère, nous devons estimer le minimum de chacun de ses morceaux et vérifier s'il correspond à un minimum à considérer, c'est-à-dire si le minimum du morceau est compris dans l'intervalle de définition de celui-ci. Dans la section précédente, nous décrivons la manière de calculer les coefficients temporaires stockés dans chaque frontière. Imaginons que nous disposons d'un tableau contenant ces frontières de façon unique et par ordre croissant (cela fera l'objet de la section suivante). Pour déterminer le minimum global du critère nous devons parcourir chaque morceau de celui-ci. Comme les frontières négatives ne sont pas considérées, la première dans ce tableau est donc $f_0 = 0$ et c'est celle dont les coefficients temporaires (a'_0, b'_0, c'_0) représentent les vrais coefficients (a_0, b_0, c_0) associés au premier morceau de parabole m_0. L'abscisse du minium de m_0 est donc $\omega_0 = -b_0/(2a_0)$. Si $f_0 \leq \omega_0 \leq f_1$, ce minimum est à considérer comme un minimum local, sinon il est ignoré. Les coefficients du morceau m_1 s'obtiennent en $O(1)$ en ajoutant aux coefficients (a_0, b_0, c_0) les coefficients temporaires associés à la frontière f_1. Puis, on calcule le minimum, vérifie s'il constitue un minimum local, et on passe à m_2. Et ainsi de suite. En parcourant de cette manière le tableau, l'équation numérique de chaque morceau m_u du critère est déductible facilement et son minimum $\omega_u = -b_u/2a_u$ est local si, et seulement si, $f_u \leq \omega_u \leq f_{u+1}$. Chaque frontière nécessite un calcul en $O(1)$, si bien que la complexité en temps (il faut parcourir l'ensemble des triplets) et en espace (il faut stocker les frontières ; cf. ci-après) est en $O(n^3)$.

4.2.4.3 Structure de données associée aux frontières

Pour n souches, il y a au plus C_n^3 combinaisons sans répétitions de trois souches parmi n et comme chaque triplet peut avoir au plus deux frontières (impossible d'obtenir trois points frontières, au moins un des points est solution), il y a au plus

$$2 \times C_n^3 = \frac{n(n-1)(n-2)}{3}$$

frontières à conserver en mémoire. Avec $n = 200$, nous avons 7 880 400 frontières et en considérant le fait que nous devons au moins avoir quatre nombres réels associés à chaque frontière (un pour représenter la frontière et trois pour les coefficients temporaires), nous utilisons au plus environ 640

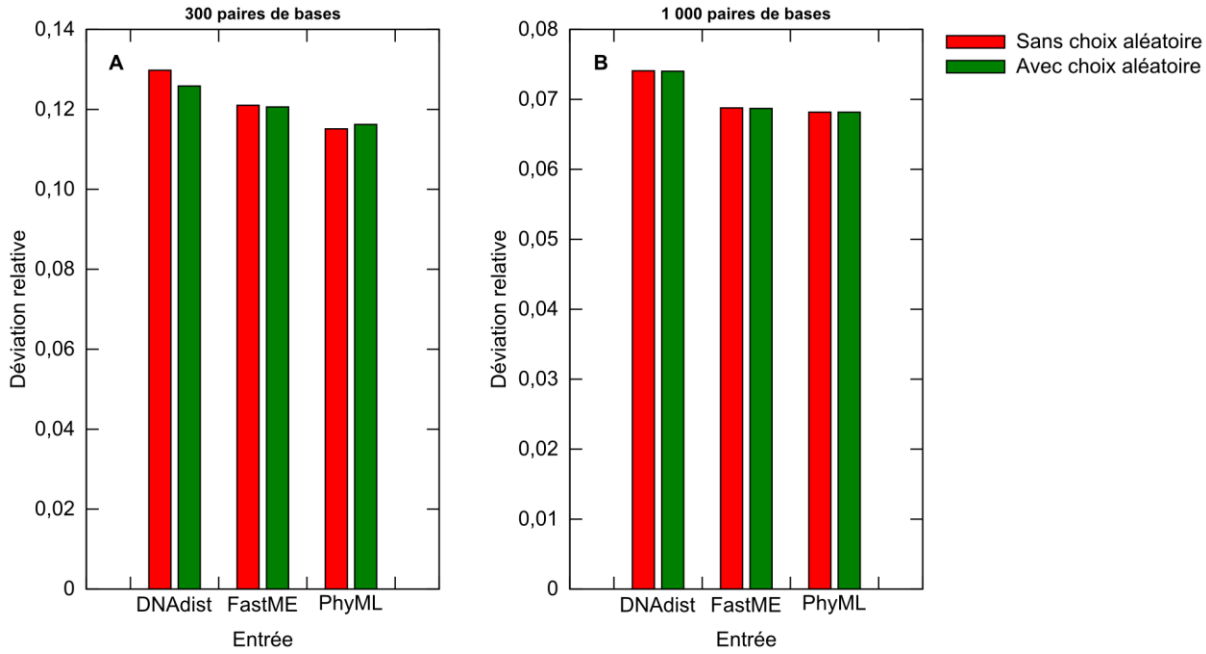
mégabits de mémoire (sachant qu'un nombre réel est codé sur 64 bits) et cela dépasse les gigabits pour $n = 300$. Bien entendu, cette estimation du nombre de frontières est beaucoup plus importante qu'en pratique. En effet, une même frontière peut appartenir à plusieurs triplets et les duplicata ne sont pas conservés. De plus, nous estimons beaucoup plus de combinaisons de triplets qu'il y en a en réalité, car considérer trois souches échantillonnées à une même date est inutile (il ne donne aucune information sur le taux de substitution puisque le critère résultant de ce triplet est constant ; en outre il n'a aucun point frontière), et tous les triplets n'ont pas forcément deux points frontières positifs. Malgré cela, le nombre de frontières reste tout de même important.

Pour stocker les frontières en mémoire, nous devons donc choisir une structure de données dynamique. Cette structure doit permettre de rechercher un élément (pour ne pas avoir deux fois le même) le plus rapidement possible et d'en insérer un nouveau (lorsqu'il n'est pas encore présent dans l'arbre) tout en conservant l'ordre établi sur la structure. La structure de données des arbres AVL (Adelson-Velskii & Landis, 1962) a l'avantage d'être rapide en termes de recherche et d'ajout d'un élément et conserve la propriété d'ordonnancement établie sur les éléments déjà lus (lors de l'ajout). Ce sont des arbres binaires de recherche dont la différence de hauteur entre le sous-arbre droit et le sous-arbre gauche d'un nœud n'excède pas un. L'ajout et la suppression d'un nœud de l'arbre nécessitent éventuellement une étape de rééquilibrage afin de conserver les propriétés spécifiques de cette structure. Le temps de calcul nécessaire pour accomplir ces deux tâches (ajout et recherche d'un élément) est en $O(\log k)$, où k est le nombre de nœuds existant dans l'arbre. Ensuite, un simple parcours infixe permet d'obtenir une liste triée des frontières. La complexité en temps est donc en $O(n^3 \log n)$, où n est le nombre de séquences.

La seule utilisation de cette structure de données n'est pas suffisante à l'obtention d'une bonne vitesse d'exécution, mais surtout elle n'influe pas sur la taille de la mémoire nécessaire au programme. En utilisant un algorithme simple d'épaississement des frontières (c.-à-d. que deux frontières f_i et f_j sont considérées comme identiques si $|f_i - f_j| < k$, avec k variable), le nombre de frontières retenues peut considérablement diminuer. Toutefois, le seuil d'acceptabilité de k dépend de la grandeur du taux de substitution à estimer et lorsqu'il est inconnu, une estimation au préalable de celui-ci est alors nécessaire. Nous préférons donc une autre alternative qui permet de gagner en plus du temps de calcul : le tirage aléatoire de triplets. Pour cela, nous fixons un seuil $s = 10^5$ au-delà duquel les triplets considérés sont obtenus par tirage aléatoire sur l'ensemble des triplets. Si le nombre « théorique » de triplets (obtenue par une fonction qui prend en considération le nombre de souches et le nombre de souches par date d'échantillonnage) est inférieur à ce seuil, alors tous les triplets sont utilisés. Dans le cas contraire seuls 10^5 triplets, choisis aléatoirement, sont considérés.

Non seulement ce principe permet de gagner considérablement de l'espace mémoire, mais il permet aussi d'avoir un gain, non négligeable, en temps de calcul puisque la méthode ne dépend plus du nombre de triplets (à partir d'un certain seuil), elle est bornée. La Figure 32 montre, sur un jeu de données de 200 simulations comprenant chacun 550 taxa, qu'il n'y a aucune différence (ou une différence négligeable) entre la précision d'estimation de la version considérant tous les triplets (sans choix aléatoire), soit environ 5×10^6 triplets, et celle utilisant le tirage aléatoire, donc 10^5 triplets, c'est-à-dire seulement 2% de la totalité des triplets possibles.

Figure 32. Performance en précision d'estimation avec ou sans choix aléatoire de triplets.
Ces graphiques indiquent en ordonnée la précision d'estimation (plus les valeurs sont petites, meilleures sont les estimations ; fonction déviation relative, cf. section 4.3.1.2) d'ULS sans ou avec choix aléatoire de triplets (en rouge et vert respectivement) pour différentes entrées (matrices de distances (DNAdist), arbres FastME et PhyML) et différentes longueurs d'alignement (300 et 1 000 sites). Les 200 simulations de ce jeu de données contiennent chacun 550 taxa (cf. section 4.3.1.1).



4.2.5 Description de l'algorithme

L'algorithme ULS permet d'estimer le taux de substitution relatif aux séquences hétérochrones dont la matrice de distances D est donnée en entrée (Figure 33). Le nombre n de séquences ainsi qu'un vecteur T contenant les intervalles de temps, exprimés en unité de temps, de chaque séquence entre leur date d'échantillonnage et la date d'échantillonnage la plus récente, sont aussi donnés en entrée. Cet algorithme renvoie un nombre qui correspond au taux de substitution ω, exprimé en substitutions par site et par unité de temps, à estimer.

Cet algorithme fonctionne en deux étapes. La première (lignes 2 à 11) parcourt l'ensemble des triplets et construit progressivement l'arbre AVL contenant les frontières et leurs coefficients temporaires (en $O(n^3 \log n)$). Il existe deux manières différentes de parcourir les triplets, soit en utilisant le

principe du tirage aléatoire (lignes 3 à 6), soit en parcourant la totalité des triplets (lignes 8 à 10). Le choix de considérer l'une ou l'autre manière est déterminé à la ligne 2. La deuxième étape (lignes 15 à 21) balaie l'ensemble ordonné des frontières pour rechercher le minimum global du critère (en $O(n^3)$). Lorsque tous les morceaux ont été parcourus et que le minimum global, correspondant alors à l'estimation du taux de substitution, est trouvé, l'algorithme le renvoie (ligne 22). La complexité algorithme est donc en $O(n^3 \log n)$, mais elle est bornée à partir d'un certain seuil de n.

Figure 33. Description de l'algorithme *Ultrametric Least Squares*.
Cet algorithme estime le taux de substitution ω à partir d'une matrice de distances D de taille n et de T un vecteur contenant pour chaque taxon i l'intervalle $T_i = t_0 - t_i$.

Entrée : D une matrice de distances, T un vecteur temps, n le nombre de taxa
1. $r \longleftarrow$ Créer un nœud AVL et l'initialiser avec la frontière $f_0 = 0$ et les coefficients temporaires $(a, b, c) = (0,0,0)$;
2. **si** $\mathrm{nombreTriplet}(n, T) \geq 10^5$ **alors**
3. **répéter**
4. Choisir un triplet au hasard ;
5. Ajouter ou rechercher dans r les frontières du triplet et actualiser leurs coefficients temporaires ;
6. **jusqu'à** 10^5 fois
7. **sinon**
8. **pour** chaque triplet **faire**
9. Ajouter ou rechercher dans r les frontières du triplet et actualiser leurs coefficients temporaires ;
10. **fin pour**
11. **fin si**
12. $t \longleftarrow$ Lister les nœuds de r à l'aide d'un parcours infixe ;
13. $(\omega, q) \longleftarrow (0, t[0].c)$;
14. $(a, b, c) \longleftarrow (0,0,0)$;
15. **pour** chaque élément de t **faire**
16. Mettre à jour les coefficients (a, b, c) ;
17. Calculer les coordonnées(ω_0, q_0) du minimum ;
18. **si** il est à considérer **et** $q_0 \leq q$ **alors**
19. $(\omega, q) \longleftarrow (\omega_0, q_0)$;
20. **fin si**
21. **fin pour**
22. **retourner** ω ;

4.2.6 Utilisation de la méthode dans le cas de taux variant par intervalle de temps

ULS peut facilement s'adapter à l'estimation de k taux de substitution $\omega_1, \cdots, \omega_k$ dans le cadre du modèle MRDT (Drummond *et al*, 2001), c'est-à-dire avec un taux de substitution par intervalle de temps entre deux dates d'échantillonnage consécutives. Soient deux souches i et j, échantillonnées

aux temps t_i et t_j, alors $\delta_{ij}(\omega_1, \cdots, \omega_k)$ (qui voit i et j comme contemporains) s'exprime par la relation

(9) $$\delta_{ij}(\omega_1, \cdots, \omega_k) = \hat{d}_{ij} + \sum_{m=1}^{i} \omega_m(t_m - t_{m-1}) + \sum_{m=1}^{j} \omega_m(t_m - t_{m-1}).$$

Il est impossible d'estimer simultanément ces k taux de substitution avec l'algorithme ULS. Pour estimer ces k taux de substitution, nous en supposons $k-1$ fixes et estimons le dernier taux, puis itérons le processus jusqu'à convergence. Dans ce cas, l'équation (9) est uniquement dépendante d'un seul taux de substitution et peut être exprimée comme à l'équation (1). L'algorithme ULS peut alors être utilisé pour estimer le taux de substitution non fixé.

Ce procédé nécessite d'abord d'initialiser les k taux de substitution, par exemple avec la valeur obtenue par ULS dans le cadre du modèle SRDT (où un seul taux de substitution est supposé). Ensuite, les taux de substitution sont fixés, hormis le $u^{\text{ème}}$, et la matrice de distances et le vecteur temps sont modifiés en conséquence, c'est-à-dire que pour chaque distance $\hat{d}_{ij}$ la mesure

$$\sum_{m=1, m\neq u}^{i} \omega_m(t_m - t_{m-1}) + \sum_{m=1, m\neq u}^{j} \omega_m(t_m - t_{m-1})$$

y est ajoutée et les intervalles de temps sont modifiés en retranchant les mesures

$$\sum_{m=1, m\neq u}^{i} t_m - t_{m-1}$$

et

$$\sum_{m=1, m\neq u}^{j} t_m - t_{m-1}$$

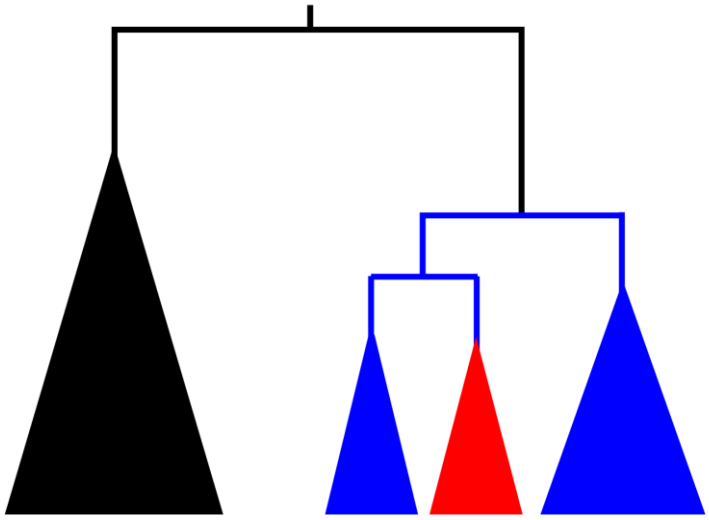
à T_i et à T_j respectivement. Dès lors, nous sommes dans le contexte du modèle SRDT et pouvons estimer le $u^{\text{ème}}$ taux de substitution. Cette procédure est ensuite itérée pour le $u+1^{\text{ème}}$ taux de substitution et, ainsi de suite, jusqu'au $k^{\text{ème}}$. Lorsque les k taux de substitution sont estimés et si au moins un est modifié significativement, alors la procédure est de nouveau itérée, mais en conservant cette fois-ci les dernières estimations des k taux de substitution comme valeurs initiales, et cela jusqu'à stabilisation des valeurs.

Ce procédé est facilement adaptable au modèle MRDTa, c'est-à-dire le modèle où l'utilisateur choisit lui-même les intervalles de temps pour lesquels il souhaite connaître un taux de substitution. Ces intervalles ne sont pas forcément en adéquation avec les dates de collecte, mais compris entre la date d'échantillonnage la plus ancienne et la plus récente. Toutefois, il ne peut y avoir plus d'intervalles de temps qu'en a le modèle MRDT sur le même jeu de donnée. En effet, il faut au moins avoir une souche par intervalle de temps pour que le taux de substitution correspondant puisse être estimé. Donc, le nombre maximum d'intervalles de temps est donné par le nombre de dates d'échantillonnage différentes moins une (une date est utilisée comme référence) (Drummond *et al*, 2001). Dans le cas contraire, il y a forcément un intervalle de temps qui ne contient pas de souches et le taux de substitution correspondant ne peut être estimé.

4.2.7 Utilisation de la méthode dans le cas de taux variant par lignage

Le modèle par lignage (horloges moléculaires locales) permet d'estimer un taux de substitution par lignage (ou sous-arbre). La donnée comprend un arbre (non nécessairement enraciné) et les sous-arbres où le taux de substitution est différent du taux global affecté au reste de l'arbre (Figure 34). Ce modèle peut, par exemple, être utilisé lorsque l'on considère plusieurs sous-types du VIH-1 dans une même phylogénie, on peut alors affecter à chaque sous-type un taux de substitution différent.

Figure 34. Schéma représentant le modèle par lignage.
Ce modèle prend en considération le fait qu'un ou des sous-arbres peuvent évoluer à des vitesses différentes de celle de la phylogénie globale, c'est-à-dire en considérant des horloges moléculaires locales. Dans cet exemple, il y a trois taux de substitution, un global (en noir) et deux locaux (en bleu et rouge).



Soit un arbre binaire A. L'ensemble de ses nœuds est noté V et l'ensemble de ses feuilles $F \subset V$. On pose $|F| = n$. Soit maintenant $\hat{d}: V \times V \to \mathbb{R}_+$ une mesure de distance telle que, pour tout i, j de V, $\hat{d}(i,j) = \hat{d}_{ij}$ renvoie la longueur du chemin reliant i avec j et soit P_{ij} l'ensemble contenant la

suite de nœuds consécutifs de ce chemin. Le premier nœud dans la suite P_{ij} est i et le dernier j. Supposons maintenant que chaque arête est associée à un taux de substitution. Le taux de substitution correspondant à l'arête ayant u et v comme sommets est noté ω_{uv} et $m = 2n - 3$ correspond au nombre d'arêtes.

La distance corrigée δ_{ij}, exprimée ici en unité de temps, entre deux feuilles i et j vues comme contemporaines, vaut

$$\delta_{ij} = T_i + T_j + \sum_{k \in P_{ij}} \frac{\hat{d}_{k(k+1)}}{\omega_{k(k+1)}}.$$

Procédons maintenant d'une manière analogue à celle du modèle MRDT. Soit c le nombre de taux de substitution à estimer. Au préalable, on initialise les c taux de substitution avec la valeur retournée par l'algorithme ULS, en considérant le modèle SRDT. Puis, fixons $c - 1$ taux de substitution et imagions que l'on souhaite estimer le taux de substitution ω_c. Soit maintenant, $V_c = \{(u, v) \in V^2 | \omega_{uv} = \omega_c\}$ l'ensemble d'arêtes correspondant au taux de substitution ω_c à estimer. Ainsi,

$$\delta'_{ij} = \omega_c \delta_{ij} = \omega_c \lambda_{ij} + \mu_{ij}$$

avec

$$\lambda_{ij} = T_i + T_j + \sum_{k \in P_{ij}, (k,(k+1)) \notin V_c} \frac{\hat{d}_{k(k+1)}}{\omega_{k(k+1)}}$$

et

$$\mu_{ij} = \sum_{k \in P_{ij}, (k,(k+1)) \in V_c} \hat{d}_{k(k+1)}.$$

Les termes λ_{ij} et μ_{ij} sont indépendants de ω_c. Ainsi, δ'_{ij} a la même forme que l'équation (1) et nous pouvons y appliquer le critère Q afin d'estimer le taux de substitution ω_c. L'algorithme ULS peut donc être appliqué, et on itère ce processus jusqu'à convergence des c taux de substitution.

4.2.8 Mise en œuvre

L'algorithme ULS est implémenté en langage C et présente une interface similaire à celle des logiciels de la suite PHYLIP (Felsenstein, 1993). Ce logiciel permet d'estimer le taux de substitution à partir d'une matrice de distances ou d'un arbre (raciné ou non) dont on extrait les distances patristiques. Il permet aussi d'estimer la date de l'ancêtre commun aux taxa (à l'aide d'un arbre UPGMA calculé d'après la matrice de distances corrigées) et propose l'enracinement d'un arbre avec la méthode de

minimisation de la variance spécifiquement adaptée aux arbres avec feuilles hétérochrones. Les adaptations d'ULS aux modèles MRDT, MRDTa et lignage sont aussi disponibles mais nécessiteraient d'être testées avec soin.

4.3 Confrontation aux autres méthodes de distances et à celle de référence (BEAST)

Nous avons confronté, sur données simulées, ULS aux autres méthodes de distances qui permettent d'estimer le taux de substitution sous les hypothèses du modèle SRDT (horloge moléculaire stricte et feuilles hétérochrones), à savoir *Pairwise-Distance*, *Root-to-Tip*, sUPGMA et TREBLE, ainsi qu'à la méthode probabiliste de référence BEAST (cf. Chapitre 2). Dans un second temps, ULS est appliquée sur deux jeux de données du sous-type C du virus de l'immunodéficience humaine de type 1 (VIH-1C).

4.3.1 Confrontation sur jeux de données simulées

À notre connaissance, il n'existe qu'un seul générateur d'arbres qui émette les hypothèses du modèle SRDT : *Serial SimCoal* (Anderson *et al*, 2005). Cependant, il génère des arbres sous le modèle du coalescent (Kingman, 1982), basé sur la génétique des populations, et est une approche différente au modèle phylogénétique classique qui est celui supposé par ULS. Nous avons donc généré nos propres jeux de données simulées, en adaptant le modèle de Yule (Yule, 1925), étendu par Raup et al. (1973) en y incluant un taux de mort constant, aux hypothèses du modèle SRDT.

4.3.1.1 Construction des jeux de données simulées

Le processus stochastique utilisé pour générer les jeux de données simulées démarre d'un individu. Puis, chaque individu vivant a autant de chance de donner naissance à un nouvel individu, jusqu'à en obtenir n. Dès que les n individus sont obtenus, $m < n$ individus sont choisis aléatoirement et disparaissent du processus (ils sont morts). Parmi ces individus morts, $N < m$ sont sélectionnés aléatoirement et sont considérés comme échantillonnés au temps t_{k-1}. Puis ce processus est recommencé avec les $n - m$ individus vivants jusqu'à en obtenir de nouveau n. Et ceci encore $k - 1$ fois, jusqu'au temps d'échantillonnage t_0. Ainsi, le sous-arbre contenant toutes les feuilles échantillonnées correspond à l'arbre souhaité. Ce processus est à l'opposé de celui du coalescent où l'on démarre avec n individus pour ne terminer qu'avec un seul individu (Steel & McKenzie, 2001), aussi appelé modèle de Hey (Hey, 1992). L'algorithme *GenTree* permet de générer un arbre avec le principe décrit ci-dessus et de telle sorte qu'il se passe a années entre deux dates d'échantillonnage (Figure 35).

Classiquement, sous le modèle de Yule, la probabilité qu'un évènement de spéciation survienne dans une lignée à l'instant t suit une loi exponentielle de paramètre λ, où λ représente le nombre moyen d'évènements de spéciation qui se produisent dans une lignée par unité de temps (Mooers *et al*, 2007), et de moyenne $1/\lambda$. Si k lignées sont présentes à un moment donné, alors l'instant t jusqu'au prochain évènement de spéciation suit aussi une loi exponentielle mais d'espérance $1/(k\lambda)$ (Steel & Mooers, 2009). Afin de simplifier cette hypothèse, nous supposons que l'espérance de la loi exponentielle fournit le temps jusqu'au prochain évènement de spéciation, soit $1/k$ lorsque k lignées sont présentes, en posant $\lambda = 1$. Ceci permet d'avoir des arbres dont les intervalles de temps sont identiques et non stochastiques.

Figure 35. Description de l'algorithme *GenTree*.
Cet algorithme génère un arbre sous les hypothèses du modèle SRDT.

Entrée : m le nombre d'individus morts, n le nombre d'individus à chaque temps d'échantillonnage, N le nombre d'individus échantillonnés à chaque temps d'échantillonnage, k le nombre de temps d'échantillonnage, a le nombre d'années entre deux dates d'échantillonnage et ω la valeur du taux de substitution souhaité.

Créer une feuille et la stocker dans un tableau T ;
$x \longleftarrow 1$;
$C \longleftarrow \sum_{i=n-m+1}^{n} \frac{1}{i}$;
répéter
    répéter
        Choisir aléatoirement une feuille f dans T ;
        Créer deux nouvelles feuilles qui sont les fils gauche (g) et droit (d) de f ;
        Supprimer f du tableau et y ajouter g et d ;
        $x \longleftarrow x + 1$;
        pour toutes les longueurs de branche l des feuilles de T **faire**
            $l \longleftarrow l + a\omega/(Cx)$;
        fin pour
    jusqu'à $x = n$
    Choisir aléatoirement N feuilles que l'on marque et que l'on supprime du tableau T ;
    Choisir aléatoirement $m - N$ feuilles que l'on supprime du tableau ;
    $x \longleftarrow x - m$;
jusqu'à k fois
pour chaque feuille i marquée **faire**
    Affecter le temps d'échantillonnage d_i/ω, où d_i est la distance séparant la feuille i de la racine ;
fin pour
$R \longleftarrow$ Extraire le sous-arbre contenant toutes les feuilles marquées ;
retourner R ;

Nous voulions que les paramètres utilisés pour générer les jeux de données reflètent au mieux la topologie des phylogénies intra- et inter-hôtes du VIH. Pour cela, nous avons respectivement utilisé un taux de mort à 995 et 750 sur 1 000 taxa à chaque temps d'échantillonnage ($m = \{995,750\}$,

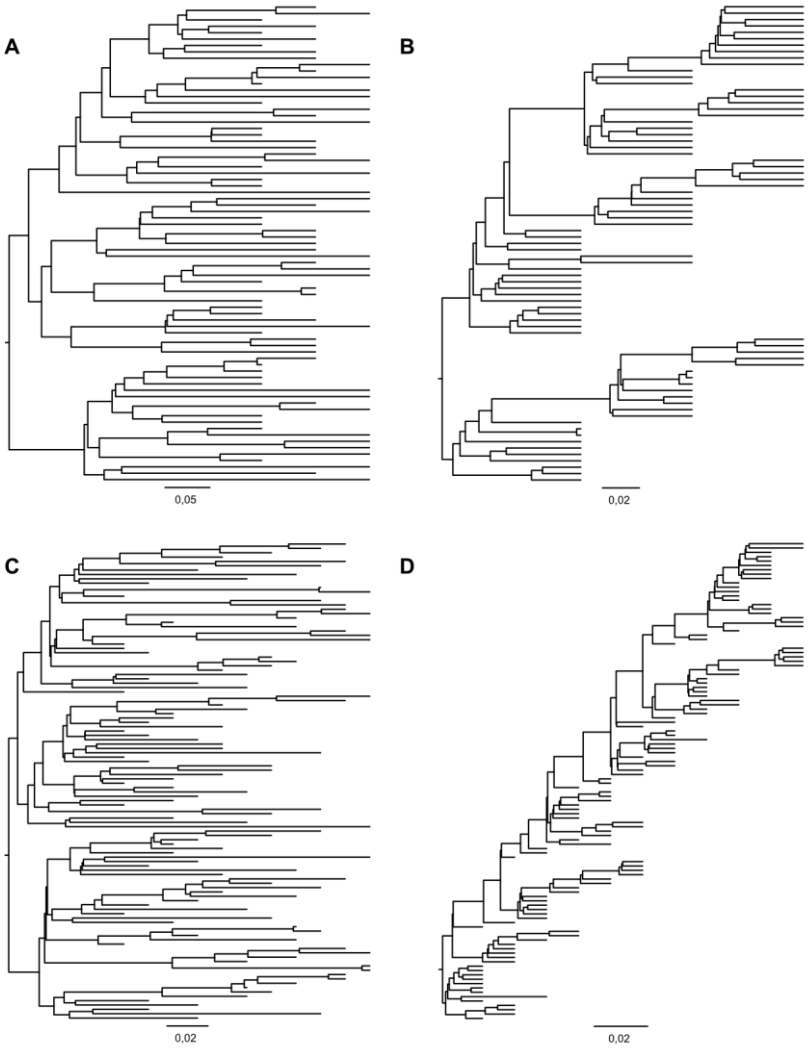
$n = 1000$). Pour chaque taux de mort, quatre jeux de données, contenant chacun 100 arbres, sont générés. Les deux premiers contiennent 3 temps d'échantillonnage chacun séparé de 10 ans ($k = 3$ et $a = 10$), avec respectivement 25 et 100 feuilles collectées à chaque date ($N = \{25,100\}$). Les deux derniers contiennent 11 temps d'échantillonnage chacun séparé de 2 ans ($k = 11$ et $a = 2$), avec respectivement 10 et 50 feuilles collectées à chaque date ($N = \{10,50\}$). Ces quatre jeux de données veulent représenter le suivi de l'infection au VIH pour un individu ($m = 995$) ou une population ($m = 750$) sur 20 ans, avec un échantillonnage de la population virale tous les 10 ans ou tous les 2 ans. Le taux de substitution attribué à chaque jeu de données est de 6×10^{-3} substitutions par site et par année. Il correspond approximativement à celui obtenu par Bello *et al*. (2008) sur la région *env* du génome ($5{,}8 \times 10^{-3}$ dans leur étude pour une estimation avec une horloge moléculaire stricte). Ainsi, nous avons huit collections d'une centaine d'arbres générés sous le modèle SRDT et reflétant l'évolution intra- et inter-hôte du VIH, avec un taux de substitution de 6×10^{-3} substitutions par site et par année. La Figure 36 montre quatre exemples de topologies extraites de ces jeux de données simulées.

Ces arbres servent ensuite de guide à SeqGen (Rambaut & Grassly, 1997) pour générer les alignements correspondants (un de 1 000 sites et l'autre de 300 sites) sous le modèle d'évolution F84 (Felsenstein, 1984; Kishino & Hasegawa, 1989), similaire au modèle d'évolution HKY85 (Hasegawa *et al*, 1985), avec une loi gamma de paramètre 1 à 8 catégories de taux et un taux de transition/transversion (Ts/Tv) à 2,5. Les fréquences des nucléotides sont respectivement de 0,35, 0,2, 0,2 et 0,25 pour les bases A, C, G et T. Ces paramètres correspondent approximativement à ceux obtenus sur *env* pour des virus appartenant au groupe M du VIH-1 (Posada & Crandall, 2001), groupe responsable de la pandémie mondiale. Ces 16 jeux de données (8 de longueurs 300 sites et 8 de longueurs 1 000 sites) servent de base pour nos analyses comparatives. Comme les formats d'entrée des différentes méthodes varient (alignements, matrices de distances, arbres), nous avons généré les matrices de distances correspondantes avec DNAdist v3.69 du package PHYLIP (Felsenstein, 1989). Le modèle d'évolution utilisé et les paramètres sont choisis en concordance avec ceux utilisés pour générer les alignements. À partir des matrices de distances, des arbres FastME v2.07 (Desper & Gascuel, 2002) sont calculés en utilisant l'option SPR pour parcourir l'espace des arbres. Thu Hien TO a généré les arbres calculés avec PhyML v3.0 (Guindon *et al*, 2010; Guindon & Gascuel, 2003) sous le modèle F84 et en laissant PhyML optimiser les paramètres. À nouveau l'option SPR est choisie pour parcourir l'espace des arbres. Les arbres ainsi obtenus sont aussi convertis en matrices de distances patristiques pour les méthodes prenant celles-ci en entrée.

Les méthodes de distances utilisées pour ces tests sont de notre propre implémentation. Toutefois, quand cela a été possible, nous avons vérifié les résultats obtenus par nos implémentations

contre celles des auteurs sur quelques jeux de données. L'algorithme *Root-to-tip* utilisé ici est celui implémenté dans la version 1.3 de Path-O-Gen, c'est-à-dire celui de la minimisation des résidus (cf. Chapitre 2). L'implémentation de TREBLE est une adaptation en C de celle disponible à l'adresse http://jacobian.wikidot.com/software (programmée pour R). Cette version correspond à celle publiée mais où le paramètre α est mis à 0 (selon l'auteur, lors d'une conversation par courriel ; cf. Chapitre 2). La valeur de pondération d'ULS correspond à celle de l'équation (6) pour $k = 1$. Pour BEAST v1.6.2, le modèle d'évolution choisi est HKY85 et le modèle démographique est *Constant Size* avec une horloge moléculaire stricte. La prior pour le paramètre *clock.rate* est une loi uniforme entre 0 et 1. La longueur de la chaîne de Markov par technique de Monte Carlo (MCMC) est de 5×10^6 générations avec un échantillonnage toutes les 5×10^3 générations.

Figure 36. Exemples de topologies d'arbre simulé.
Quatre exemples de topologies d'arbre extrait de nos jeux de données. Les arbres A et C (topologie approximant une phylogénie du VIH inter-hôte) proviennent des jeux de données ayant 750 morts (sur 1 000 taxa) par date d'échantillonnage et les arbres B et D (topologie approximant une phylogénie du VIH intra-hôte) des jeux de données ayant 995 morts par date d'échantillonnage. Les arbres A et B ont chacun 3 temps d'échantillonnage avec 25 feuilles par date d'échantillonnage et les arbres C et D ont chacun 11 dates d'échantillonnage avec 10 feuilles par date d'échantillonnage.



4.3.1.2 Performance en précision d'estimation

Connaissant, pour chaque jeu de données, le taux de substitution théorique ω (6×10^{-3} substitutions par site et par année), il est alors facile de mesurer la performance des méthodes en comparant les taux estimés $\Omega = \{\widehat{\omega}_k, k = 1, \cdots, 100\}$ à ω. Ainsi, nous définissons les fonctions déviation relative

(10) $$D(\omega, \Omega) = \frac{1}{\omega}\sqrt{\frac{1}{100}\sum_{k=1}^{100}(\widehat{\omega}_k - \omega)^2}$$

et biais relatif

(11) $$B(\omega, \Omega) = \frac{1}{\omega}\left(\frac{1}{100}\left(\sum_{k=1}^{100}\widehat{\omega}_k\right) - \omega\right).$$

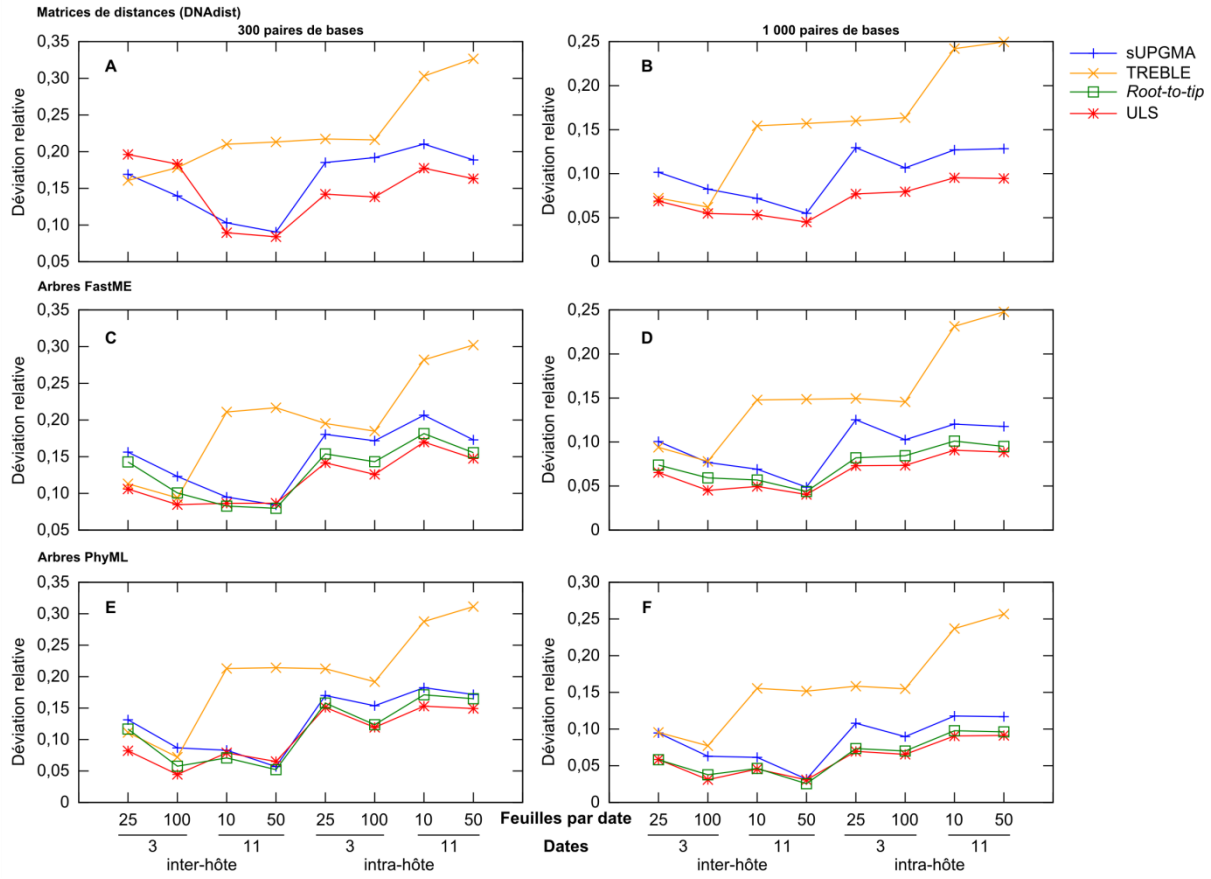
La fonction (10) mesure la performance moyenne des méthodes. Plus les valeurs de cette fonction sont petites, plus les estimations des taux de substitution sont proches de la valeur théorique. Ainsi, la méthode parfaite, celle qui estime à chaque fois le bon taux de substitution, a zéro comme valeur de déviation relative. La fonction (11) mesure une autre information, la tendance moyenne à sur- ou sous-estimer le taux de substitution réel. Si sa valeur est négative alors les estimations du taux de substitution sont majoritairement sous-estimées, tandis que si elle est positive, les estimations sont majoritairement surestimées.

La Figure 37 montre pour chaque jeu de données et pour chaque entrée possible (matrices de distances (DNAdist), arbres FastME et PhyML) la performance en précision d'estimation (déviation relative) des méthodes de distances testées (sUPGMA en bleu, TREBLE en orange, *Root-to-tip* en vert et ULS en rouge). Les performances de la méthode *Pairwise-Distance* sont aussi mesurées, mais elles sont trop faibles pour être représentées. En effet, pour les jeux de données avec 750 morts (inter-hôte) les valeurs de déviation relative sont toujours supérieures à 0,9 et pour les jeux de données avec 995 morts (intra-hôte) elles sont toujours supérieures à 0,24.

Comme attendu, les performances en précision d'estimation augmentent lorsque les séquences contiennent plus d'information, c'est-à-dire plus de nucléotides. Pour les jeux de données intra-hôtes, la méthode ULS est la plus performante, parfois égalée par la régression linéaire *Root-to-tip*, et cela quelle que soit la longueur des séquences ou le format d'entrée (arbres ou matrices de distances). La question est autre sur les jeux de données inter-hôte. En considérant 11 temps d'échantillonnage, la performance en précision d'estimation d'ULS semble à chaque fois égaler la meilleure des autres méthodes, hormis sur le graphique B avec 10 feuilles par temps

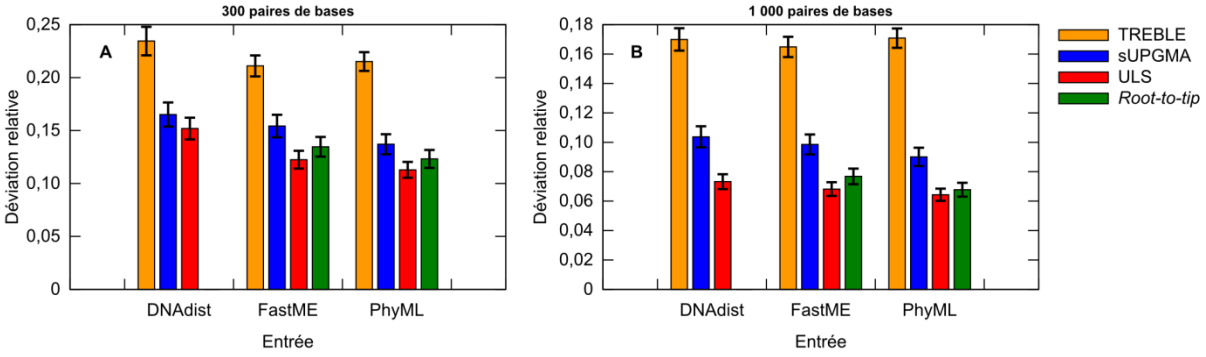
d'échantillonnage où elle est la plus performante. En revanche, avec les jeux de données contenant 3 dates d'échantillonnage, la performance d'ULS est souvent égalée, voire dépassée par d'autres méthodes, comme sur le graphique A. Or, dans ce dernier cas, la distance paire à paire moyenne avoisine les 0,6 substitutions par site, tandis que sur les autres jeux de données elle avoisine les 0,2 substitutions par site, sauf sur les jeux de données intra-hôte avec 11 temps d'échantillonnage où elle avoisine les 0,1 substitutions par site. Cela suggère que la méthode ULS reste dépendante de la précision d'estimation des distances (elles deviennent meilleures au fur et à mesure que la longueur des séquences augmentes où que les distances sont petites). D'ailleurs, les précisions d'estimation de la méthode TREBLE semblent aussi fluctuer en fonction de cette observation (elles deviennent de plus en plus précises au fur et à mesure que la distance paire à paire moyenne augmente, à l'inverse de ce que l'on attend). Remarquons que ce défaut est corrigé avec les arbres FastME et semble inexistant avec les arbres PhyML.

Figure 37. Performance en précision d'estimation des différentes méthodes de distances (fonction déviation relative). Graphiques montrant les valeurs de la fonction déviation relative en ordonnée pour chaque méthode de distances (sUPGMA en bleu, TREBLE en orange, *Root-to-tip* en vert et ULS en rouge) et pour chacun des 8 jeux de données simulées. Les graphiques A, C et E sont obtenus sur l'alignement de 300 sites et les graphiques B, D et F sur celui de 1 000 sites. Les résultats des graphiques A et B sont obtenus à partir de matrices de distances (DNAdist), les graphiques C et D à partir des arbres FastME et les graphiques E et F à partir des arbres PhyML. La méthode *Pairwise-Distance* n'est pas reportée dans les graphiques à cause de valeurs très différentes. En effet, quelle que soit l'entrée, les valeurs de la déviation relative en inter-hôte (750 morts) sont toujours supérieures à 0,9 et en intra-hôte (995 morts) toujours supérieures à 0,24.



Afin de déterminer quelle méthode est, dans l'absolu, la plus performante, leur précision d'estimation (déviation relative) est calculée sur les 800 simulations pour chaque longueur de séquences (300 et 1 000 paires de bases) et pour chaque entrée (matrices de distances (DNAdist), arbres FastME et PhyML) (Figure 38). Les intervalles de confiance à 95% sont précisés au sommet de chaque barre. Contre sUPGMA et TREBLE, la méthode ULS est toujours la plus performante, sauf pour des séquences de 300 paires de bases et avec des matrices de distances où l'intervalle de confiance recouvre celui de sUPGMA. Dans ce cas, la perte en précision d'estimation provient des deux jeux de données inter-hôtes avec 3 dates d'échantillonnage (cf. paragraphe précédent). Autrement, la méthode ULS est toujours, en moyenne, plus performante que la méthode *Root-to-tip*, mais nous ne pouvons affirmer que la précision d'estimation d'ULS est significativement meilleure que celle de *Root-to-tip*, étant donné que les intervalles de confiances à 95% sont systématiquement recouvrants. En revanche, un test du signe qui prend en considération le fait que les échantillons comparés proviennent de la même population (ce qui n'est pas pris en compte en comparant les intervalles de confiance), indique qu'ULS est significativement meilleure que la régression linéaire *Root-to-tip* avec les arbres FastME ($p < 0{,}01$; 445 contre 355 [resp. 485 contre 315] avec des séquences de 300 [1 000] paires de bases), mais rien ne peut être affirmé avec les arbres PhyML ($p = 0{,}15$ [421 contre 379] et $p = 0{,}07$ [426 contre 374] pour 300 et 1 000 paires de bases respectivement). Sur ce graphique, nous observons aussi que la précision d'estimation des méthodes (hormis TREBLE) est, en moyenne, plus performante avec des arbres PhyML qu'avec des arbres FastME, bien que généralement les intervalles de confiance soient recouvrants.

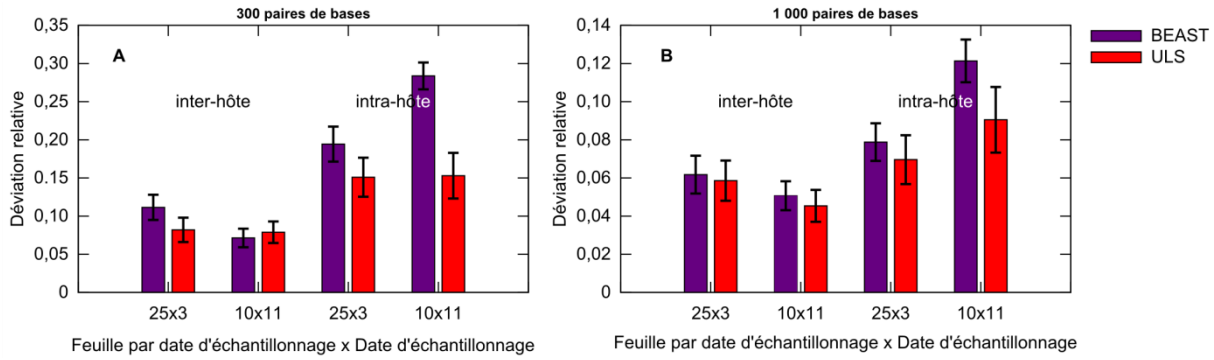
Figure 38. Performance en précision d'estimation (déviation relative) pour toutes simulations confondues.
Ces graphiques représentent la précision d'estimation (déviation relative) des différentes méthodes testées (de gauche à droite : TREBLE en orange, sUPGMA en bleu, ULS en rouge et *Root-to-tip* en vert) sur l'ensemble des 800 simulations et cela pour chaque entrée (matrices de distances (DNAdist), arbres FastME et PhyML) et chaque longueur d'alignement (300 sites pour le graphique A et 1 000 sites pour le graphique B). Les intervalles de confiance à 95% sont indiqués au sommet de chaque barre. Les performances de la méthode *Pairwise-Distance* sont toujours supérieures à 0,70 et ne sont donc pas représentées.



Les précisions d'estimation de la méthode de distances ULS (en rouge) sont comparées avec celles de la méthode probabiliste de référence BEAST (en mauve) (Figure 39). Ces comparaisons sont seulement faites sur les petits jeux de données inter- et intra-hôtes (3 et 11 dates d'échantillonnage avec

respectivement 25 et 10 feuilles par date d'échantillonnage), en raison du temps de calcul prohibitif nécessaire à BEAST sur les grands jeux de données. Les précisions d'estimation d'ULS sont toujours, en moyenne, plus performantes que celles de BEAST (en particulier sur les jeux de données intra-hôtes), exception faite sur un jeu de donnée (graphique A, 10x11). Mais dans ce dernier cas, et dans d'autres (surtout en inter-hôte), la différence de précision n'est pas significative puisque les intervalles de confiance sont recouvrants. Donc ULS est au pire équivalent à BEAST. La perte en précision d'estimation de BEAST sur les jeux de données intra-hôte provient sans soute du fait que BEAST est basé sur le modèle du coalescent, en opposition avec ces jeux de données qui sont générés avec un modèle de spéciation, or il est impossible de choisir un tel modèle avec la version de BEAST utilisée.

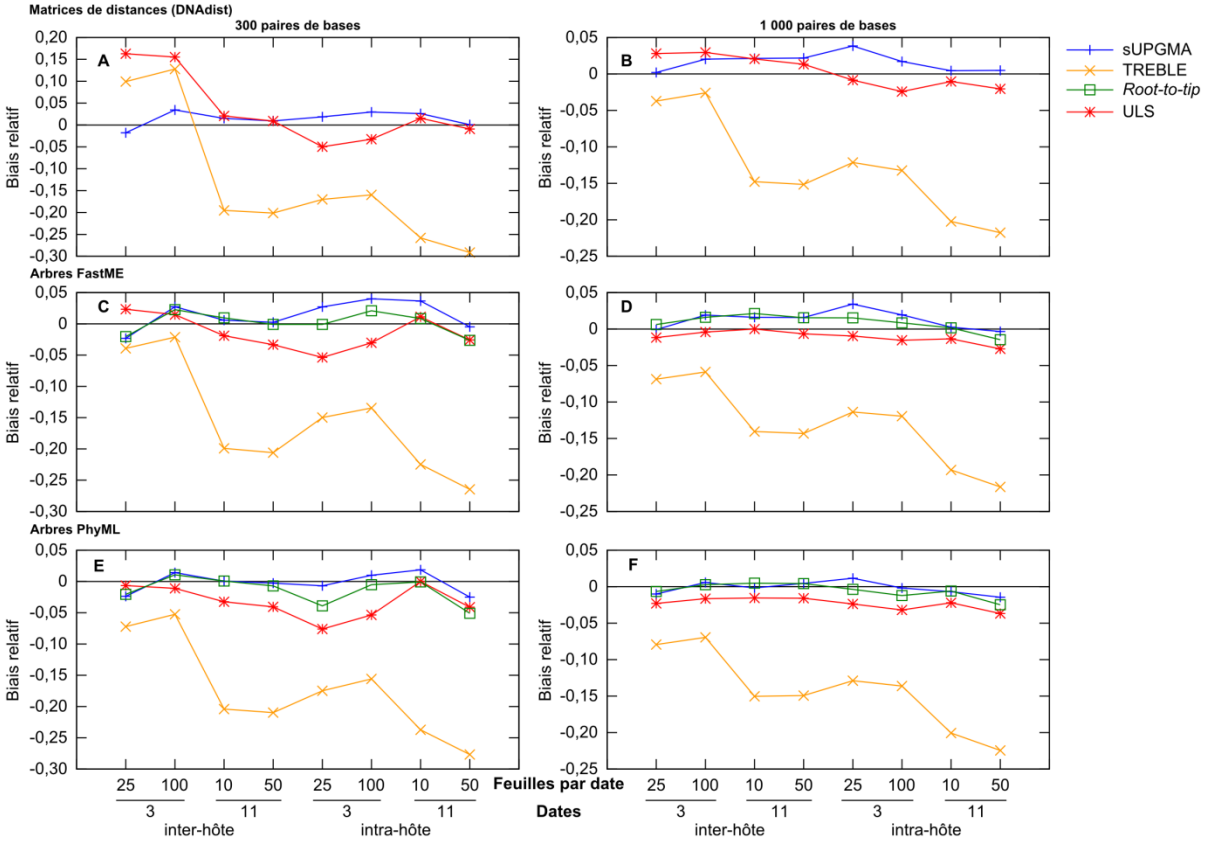
Figure 39. Comparaison de la précision d'estimation entre BEAST et ULS.
Ces graphiques montrent les précisions d'estimation en ordonnée de la méthode de distances ULS (sur arbres PhyML ; en rouge) et celles de la méthode probabiliste de référence BEAST (en mauve). Seuls les petits jeux de données inter-hôte (750 morts) et intra-hôte (995 morts) sont utilisés, à savoir ceux avec 25 feuilles par date d'échantillonnage et 3 dates (25x3) et ceux avec 10 feuilles par date d'échantillonnage et 11 dates (10x11), avec des séquences de 300 (graphique A) et 1 000 (graphique B) paires de bases. Les intervalles de confiance à 95% sont indiqués au sommet de chaque barre.



Les tendances des méthodes de distances (sUPGMA en bleu, TREBLE en orange, *Root-to-tip* en vert et ULS en rouge) à sur- (biais relatif positif) ou sous-estimer (biais relatif négatif) le taux de substitution sur les différents jeux de données sont présentées à la Figure 40. Le biais de la méthode *Pairwise-Distance* n'est pas présenté puisqu'il est toujours inférieur à –0,9 sur les jeux de données inter-hôte (750 morts) et toujours inférieur à –0,17 sur les jeux de données intra-hôte (995 morts). Comme on s'y attend, le biais est moins important lorsque les séquences ont 1 000 paires de bases. La méthode TREBLE a tendance à sous-estimer le taux de substitution (excepté pour les deux premiers cas du graphique A). Cela provient du fait qu'elle suppose initialement que le taux de substitution est zéro. Une correction à ce problème est apportée par les auteurs par l'instauration d'un critère qui rejette successivement les *outgroups* invalides à l'aide d'un processus itératif, mais ce critère n'est pas mis en œuvre dans la dernière version (pour R) proposée par les auteurs. En ce qui concerne les autres méthodes, il n'y a pas de tendance particulière qui ressort (tantôt positif, tantôt négatif). Remarquons, tout de même, qu'ULS semble sous-estimer le taux de substitution lorsque cette méthode utilise des arbres en entrée et à le surestimer avec des matrices de distances. De plus, le

biais d'ULS est plus important que celui des méthodes sUPGMA et *Root-to-tip* avec des arbres PhyML. Ces dernières méthodes, quant à elles, ont tendance à surestimer le taux de substitution sur les arbres FastME, et sUPGMA semble surestimer le taux de substitution avec des matrices de distances.

Figure 40. Biais dans les estimations des différentes méthodes de distances (fonction biais relatif).
Graphiques montrant les valeurs de la fonction biais relatif en ordonnée pour chaque méthode de distances (sUPGMA en bleu, TREBLE en orange, *Root-to-tip* en vert et ULS en rouge) et pour chacun des 8 jeux de données simulées. Les graphiques A, C et E sont obtenus de l'alignement de 300 sites et les graphiques B, D et F de celui de 1 000 sites. Les résultats des graphiques A et B sont obtenus à partir des matrices de distances (DNAdist), les graphiques C et D à partir des arbres FastME et les graphiques E et F à partir des arbres PhyML. La méthode *Pairwise-Distance* n'est pas reportée dans les graphiques à cause de valeurs très différentes. En effet, quelle que soit l'entrée, les valeurs du biais relatif en inter-hôte (750 morts) sont toujours inférieures à –0,9 et en intra-hôte (995 morts) toujours inférieures à –0,17.



En résumé, ces résultats, sur données simulées, suggèrent qu'ULS est plus précise que les méthodes de distances *Pairwise-Distance*, sUPGMA et TREBLE. Elle est aussi plus précise que la régression linéaire *Root-to-tip* avec en entrée des arbres FastME ou sur des jeux de données inter-hôtes, tandis qu'elle est équivalente à cette dernière sur des jeux de données intra-hôtes ou sur des arbres PhyML. Elle est aussi plus précise que la méthode probabiliste BEAST sur des jeux de données intra-hôtes et au pire équivalente à cette dernière sur des jeux de données inter-hôtes.

4.3.1.3 Performance en temps de calcul

Après avoir présenté la performance en précision d'estimation des différentes méthodes d'estimation de taux de substitution, nous montrons dans le Tableau 3 la performance de ces mé-

thodes (*Pairwise-Distance*, sUPGMA, TREBLE, *Root-to-tip*, ULS avec ou sans choix aléatoire et BEAST) en temps de calcul sur l'ensemble des jeux de données simulées (soit 1 600 jeux de données). À titre indicatif, nous présentons aussi les temps de calcul nécessaire aux outils d'inférence phylogénétique (DNAdist, FastME et PhyML) dont dépendent les méthodes de distances. Les temps de calcul sont donnés en minutes et lorsqu'ils dépassent le jour de calcul, ils sont donnés approximativement en jours. Notons que le temps de calcul de BEAST est seulement donné pour une partie des jeux de données simulées (ceux dont le nombre de feuilles dans l'arbre est inférieur ou égal à 110 alors que certains jeux de données vont jusqu'à 550 feuilles).

Tableau 3. Performance en temps de calcul des différentes méthodes d'estimation de taux de substitution.
Ce tableau présente les temps de calcul (en minutes) nécessaires à chaque méthode d'estimation de taux de substitution pour estimer les dits taux sur les 1 600 jeux de données simulées et pour les différentes entrées possibles (alignements, matrices de distances ou arbres). À titre d'information, nous indiquons aussi le temps de calcul de chaque méthode d'inférence phylogénétique utilisée pour générer l'ensemble des jeux de données simulées. Notons que le temps de calcul de la méthode BEAST est uniquement estimé sur les jeux de données comptant au plus 110 feuilles, alors que certaines simulations en comptabilisent 550.

	Entrées		
	Alignements	Matrices de distances	Arbres
Méthodes d'inférence phylogénétique			
DNAdist	≈ 2 jours 200[a]	-	-
FastME	-	109	-
PhyML	≈ 30 jours ≈ 2 jours[a]	-	-
Méthodes d'estimation de taux de substitution			
Pairwise-Distance	-	5	-
sUPGMA	-	14	-
TREBLE	-	60	-
Root-to-tip	-	-	19
ULS sans choix aléatoire	-	367	293
ULS avec choix aléatoire	-	38	30
BEAST	≈ 129 jours[a]	-	-

[a] uniquement les petits jeux de données (<110 feuilles par phylogénie, alors que certains vont jusqu'à 550).

Ces résultats montrent qu'ULS n'est pas la méthode d'estimation la plus rapide, les méthodes *Pairwise-Distance* et sUPGMA sont plus rapides qu'ULS, mais elles ne sont pas très performantes en précision d'estimation (cf. section précédente). Le temps de calcul de la méthode ULS est quasiment multiplié par 10 entre la version avec et sans le choix aléatoire. Dans le cas où l'on considère le choix aléatoire, il faut approximativement 30 minutes de calcul pour obtenir l'ensemble des estimations. Rappelons que les précisions d'estimation entre ces deux versions sont similaires (cf. section 4.2.4.3). Cette amélioration en fait une méthode très rapide (moins de 5 secondes sur un arbre avec 550 feuilles). Le temps de calcul de la méthode *Root-to-tip* est assez semblable au notre (environ 10 minutes d'écart en faveur de *Root-to-tip*), mais rappelons que cette méthode reste dépendante du nombre de feuilles dans la phylogénie, ce qui n'est plus le cas avec ULS en considérant le choix aléatoire. Notons que le temps de calcul d'ULS est plus rapide avec un arbre en entrée qu'avec une ma-

trice de distances, cela provient du format d'encodage des arbres et des matrices où la quantité de données à lire (nombre de caractères) est beaucoup moins importante avec des arbres au format NEWICK, qu'avec des matrices de distances. Le temps de calcul de la méthode BEAST, avoisinant les 129 jours de calcul mais uniquement sur les petits jeux de données (au plus 110 feuilles), en fait la méthode d'estimation de taux de substitution la plus lente, et cela même en considérant des arbres PhyML en entrée puisque, sur l'ensemble des petits jeux de données, il faut environ 2,3 jours de calcul à PhyML pour inférer les arbres. D'autant plus que sa précision d'estimation est équivalente à (ou moins bonne que) celle d'ULS sur des arbres PhyML. Enfin, notons que le temps de calcul des arbres FastME (en considérant bien sûr le temps nécessaire au calcul des matrices de distances) est beaucoup plus rapide que celui des arbres PhyML et pour une précision d'estimation quasi-équivalente (cf. section précédente). En résumé, sur un jeu de données contenant 550 feuilles, les temps de calcul des méthodes de distances sont approximativement de 5 secondes pour ULS, 1 seconde pour *Pairwise-Distance*, 2 secondes pour sUPGMA, 3 secondes pour *Root-to-tip* et 10 secondes pour TREBLE. Le temps de calcul, avec des séquences de 1 000 paires de bases, pour inférer un arbre PhyML est approximativement de 2 heures, celui d'un arbre FastME de 6 secondes et celui d'une matrice de distances DNAdist de 8 minutes. Quant à BEAST, il met environ 30 minutes sur un jeu de données de 1 000 paires de bases contenant 110 feuilles.

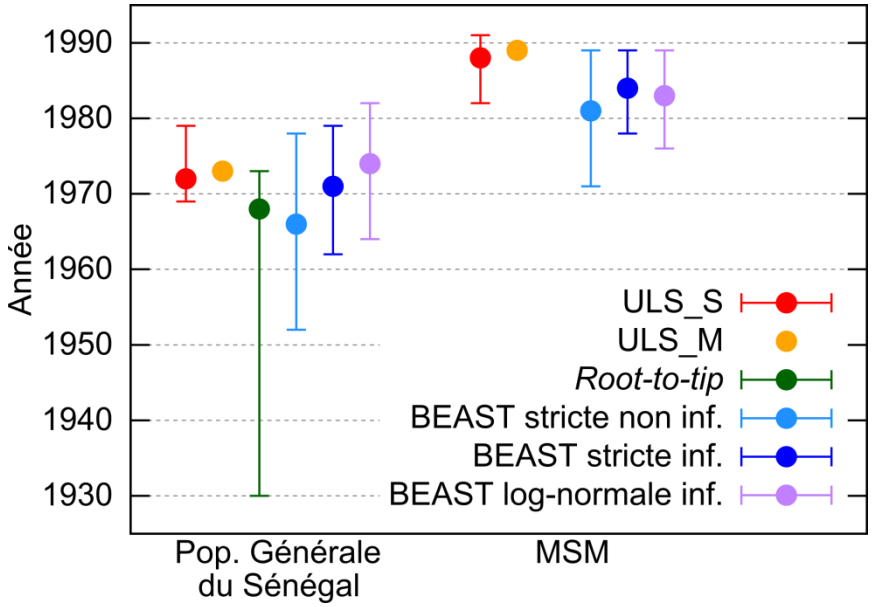
4.3.2 Application au sous-type C du VIH-1

Notre méthode est testée sur deux jeux de données contenant chacun des séquences du sous-type C du VIH-1 (VIH-1C). Le premier est extrait de l'étude sur l'origine géographique et temporelle de l'épidémie du VIH-1C au Sénégal (Chapitre 5) et l'autre de l'étude épidémiologique mondiale du VIH-1C (Chapitre 6).

Le premier jeu de données contient 56 séquences *pol* du VIH-1C collectées au Sénégal (Jung *et al*, 2012). La conception de l'alignement de 1 011 sites est décrite au Chapitre 5. À partir de cet alignement, un arbre de maximum de vraisemblance est calculé avec PhyML v3.0 (Guindon *et al*, 2010; Guindon & Gascuel, 2003) sous le modèle GTR+I+Γ4 (*general time reversible* avec une loi gamma à 4 catégories de taux et des sites invariants), en accord avec Jung *et al*. (2012). Les paramètres du modèle d'évolution sont estimés par PhyML. L'option SPR (*Subtree Pruning and Regrafting*) est choisie afin d'explorer l'espace des arbres. Les intervalles de confiance à 95% sont obtenus à partir de 100 arbres calculés de la même manière, mais sur la base d'alignements obtenus avec la technique du *bootstrap* par le logiciel *seqboot* v3.69 du package PHYLIP (Felsenstein, 1989). Le taux de substitution estimé par ULS sur ce jeu de données est de $2{,}01 \times 10^{-3}$ [$1{,}78 \times 10^{-3}$; $2{,}63 \times 10^{-3}$] substitutions par site et par année, tandis que celui estimé par BEAST dans Jung *et al*. (2012) est de $1{,}59 \times 10^{-3}$

[$1{,}02 \times 10^{-3}$; $2{,}19 \times 10^{-3}$] substitutions par site et par année, sous le modèle d'horloge moléculaire stricte et avec une *prior* non informative (c.-à-d. les mêmes conditions que dans ULS). Ces résultats sont largement recouvrants et la différence n'est pas statistiquement significative. Cependant, le taux de substitution d'ULS semble plus élevé que celui de BEAST et notre intervalle de confiance à 95% plus étroit. La régression linéaire *Root-to-tip*, estime le taux de substitution à $1{,}47 \times 10^{-3}$ [$7{,}06 \times 10^{-4}$; $2{,}11 \times 10^{-3}$] substitutions par site et par année, très proche de celui de BEAST. La date de l'ancêtre commun aux souches du Sénégal est estimée par ULS à 1972 [1969 ; 1979] et par BEAST à 1966 [1952 ; 1978] (Figure 41). L'approche pour ULS consiste à corriger les distances évolutives entre les feuilles non contemporaines et à employer la méthode UPGMA (cf. Chapitre 1) pour reconstruire l'arbre enraciné, dans lequel la distance évolutive qui sépare les feuilles contemporaines de la racine est déduite. Ici aussi ULS propose une estimation plus élevée, mais toujours avec des intervalles de confiance recouvrants et pas de différence significative. De plus, l'intervalle d'ULS est nettement plus serré (environ 10 ans) que celui de BEAST (environ 30 ans). La régression linéaire *Root-to-tip* estime l'ancêtre commun à 1968 [1930 ; 1973], à nouveau proche de l'estimation de BEAST mais avec un intervalle de confiance très large (plus de 40 ans). Cependant, en considérant pour BEAST une *prior* informative (d'après des estimations publiées), ses estimations deviennent assez similaires à celles d'ULS : taux de substitution à $1{,}85 \times 10^{-3}$ [$1{,}36 \times 10^{-3}$; $2{,}37 \times 10^{-3}$] substitutions par site et par année et date de l'ancêtre commun à 1971 [1962 ; 1979]. En revanche, le choix de la *prior* (informative ou non) influe très peu sur les estimations de la date de l'ancêtre commun aux souches des hommes ayant des rapports sexuels avec des hommes (MSM), et est estimée par BEAST au début des années quatre-vingt. Par exemple, pour la *prior* non informative, BEAST date l'ancêtre commun aux souches isolées chez les MSM à 1981 [1971 ; 1989] et pour une *prior* informative à 1984 [1978 ; 1989]. Quant à ULS, il l'estime à 1988 [1982 ; 1991], environ 5 ans plus tard que BEAST (mais avec des intervalles compatibles et donc des différences non significatives). La stabilité de BEAST dans les estimations de la date de l'ancêtre commun aux souches des MSM s'étend aussi au-delà du modèle d'horloge moléculaire. Par exemple, avec une horloge moléculaire relâchée en log-normal et une prior informative, BEAST l'estime à 1983 [1976 ; 1989]. En revanche, sous ce même modèle, l'estimation de la date de l'ancêtre commun aux souches collectées au Sénégal (1974 [1964 ; 1982]) s'accorde mieux avec celle d'ULS. Le temps de calcul nécessaire à BEAST sur ce jeu de données avoisine les 12 heures de temps de calcul, tandis que le temps de calcul de la méthode ULS est à peine de 4 secondes (en considérant, en plus, le calcul de l'intervalle de confiance). Notons cependant que PhyML met environ 4 minutes pour estimer une phylogénie (donc environ 6 heures et demi sont nécessaires à PhyML pour calculer les 100 phylogénies).

Figure 41. Estimations temporelles d'ULS sur deux jeux de données du sous-type C du VIH-1.
Ce graphique montre les estimations par ULS de la date de l'ancêtre commun aux souches collectées au Sénégal et celle de l'ancêtre commun aux souches isolées chez les MSM à partir d'une phylogénie contenant uniquement les séquences collectées au Sénégal (ULS_S, en rouge ; cf. Chapitre 5) et d'une autre phylogénie contenant l'ensemble des séquences du VIH-1C (ULS_M, en orange ; cf. Chapitre 6). L'estimation de la date de l'ancêtre commun des souches collectées au Sénégal par *Root-to-tip* (en vert) et les estimations de BEAST du Chapitre 5 (horloge moléculaire stricte avec une *prior* non informative, en bleu clair, et informative, en bleu foncé, et horloge moléculaire relâchée en log-normal avec une *prior* informative, en mauve) sont aussi présentées. Les estimations de ces deux dernières méthodes sont réalisées à partir d'un jeu de données contenant uniquement les séquences collectées au Sénégal. La date de l'ancêtre commun aux souches du VIH-1C est estimée par ULS à 1964 et par *Root-to-tip* à 1782.



Le second jeu de données considéré contient 3 609 taxa. Les détails de la conception de l'alignement de 1 011 sites et les paramètres utilisés par PhyML pour inférer la phylogénie sont donnés au Chapitre 6. Sur ce jeu de données (*ingroup* uniquement) qui contient l'ensemble des souches *pol* disponibles du VIH-1C, ULS estime le taux de substitution à $4{,}70 \times 10^{-3}$ substitutions par site et par année et la date de l'ancêtre commun à 1964. L'estimation de la date de l'ancêtre commun d'ULS semble être du même ordre de grandeur que celle admise pour les souches du sous-type C (Abecasis *et al*, 2009; Rousseau *et al*, 2007; Travers *et al*, 2004). Par exemple, Rousseau *et al*. (2007) l'estiment à 1961 [1947 ; 1962] et le taux de substitution à $5{,}1 \times 10^{-3}$ [$3{,}9 \times 10^{-3}$; $5{,}3 \times 10^{-3}$] substitutions par site et par année ; mais à partir de génomes presque complets. Dalai *et al*. (2009), quant à eux, estiment avec BEAST la date de l'ancêtre commun aux souches du sous-type C collectées au Zimbabwe à 1972 [1969-1974] et un taux de substitution moyen à $2{,}33 \times 10^{-3}$ substitutions par site et par année sur des séquences *pol*, suggérant que notre taux de substitution est encore une fois plus important que celui de BEAST ; mais les données sont bien différentes. Sur ce jeu de données, *Root-to-tip* estime un taux de substitution de $8{,}04 \times 10^{-4}$ substitutions par site et par année. Ce taux est très nettement inférieur à ceux mentionnés ci-dessus qui semblent être plus en accord avec l'estimation d'ULS. La date de l'ancêtre commun estimée par *Root-to-Tip*, qui est de 1782, est complètement différente de celle qui est communément admise par la communauté scientifique (début

de la deuxième moitié du xx^{e} siècle) et montre ainsi la limite de cette méthode. Notons cependant que cette date de référence se fonde uniquement sur les estimations moyennes de plusieurs références bibliographiques, revues dans Hemelaar *et al*. (2012), et n'intègre pas l'information des intervalles de confiance associés à ces estimations, qui sont parfois très larges (couvrant la période de 1933 à 1973). Ces intervalles de confiance montrent l'incertitude associée aux estimations ponctuelles qui vient probablement du manque de signal dans les données étudiées.

Cette phylogénie inclut aussi les souches collectées au Sénégal, y compris celles isolées chez les MSM (cf. ci-dessus), et les dates associées à leur ancêtre commun peuvent donc être estimées. La date de l'ancêtre commun aux souches collectées au Sénégal est estimée par ULS à 1973 et celle de l'ancêtre commun aux souches des MSM à 1989 (Figure 41). Ces deux estimations sont tout à fait cohérentes avec celles estimées précédemment en ne considérant que les souches collectées au Sénégal (cf. ci-dessus). Sur ce jeu de données, ULS met moins de 3 minutes à estimer le taux de substitution et la date de l'ancêtre commun, tandis que *Root-to-tip*, dépendant du nombre de feuilles de la phylogénie, met un peu plus de 5 minutes.

4.4 Conclusion

Nous présentons une nouvelle méthode de distances, *Ultrametric Least Squares* (ULS), basée sur le principe des moindres carrés, qui permet d'estimer le taux de substitution sous les hypothèses du modèle *Single Rate Dated Tips* (SRDT) (feuilles hétérochrones et horloge moléculaire stricte). Pour ce faire, elle minimise un critère parabolique par morceaux qui mesure l'ultramétricité d'une distance en $O(n^3 \log n)$, où n est le nombre de feuilles. Un algorithme de type Monte Carlo borne cette complexité, et cela sans perte de précision, à partir d'un certain seuil déterminé en fonction de n et du nombre de feuilles par date d'échantillonnage. Cette méthode est aussi étendue aux modèles *Multiple Rates with Dated Tips* (MRDT) et *Different Rate* (DR) mais seulement avec des horloges moléculaires locales. L'implémentation de cette méthode en langage C fournit aussi l'opportunité d'estimer la date de l'ancêtre commun aux souches du jeu d'entrée, ainsi que d'enraciner une phylogénie en considérant les dates de prélèvement associées à chaque feuille.

Le principe itératif utilisé afin d'adapter ULS au modèle d'horloge moléculaire MRDT et à l'estimation de plusieurs taux de substitution par lignage (horloges moléculaires locales) peut être appliqué à n'importe quelle autre méthode d'estimation de taux de substitution faisant les hypothèses du modèle SRDT. À notre connaissance, seules deux autres méthodes permettent d'estimer le taux de substitution sous le modèle MRDT : sUPGMA, une méthode de distances, et TipDate, une méthode probabiliste (Drummond *et al*, 2001). Les horloges moléculaires locales sont très appréciées

puisqu'elles reflètent mieux la réalité des données surtout quand, par exemple, plusieurs sous-types du VIH sont étudiés en même temps. Toutefois, ces hypothèses de taux variant par branche ou par lignée nécessitent la donnée d'une phylogénie. Notre modèle suppose que l'on connaisse *a priori* les lignées évoluant avec un taux de substitution différent et que chaque lignée évolue avec le même taux. Yoder et Yang (2000) présentent un exemple d'application avec ce modèle. D'autres approches existent, comme celle proposée par Sanderson (1997) qui suppose que chaque branche de la phylogénie évolue avec un taux unique, mais en supposant une auto-corrélation des taux ; il minimise l'écart entre les taux de substitution d'une même lignée. Adapter la méthode ULS à un tel modèle serait un atout supplémentaire pour cette méthode.

La comparaison entre la précision d'estimation d'ULS et celle des méthodes de distances sUPGMA, TREBLE, ainsi que celle des régressions linéaires *Root-to-tip* et *Pairwise-Distance* sur différents jeux de données simulées indique qu'ULS est, en moyenne, la méthode la plus précise. Malgré cela, au cas par cas ULS n'est pas systématiquement la méthode la plus performante et est souvent en concurrence avec la méthode *Root-to-tip*. De plus, les estimations avec des matrices de distances laissent souvent quelque peu à désirer, en particulier sur les jeux de données intra-hôte avec 3 temps d'échantillonnage et des séquences de 300 paires de bases. Ces jeux de données contiennent, en moyenne, de grandes distances et cela suggère que la performance d'ULS est (logiquement) grandement dépendante de la justesse des estimations des distances évolutives qui sont mieux estimées lorsqu'elles sont petites. L'utilisation d'une autre variance, plus appropriée, pourrait balancer en notre faveur. En attendant, l'utilisation d'arbres FastME, rapides à obtenir, semble corriger ce problème.

La différence en précision d'estimation entre ULS et la méthode probabiliste de référence, BEAST, est en moyenne à notre avantage mais reste généralement non significative sur les jeux de données simulées en inter-hôtes. En revanche, la performance d'ULS contre BEAST est autre sur les jeux de données intra-hôtes contenant un taux de mort élevé (995 sur 1 000 à chaque temps d'échantillonnage). Récemment, une nouvelle version de BEAST est disponible et cette version permet d'utiliser un modèle de spéciation, plus adapté que celui du coalescent (seul modèle disponible avec la version 1.6.2 pour des données hétérochrones), considérant un taux de naissance et de mort constant avec des données hétérochrones (Drummond *et al*, 2012; Stadler, 2010). Une comparaison avec cette nouvelle version est nécessaire et permetterait de confirmer ou d'infirmer les résultats présentés dans ce manuscit.