Université de Montréal

Évolution de familles de gènes par duplications et pertes - Algorithmes pour la correction d'arbres bruités

par
Andrea Doroftei

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des arts et des sciences
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

Décembre, 2011

Université de Montréal
Faculté des arts et des sciences

Ce mémoire intitulé:

Évolution de familles de gènes par duplications et pertes - Algorithmes pour la correction d'arbres bruités

présenté par:

Andrea Doroftei

a été évalué par un jury composé des personnes suivantes:

Sylvie Hamel,	président-rapporteur
Nadia El-Mabrouk,	directeur de recherche
Pierre McKenzie,	membre du jury

Mémoire accepté le: . . . . . .

RÉSUMÉ

Les gènes sont les parties du génome qui codent pour les protéines. Les gènes d'une ou plusieurs espèces peuvent être regroupés en "familles", en fonction de leur similarité de séquence. Cependant, pour connaître les relations fonctionnelles entre ces copies de gènes, la similarité de séquence ne suffit pas. Pour cela, il est important d'étudier l'évolution d'une famille par duplications et pertes afin de pouvoir distinguer entre gènes *orthologues*, des copies ayant évolué par spéciation et susceptibles d'avoir conservé une fonction commune, et gènes *paralogues*, des copies ayant évolué par duplication qui ont probablement développé des nouvelles fonctions.

Étant donnée une famille de gènes présents dans n espèces différentes, un arbre de gènes (obtenu par une méthode phylogénétique classique), et un arbre phylogénétique pour les n espèces, la "*réconciliation*" est l'approche la plus courante permettant d'inférer une histoire d'évolution de cette famille par *duplications*, *spéciations* et *pertes*. Le degré de confiance accordé à l'histoire inférée est directement relié au degré de confiance accordé à l'arbre de gènes lui-même. Il est donc important de disposer d'une méthode préliminaire de correction d'arbres de gènes.

Ce travail introduit une méthodologie permettant de "corriger" un arbre de gènes : supprimer le minimum de feuilles "mal placées" afin d'obtenir un arbre dont les sommets de duplications (inférés par la réconciliation) sont tous des sommets de "duplications apparentes" et obtenir ainsi un arbre de gènes en "accord" avec la phylogénie des espèces. J'introduis un algorithme exact pour des arbres d'une certaine classe, et une heuristique pour le cas général.

Mots clés:Algorithmique, Bio-informatique, Génomique évolutive, Familles de gènes, Duplication, Rèconciliation.

ABSTRACT

Genes are segments of genomes that code for proteins. Genes of one or more species can be grouped into gene families based on their sequence similarity. In order to determine functional relationships among these multiple gene copies of a family, sequence homology is insufficient as no direct information on the evolution of the gene family by duplication, speciation and loss can be inferred directly from a family of homologous genes. And it is precisely this information that allows us to distinguish between *orthologous* gene copies, that have evolved by speciation and are more likely to preserve the same function and *paralogous* gene copies that have evolved by duplication and usually acquire new functions.

For a given gene family contained within n species, a gene tree (inferred by typical phylogenetic methods) and a phylogenetic tree of the considered species, reconciliation between the gene tree and the species tree is the most commonly used approach to infer a duplication, speciation and loss history for the gene family. The main criticism towards reconciliation methods is that the inferred duplication and loss history for a gene family is strongly dependent on the gene tree considered for this family. Indeed, just a few misplaced leaves in the gene tree can lead to a completely different history, possibly with significantly more duplications and losses. It is therefore important to have a preliminary method for "correcting" the gene tree, i.e. removing potentially misplaced branches.

N. El-Mabrouk and C. Chauve introduced *"non-apparent duplications"* as nodes that are likely to result from the misplacement of one leaf in the gene tree. Simply put, such a node indicates that one or more triplets contradict the phylogeny given by the species tree. In this work, the problem of eliminating non-apparent duplications from a given gene tree by a minimum number of leaf removals is considered. Depending on the disposition of this type of nodes in the gene tree, the algorithm introduced leads to an O(n*log*n) performance and an optimal solution in a best case scenario . The general case however is solved using an heuristic method.

Keywords: Algorithmics, Bio-informatics, Evolution Genomics, Gene Family, Duplication, Reconciliation.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES ANNEXES

LISTE DES SIGLES

AD	Apparent Duplication - Duplication Apparente
NAD	Non-Apparent Duplication - Duplication Non-Apparente
LCA	Least Common Ancestor - Ancêtre commun le plus récent

NOTATION

$\mathscr{G} = \{1, 2, 3, \cdots, n\}$	ensemble de n génomes
T	arbre de gènes sur $\mathscr{G}$
S	arbre d'espèces pour $\mathscr{G}$
T_x	sous-arbre de T de racine x
$\mathscr{G}(x)$	sous-ensemble de $\mathscr{G}$ défini par les étiquette des feuilles de T_x
$\|T\|$	taille de T

À la mémoire de mon grand-père, *Popescu Constantin*.

REMERCIEMENTS

Je remercie tous ceux qui m'ont soutenu pour la réalisation de ce travail. Particulièrement, j'aimerais mentionner *Nadia*, ma directrice de recherche, pour son encadrement, sa patience et son support.
Je remercie *Daniela* et *Liviu* pour leur amour, je remercie mon bien aimé *Alexander*, le grand *moi*, pour son amour et son encouragement, je remercie *Mimi* et *Ileana* pour ma belle enfance et non dernièrement je remercie *Anatoli* et *Daniela-Veronica* pour leur amitié.

CHAPITRE 1

INTRODUCTION

La biologie moléculaire est une discipline dédiée à l'étude des molécules messagères du matériel héréditaire (l'ADN - acide désoxyribonucléique, l'ARN - acide ribonucléique et les protéines), de leur structure, de leur synthèse et de leur évolution. La découverte de la structure de l'ADN vient révolutionner l'étude des phénomènes biologiques en introduisant la dimension moléculaire. Le dogme central de la biologie moléculaire décrit l'expression de l'information génétique à partir de son support, l'ADN, jusqu'aux protéines. La bio-informatique constitue l'intersection de la biologie moléculaire et de l'informatique, ayant comme but principal de comprendre les mécanismes de fonctionnement de la cellule au niveau moléculaire et de résoudre les problèmes scientifiques qui en découlent. L'interprétation informatique des concepts biologiques est nécessaire pour le développement des techniques qui permettent de résoudre plusieurs de ces problèmes. Ce mémoire se consacre à l'étude comparative des arbres évolutifs construits à partir des séquences génétiques contenues dans les molécules ADN.

On suppose que les espèces actuelles sont issues d'un ancêtre commun et que leur histoire évolutive peut se traduire par un arbre binaire, appelé arbre d'espèces. Il reflète l'évolution des espèces (les feuilles de l'arbre), à partir du noeud racine - l'ancêtre commun. Les noeuds internes représentent les espèces ancestrales à partir desquelles les espèces à l'étude ont évolué. Cette évolution des génomes est due à des mutations locales affectant les séquences de nucléotides, mais également à des événements plus larges de duplication, perte ou transfert de segments plus ou moins longs, pouvant contenir des gènes. La duplication, par exemple, est le phénomène responsable du fait que les génomes contiennent des gènes présents en plusieurs copies. On désigne par "*gènes homologues*" les gènes contenus dans une ou plusieurs espèces provenant d'une copie ancestrale commune. En pratique, les gènes homologues sont identifiés par similarité de séquence et le traitement de l'information obtenue peut alors être utilisé pour construire un arbre phylogénétique, ou arbre de gènes, retraçant leur évolution.

D'un point de vue fonctionnel, regrouper les gènes par homologie de séquence ne suffit pas à inférer une fonction commune à toutes les copies. Dans ce cadre, il est important de distinguer entre orthologues et paralogues. Les orthologues sont des copies de gènes dont le dernier ancêtre commun a évolué par spéciation (le mécanisme évolutif qui résulte en la spéciation d'une espèce en deux espèces distinctes), alors que les paralogues sont des copies dont le dernier ancêtre commun a évolué par duplication. Ce sont les gènes orthologues qui conservent généralement la même fonction. La duplication de gènes constitue l'événement évolutif principal à l'origine de la création de nouvelles fonctions et par conséquent, dans l'évolution des espèces, particulièrement des eukaryotes [6, 20]. Inversement, les pertes de gènes, découlant généralement de l'accumulation de mutations ponctuelles rendant le gène non-fonctionnel, représentent également un moteur essentiel de l'évolution. La méthode la plus utilisée pour inférer l'histoire évolutive d'une famille de génes par duplication, pertes et spéciation est la *réconciliation* entre l'arbre des gènes et l'arbre des espèces. En considérant les duplications et pertes, la reconciliation, introduite par Goodman [11], consiste à "emboîter" l'arbre des gènes dans l'arbre des espèces". Ma, Li et Zhang [21] introduit une généralisation de ce modèle en considérant, en plus des duplications et pertes, le transfert horizontal de gènes.

C. Chauve et N. El-Mabrouk [3] introduisent la réconciliation comme une extension de l'arbre de gènes obtenue par une série d'insertions de sous-arbres permettant d'obtenir une structure d'arbre qui soit en "accord" avec celle de l'arbre d'espèces. Ceci permet de reconstruire l'histoire évolutive de la famille de gènes par duplications et pertes. Évidemment plusieurs réconciliations sont possibles et une approche naturelle consiste à choisir celle qui optimise un critère donné, soit les duplications, les pertes ou les mutations (duplications et pertes combinées). Le mapping LCA, "couplage" du plus récent ancêtre commun [13, 25], définit une réconciliation qui minimise les duplications et les mutations (duplications+pertes) [4, 21]. Par conséquent elle induit une histoire évolutive qui optimise ces critères. Il s'agit de la méthode la plus utilisée [8, 23, 25, 31].

La réconciliation résout le problème d'incompatibilité évolutive entre l'arbre des espèces et l'arbre de gènes en supposant que ce dernier est correctement inféré par les méthodes phylogénétiques classiques. Mais quelques feuilles mal placées de l'arbre de

gènes peuvent mener à un scénario d'évolution très différent, signalant beaucoup plus de duplications et pertes. Par conséquent, l'évaluation de l'arbre de gènes est une étape préliminaire nécessaire afin d'obtenir un scénario d'évolution digne de confiance. Dans ce mémoire je propose un algorithme polynomial permettant d'évaluer et de corriger l'arbre des gènes. Cet algorithme peut être vu comme une méthode de prétraitement d'un arbre de gènes, précédant la réconciliation.

Le présent travail exploite une notion introduite par C. Chauve et N. El-Mabrouk [3] comme moyen de repérer, dans un arbre des gènes G, des feuilles potentiellement mal placées, soit les *duplications non-apparentes*. Il s'agit de sommets qui contredisent la phylogénie des espèces et qui résultent possiblement d'une erreur dans la construction de l'arbre de gènes. Plus précisément, les sommets de duplication non-apparente contredisent l'arbre des espèces sur la phylogénie d'un ou plusieurs triplets. On introduit ici des méthodes algorithmiques qui utilisent cette propriété de l'arbre de gènes pour effectuer sa correction en temps polynomial, afin d'obtenir un arbre en accord avec l'arbre des espèces. Le problème consiste à trouver le minimum de feuilles à supprimer de l'arbre de gènes afin d'éliminer tous les sommets de duplication non-apparente. La distribution de ces derniers dans l'arbre de gènes, en terme de hiérarchie, conduit dans certains cas à un résultat exact, où le problème peut être réduit à retrouver le plus grand sous-arbre de l'arbre de gènes qui soit en accord avec l'arbre des espèces. Le cas général mène à une résolution approchée qui permet d'éliminer tous les sommets de duplication non-apparente en plusieurs itérations ce qui ne garantit pas l'optimalité. Ce travail a fait l'objet d'un article publié dans les actes de la conférence internationale WABI 2011 [7]. L'article est présenté dans son integralité en annexe de ce mémoire.

CHAPITRE 2

CONTEXTE GÉNÉRAL DE LA RECHERCHE

2.1 Contexte Biologique

2.1.1 La génomique évolutive

Le génome est l'ensemble du matériel génétique qui caractérise une espèce. Lors de la division cellulaire, ce contenu génétique se transmet de la cellule mère à la cellule fille par le biais des séquences d'acide désoxyribonucléique (ADN) organisées en des structures plus complexes appelées *chromosomes*. Ce contenu peut être vu comme l'ensemble des instructions nécessaires au fonctionnement et au développement de tout organisme vivant. La molécule d'ADN est constituée de deux longues chaînes ordonnées - les deux brins de l'ADN - formées par une succession de quatre types de bases azotées, *l'adénine (A)* , la *guanine (G)*, la *cytosine (C)* et la *thymine (T)*, reliées entre elles par une structure de phosphates et sucres. Ces quatre bases peuvent être subdivisées en deux ensembles de bases complémentaires : (A,T) et (C,G). Les deux brins de l'ADN sont complémentaires, dans le sens où l'un des deux brins est obtenu à partir de l'autre en remplaçant chaque base par son complément. Ils sont maintenus en une structure de double-hélice grâce aux liaisons hydrogène qui se forment entre les bases complémentaires.

Le génome d'un organisme est constitué d'une ou de plusieurs doubles hélices d'ADN, chacune appelée *chromosome*. Dans le cas des organismes diploïdes, en particulier l'homme, chaque chromosome est présent en deux copies. Par exemple, le génome humain est formé de 23 paires de chromosomes.

L'ADN contient le code génétique à l'origine de la synthèse des protéines, qui sont des macromolécules remplissant des fonctions très diverses au sein de la cellule. Les protéines sont formées d'une succession d'acides aminés. Il y a 20 acides aminés différents. Le processus de synthèse des protéines consiste en 2 étapes : la transcription de la séquence de nucléotides en une macromolécule intermédiaire appelée l'ARN messager (acide ribonucléique messager), suivie de la traduction de l'ARNm en une suite d'acides

aminés constituant la protéine.

Avant la division cellulaire, une copie de la molécule d'ADN doit être produite afin de transmettre le contenu génétique à la cellule fille. Ce méchanisme, appelé la *réplication de l'ADN*, débute avec la séparation de la double hélice par une enzyme qui brise les liaisons hydrogènes entre les bases complémentaires des deux brins. Par la suite, chaque brin d'ADN est copié en un brin complémentaire pour donner naissance à deux nouvelles molécules constituées chacune de la matrice et du brin dupliqué.

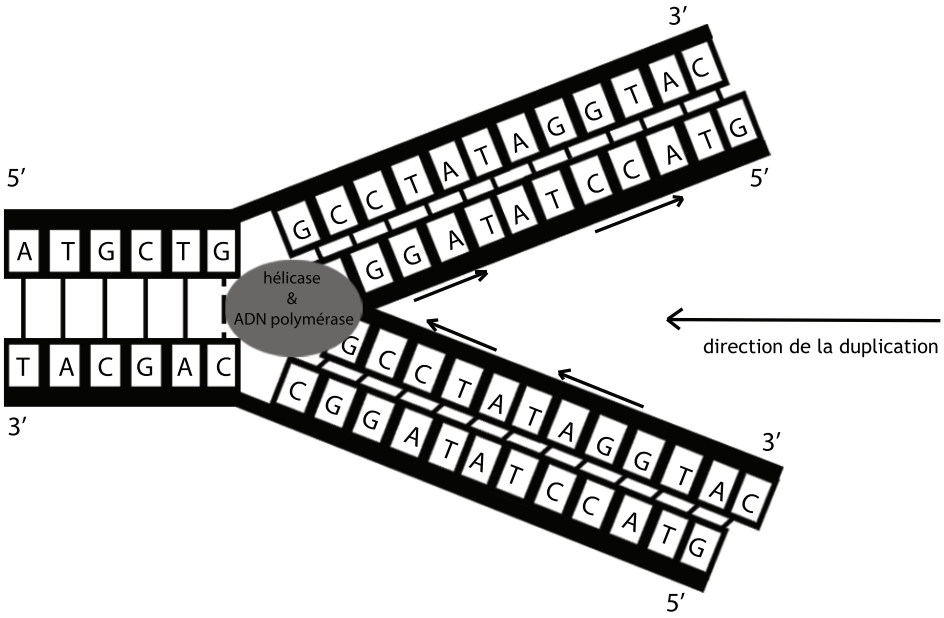

Figure 2.1 – La réplication de l'ADN. Les deux brins complémentaires sont séparées par une enzyme, l'hélicase, qui défait les liaisons hydrogène entre les bases. Par la suite, l'ADN polymérase produit deux copies simultanées des deux brins. Chaque brin dupliqué est complémentaire à sa matrice et formera avec celle-ci une nouvelle molécule d'ADN.

La réplication de l'ADN est un processus presque parfait. Des erreurs peuvent se produire lorsque l'ADN est dupliqué et des nucléotides peuvent être *insérés*, *supprimés* ou *substitués*. Une telle modification de la séquences nucléotidique est appelée *mutation*. Il s'agit du méchanisme de l'évolution qui est à la base de la variabilité des populations. Selon leur effet sur l'organisme qui les porte, certaines mutations peuvent être néfastes alors que d'autres sont responsables de l'innovation génétique. Par conséquent, ce mé-

chanisme joue un rôle essentiel dans la diversification et l'évolution des organismes vivants. Le point de divergence entre deux espèces peut être retracé dans le temps grâce au nombre de mutations qui les séparent.

La *génomique évolutive* étudie les séquences génétiques (suite de nucléotides qui forment les brins d'ADN) des espèces afin de faire ressortir les informations nécessaires à la reconstitution de leur cheminement évolutif. Les études phylogénétiques, qui sont des études sur l'évolution des espèces, se basent généralement, non pas sur la comparaison de génomes entiers, mais sur la comparaison ou l'alignement de gènes particuliers dont des copies (homologues) ont été identifiées dans chacune des espèces étudiées.Une *phylogénie*, ou un arbre d'évolution est un modèle mathématique utilisé pour expliquer les relations historiques entre un groupe d'organismes.Un arbre est constitué de noeuds reliés entre eux par des branches. Les feuilles de l'arbre représentent les espèces actuelles alors que les noeuds internes sont des ancêtres hypothétiques à partir desquels ces espèces ont évolué. Un des objectifs principaux de la génomique évolutive est de reconstruire l'arbre de la vie où la racine représente l'ancêtre commun de toute espèces connue, donc l'origine de la vie. L'arbre de la Figure 2.2 est un exemple de phylogénie selon laquelle toutes les espèces sont issues d'une seule espèce ancestrale représentant les trois domaines de la vie : les eucaryotes, les bactéries et les arché-bactéries.

2.1.2 Les familles de gènes

L'ADN n'est pas codant sur toute sa longueur. Dans le cas du génome humain par exemple, à peine 2 % du génome est transcrit. Les parties de l'ADN qui codent pour les protéines sont appelées des gènes. Un gène est donc la matrice à l'origine de la synthèse d'une (ou de plusieurs) protéines. L'étude des séquences génétiques montre que beaucoup de gènes sont présents à l'intérieur du génome non pas en une seule copie, mais en copies multiples, ce qui permet la synthèse d'une protéine par plusieurs gènes à la fois. Par exemple dans le génome humain, 15% de tous les gènes codant pour des protéines sont présents en plusieurs copies [19].

Le processus évolutif qui permet de générer de nombreuses copies d'un même gène est la duplication.

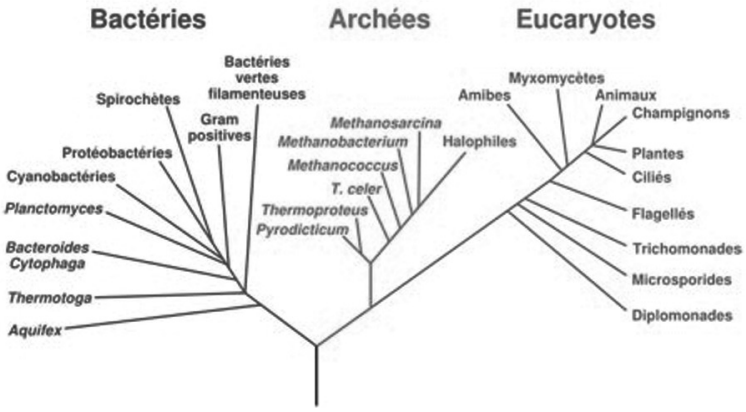

Figure 2.2 – L'arbre phylogénétique reflétant la subdivion en 3 domaines du vivant (http ://www.nasa.gov).

Ce mécanisme peut avoir diverses origines. En particulier, il peut être dû à des erreurs de recombinaisons lors de la méiose (recombinaisons inégales), ce qui donne lieu à des duplications en tandem, i.e. des copies multiples adjacentes sur le chromosome. Un autre mécanisme donnant lieu à des duplications non adjacentes est la rétrotransposition, où l'ARN messager est inversement transcrit en ADN complémentaire puis inséré dans le génome. La présence de copies dupliquées peut également être due à des mécanismes de plus grande envergure entrainant la duplication d'un chromosome, ou même du génome entier.

La duplication de gènes joue un rôle primordial dans l'évolution des espèces. En effet, c'est une source importante d'innovation génétique, et de création de nouvelles fonctions. Immédiatement après la duplication, les deux copies du gène ont exactement la même séquence. Cependant, au cours de l'évolution, elles accumulent des mutations, pouvant entraîner la perte de fonction de l'une des deux copies. Un tel gène est appelé "pseudogène". Lorsque trop de mutations se sont accumulées, le pseudogène n'est plus reconnaissable. On parle alors de la "perte" du gène.

On appelle *famille de gènes* un ensemble de gènes dans un ou plusieurs génomes ayant évolué à partir d'un ancêtre commun. Les familles de gènes sont généralement

identifiées par homologie de séquence. Par exemple, en utilisant la méthode de recherche BLAST, tous les gènes ayant un score de similarité supérieur à un certain seuil sont considérés "homologues" et regroupés dans une même famille.

Il est important de distinguer entre deux types de gènes homologues : les *orthologues* et les *paralogues*. Deux gènes homologues dans deux génomes différents sont dits orthologues s'ils sont issus de leur dernier ancêtre commun par spéciation. D'autre part, deux gènes qui sont homologues dans le même génome ou dans deux génomes différents sont dits paralogues s'ils sont issus de leur dernier ancêtre commun par duplication. Alors que les gènes orthologues conservent généralement la même fonction, les gènes paralogues peuvent en développer des nouvelles. En effet la présence de deux copies dans un même génome permet à l'une des deux copies d'évoluer et d'acquérir éventuellement une nouvelle fonction, sans affecter la fonction initiale, assurée par l'autre copie.

La figure 2.3 illustre l'évolution d'une famille de gènes dans trois génomes A, B et C, ayant évolué selon la phylogénie $((A,B),C)$ (Figure 2.3. (a)). Le gène g du génome ancestral G de A, B et C a subi une duplication précédant la première spéciation de la phylogénie. Deux descendants de chacune des deux copies créées x et y sont présents dans chacune des espèces actuelles A, B et C. Les six gènes actuels $1,2,3,4,5,6$ sont tous homologues, et forment donc une seule famille de gènes. Dans cette famille, tous les gènes de la sous-famille $F_1 = \{1,2,3\}$ descendant de x sont paralogues aux gènes de la sous-famille $F_2 = \{4,5,6\}$ descendant de y. Les gènes de chacune des sous-familles sont orthologues entre eux.

En se référant à la figure 2.3, si on étudiait seulement les gènes $1,3$ et 5, la phylogénie déduite serait $((A,C),B)$ au lieu de $((A,B),C)$ et ce malgré le fait que les espèces A et B sont plus proches l'une de l'autre que A et C. La duplication des gènes fait en sorte que des copies distinctes du même gène peuvent nous induire en erreur quant à la vraie phylogénie des espèces. Ainsi, pour représenter une histoire évolutive correcte il serait important d'avoir l'ensemble complet des gènes d'une famille. Une alternative plus réaliste est de sélectionner uniquement un ensemble de gènes orthologues. D'un point de vue pratique, il n'existe pas de méthode simple et universelle (homologie de séquence ou autre) permettant de distinguer entre gènes orthologues et gènes paralogues.

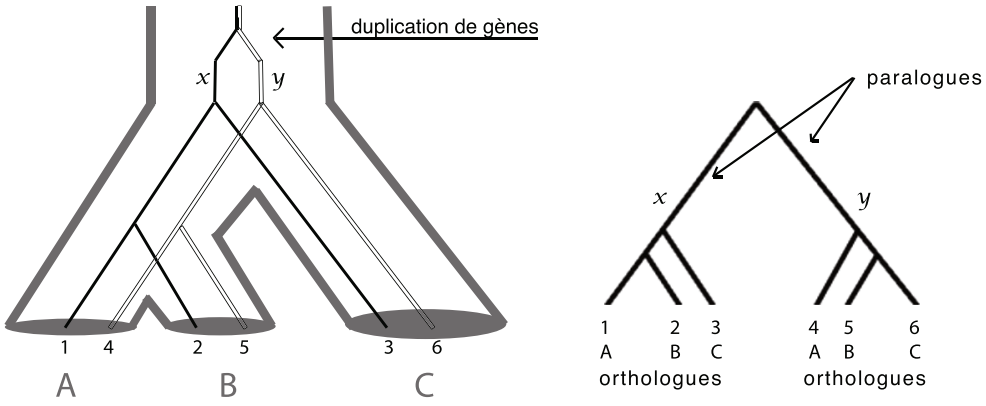

Figure 2.3 – Exemple de duplication de gènes qui donne lieu à deux lignées de gènes paralogues : le groupe des gènes $F_1 = \{1,2,3\}$ descendants de x et le groupe de gènes $F_2 = \{4,5,6\}$ descendants de y. Les gènes de chacune des familles F_1 et F_2 sont orthologues entre eux. Afin de retrouver la bonne phylogénie pour les espèces étudiées A, B et C, les copies de gènes sélectionnés doivent être des orthologues.

Une famille de gènes identifiée par homologie de séquence est généralement incomplète (gènes manquants) et contient à la fois des orthologues et des paralogues.

2.2 Contexte bioinformatique

2.2.1 Inférence de famille de gènes

Les familles multigéniques sont constituées de gènes homologues, issus d'un ancêtre commun par une succession de spéciations et de duplications. La similarité de séquence est le critère principal utilisé pour identifier les membres d'une famille de gènes. Le recherche de séquences homologues et le regroupement des gènes respectifs en famille est une étape fondamentale en génomique comparative car elle permet d'étudier les relations de parenté au sein de cette famille. Parmi les méthodes de recherche d'homologie de séquence, BLAST ("Basic Local Alignement Tool ") est la plue connue. C'est une méthode heuristique permettant de rechercher toutes les occurrences d'une séquence cible dans une séquence requête. BLAST est suffisamment rapide pour être exécuté sur une

séquence requête aussi grosse que la banque de données génomique GenBank. Un autre avantage de BLAST est le fait qu'un score statistique est attribué à chaque occurrence retrouvée : plus la P-value est faible, plus l'occurrence est significative. Afin de retrouver toutes les copies homologues à un certain gène g dans un ensemble de génomes, une méthode directe est donc d'exécuter BLAST avec la séquence cible g et successivement chacun des génomes.

Afin de ne garder que les homologies les plus significatives, une stratégie "Bi-directional Best BLAST hit" est souvent considérée. Considérons deux génomes A et B. Soit g_A un gène dans A et g_B un gène dans B. L'homologie (g_A, g_B) n'est alors considérée que si g_B est la meilleur occurrence trouvée dans B pour une recherche BLAST avec la cible g_A, et inversement, la meilleur occurrence trouvée dans A pour une recherche BLAST avec la cible g_B. Plusieurs algorithmes utilisant cette technique de base ont été développés afin de regrouper les gènes en familles, en particulier l'algorithme qui est à la base de la banque de données COG [28, 29]. Notons également les algorithmes INPARANOID [22] et OrthoMCL [18].

2.2.2 Inférence phylogénetique

Les séquences d'ADN recèlent de traces de l'évolution très précieuses, pouvant être exploitées pour retracer l'histoire des espèces.

Le postulat à la base des études phylogénétiques stipule que tous les êtres vivants descendent d'un ancêtre commun, et que tout au long de l'évolution les gènes accumulent des mutations. Lorsque celles-ci sont bénéfiques à l'organisme, elles sont fixées et transmises d'une génération à l'autre. De plus, l'isolement d'une population et l'adaptation à son environnement peut entraîner l'accumulation de suffisamment de mutations entrainant la création d'une nouvelle espèce : c'est la spéciation. Il découle de ce postulat que les séquences d'ADN peuvent être utilisées pour retracer l'histoire de spéciation ayant donné lieu aux espèces actuelles. Les comparaisons de séquences ne portent généralement pas sur les génomes complets, mais plutôt sur des séquences homologues de gènes, identifiées dans les génomes étudiés.

Le premier objectif des études phylogénétiques consiste donc à reconstruire l'arbre

de vie de toutes les espèces vivantes à partir de l'étude de leurs séquences d'ADN. Cependant, elles permettent également des études plus ciblées sur des familles particulières de gènes jouant des rôles spécifiques, ou impliqués dans des maladies particulières. En effet, une multitude de questions biologiques ne peuvent être abordées correctement sans une vision évolutive.

La phylogénie est un domaine classique de la bio-informatique. Les méthodes de construction d'arbres phylogénetiques développées peuvent être regroupées en trois catégories principales : les méthodes de distances, les méthodes de parsimonie, et les méthodes probabilistes.

Pour ce qui est des méthodes de distances, elles consistent à calculer les distances deux à deux des séquences homologues sélectionnées, et de construire l'arbre qui reflète le mieux la matrice de distance obtenue. Dans le cas des méthodes de parcimonie, toutes les topologies d'arbres sont considérées, pour chacune des topologies, un score est attribué reflétant le nombre de mutations minimales impliquées par un tel arbre, et un arbre de score minimal est sélectionné. Finalement, dans le cas des méthodes probabilistes, des probabilités de mutations sont calculées en fonction d'un modèle d'évolution spécifié à l'avance.

2.3 Inférence d'histoire évolutive

2.3.1 Incongruité entre un arbre de gènes et un arbre d'espèces

L'un des problèmes majeurs des méthodes phylogénétiques réside dans le fait que, étant donné un ensemble d'espèces $\mathscr{G}$, deux familles de gènes différentes peuvent donner lieu à des histoires évolutives différentes. Une branche des recherches phylogénétiques consiste à rechercher l'arbre "consensus" représentant le mieux l'histoire des espèces. D'autre part, supposons que les relations évolutives entre les espèces $\mathscr{G}$ soient connues (confirmées par différents résultats convergeants) et reflétées par un arbre phylogénétique S. Considérons alors une famille de gènes homologues dans les génomes $\mathscr{G}$, et considérons un arbre de gènes G pour ses gènes, obtenu en appliquant une méthode phylogénétique donnée. Il est très fréquent que l'arbre G ne soit pas isomorphe à S. En plus

des arbres de gènes, on construit des arbres qui représentent l'évolution des espèces et leurs relations de parenté. Cependant, il arrive très rarement qu'un arbre qui représente l'évolution des espèces est isomorphe à la phylogénie d'un groupe de gènes homologues issus des mêmes espèces. Si on considère que les deux phylogénies sont correctes, l'incongruence entre un arbre de gènes et un arbre d'espèces peut s'expliquer par l'évolution des gènes par duplication et pertes. À titre d'exemple dans la figure 2.4, une hypothèse plausible pour expliquer l'incongruence entre l'arbre de gènes G et la phylogénie des espèces S est une histoire qui implique une duplication du gène ancestral x résultant en la formation de deux groupes de gènes paralogues entre eux, suivie par la perte de trois gènes nécessaire pour expliquer l'arbre de gènes observé.

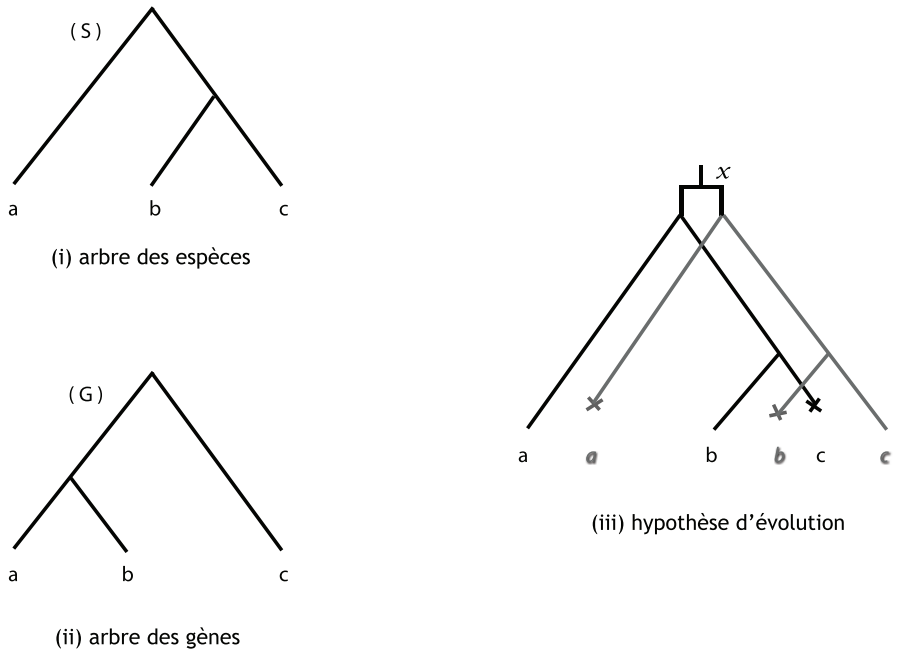

Figure 2.4 – Exemple de contradiction entre l'arbre des espèces (S) et l'arbre des gènes (G). Une hypothèse plausible est l'évolution du gène x par duplication, la création de deux groupes paralogues entre eux et la pertes de trois gènes parmi ces deux groupes.

Prenons un exemple plus concret. Chez les vertébrés on retrouve deux types de gènes appartenant à la famille de l'hémoglobine : l'α-hémoglobine et la β-hémoglobine. Ces

deux branches ont évolué à partir d'un type ancestral d'hémoglobine qui a été dupliqué avant l'apparition des vertébrés. La Figure 2.5 représente un arbre pour cette famille de gènes, pour une sous-famille de vertébrés, l'humain, le chien et le cheval.

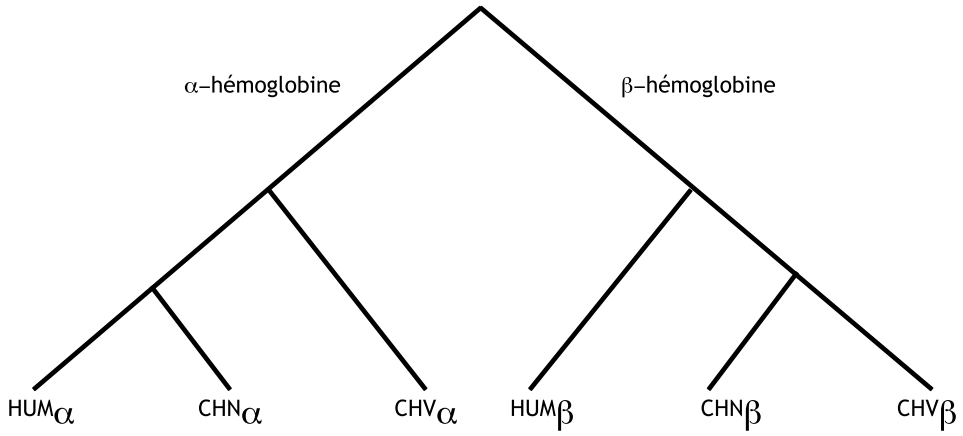

Figure 2.5 – Arbre qui représente l'évolution de l'hémoglobine par duplication donnant lieu à deux lignées : l'α et la β-hémoglobine. La duplication a lieu avant l'apparition des espèces vertebrés, dans ce cas l'humain, le chien et le cheval. L'α et la β-hémoglobine ont une grande similarité de séquence, mais chez les mammifères la similarité entre les gènes α (idem pour β) entre eux est plus grande que la similarité entre les deux lignées.

Lorsqu'on veut inférer une phylogénie pour la famille de l'hémoglobine le choix d'un groupe de gènes paralogues, soit par exemple les gènes β-chien paralogues à l'α-humain et l'α-cheval, mène vers une incongruité avec la phylogénie de ces espèces. En effet, les gènes paralogues, ayant évolué par duplication, exhibent une phylogénie qui ne reflète pas l'histoire évolutive des espèces et la plupart du temps il est difficile de connaître avec précision les relations de paralogie et d'orthologie pour un groupe de gènes homologues. Une famille de gènes peut être incomplète, dû à des pertes ou à des erreurs d'identification des gènes lors de l'annotation du génome, ce qui peut également mener à une incompatibilité des deux phylogénies.

La Figure 2.6(a) montre l'arbre de gènes pour le β-chien, l'α-humain et l'α-cheval.

Étant donné les liens de paralogie, l'arbre résultant inféré par similarité de séquence montre une fausse phylogénie, par rapport à la vraie 2.6(b) qui est représentée par exemple par la lignée α, des gènes orthologues entre-eux. Le désaccord est dû au fait que les gènes de la lignée α sont plus semblables entre eux que les α et β entre eux. La réconciliation (2.6(c)) identifie une duplication et 3 pertes (β cheval et humain, et α chien) qui expliquent la différence topologique entre l'arbre des gènes et celui des espèces,

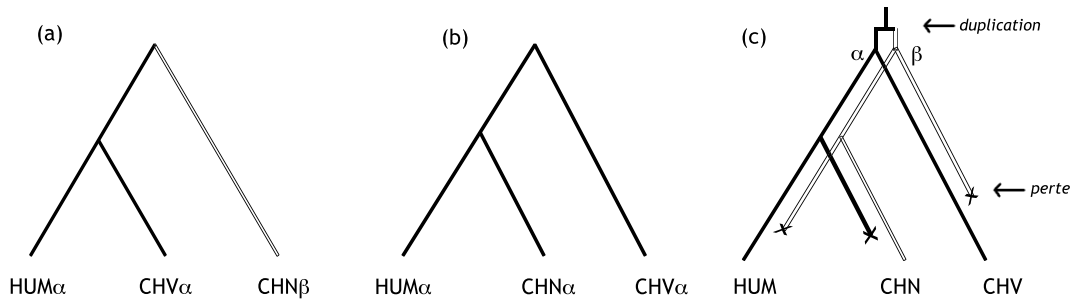

Figure 2.6 – L'arbre de gènes pour l'hémoglobine contredit la vraie phylogénie de cette famille de gènes lorsqu'on choisit des copies paralogues pour l'α-humain,l'α-chien et le β-cheval : l'α et β-hémoglobine.

2.3.2 Arbres consensus

On dit que deux arbres sont "congruents" si leurs topologies respectives sont superposables. Si on a un arbre de gènes et un arbre d'espèces qui sont incongruents, une façon d'adresser ce problème est de retrouver *l'arbre consensus*, c'est à dire une topologie commune pour les deux arbres. La réconciliation est une méthode qui permet d'inférer un arbre qui respecte les deux phylogénies proposées. Au cours de l'évolution les gènes ont divergé de leur topologie initiale par des duplications et des pertes et l'arbre

de gènes observé peut être vu comme le résultat de cette évolution sur l'arbre initial. La figure 2.7 montre un exemple de réconciliation pour l'arbre de gènes et l'arbre d'espèces de la figure 2.4. Un arbre réconcilié nous permet précisément de reconstruire une histoire d'évolution des gènes qui respecte la phylogénie des espèces. Pour les deux arbres de la figure 2.4, on voit qu'afin d'expliquer la topologie de l'arbre des gènes, la réconciliation nous propose une histoire d'évolution des gènes qui comprend une duplication et trois pertes. Le nombre total d'événements inférés par l'arbre réconcilié est le "coût" de la réconciliation. Le concept de réconciliation est plus amplement expliqué au Chapitre 2.

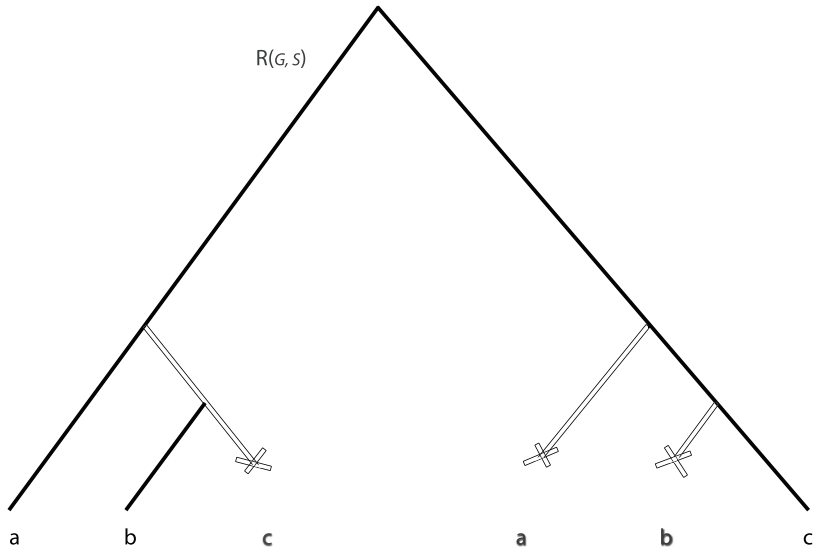

Figure 2.7 – $R(G,S)$ est un arbre reconcilié possible de l'arbre des espèces S et l'arbre des gènes G de la figure (2.4). Le coût de cette réconciliation est *une* duplication et *trois* pertes.

2.4 Correction d'arbres de gènes

Quelle que soit la méthode d'inférence de phylogénie utilisée, de mauvaises constructions peuvent être obtenues. Ceci a des répercussion immédiates sur la cohérence de

l'histoire évolutive proposée par une réconciliation entre un arbre d'espèces et un arbre des gènes. En d'autres termes, pour avoir confiance dans les résultats d'une méthode de réconciliation, il faut avoir confiance en les arbres de gènes et d'espèces considérés. En effet, comme nous le verrons dans le Chapitre 3, une branche mal placée dans l'arbre des gènes peut entraîner une histoire de réconciliation complètement différente.

Les erreurs dans l'inférence des phylogénies peuvent être dues à un mauvais choix de méthode de construction d'arbre ou bien à un mauvais échantillonnage. Par exemple, quant on choisit d'effectuer l'étude d'une phylogénie moléculaire, le taux d'évolution de la molécule choisie doit être adéquat pour le taux d'évolution des espèces. Plusieurs méthodes existent pour évaluer une phylogénie. Une façon de faire est de prendre plusieurs échantillons de la population à l'étude et d'évaluer la différence entre les estimations obtenues. La fréquence d'une estimation parmi celles obtenues permet de mesurer son degré de confiance. Pour une phylogénie, il est difficile d'effectuer un grand nombre d'échantillonnages. On utilise plutôt la technique de "bootstrapping", c'est à dire qu'à partir de l'échantillonnage de base on génère plusieurs fois des estimations en utilisant la même méthode. On évalue la fréquence d'une estimation parmi celles générées afin d'évaluer sa pertinence. On assigne une valeur aux branches d'une phylogénie en fonction du nombre de fois qu'on retrouve cette même branche dans les estimations obtenues [17] .

Étant donné un arbre avec des valeurs de bootstrapping pour chacune de ses branches, la question est de savoir comment gérer les branches faiblement supportées (valeurs de bootstrapping faibles). Une première approche, considérée dans [4], consiste à effectuer des permutations entre des branches de l'arbre, et de mesurer le coût de réconciliation obtenu pour chacun des arbres. Les mouvements autorisés sont désignés sous le nom de "Nearest Neighbour Interchange"). L'arbre choisi est celui qui implique le coût de réconciliation le plus bas.

Une autre approche développée par Chang et Eulenstein [2] consiste à transformer l'arbre binaire étudié en arbre non-binaire en effaçant les branches de faible support et en regroupant les deux noeuds incidents en un seul. Ils proposent un algorithme polynomial pour réconcilier cet arbre non-binaire avec l'arbre des espèces.

Dans ce mémoire, je propose une autre façon de "corriger" un arbre de gènes, qui ne repose pas sur les valeurs de confiance associées aux branches par la technique de "bootstraping". Une nouvelle façon d'identifier les noeuds de faible confiance est l'analyse des "duplications non-apparentes", introduites dans [3] et plus amplement expliquées au Chapitre 2. Ces noeuds peuvent être considérés comme étant le résultat d'une feuille mal placée dans l'arbre. Par des évaluations consécutives des phylogénies obtenues en enlevant ces branches potentiellement mal placées de la phylogénie étudiée, on peut estimer une réconciliation plus parcimonieuse et par conséquent une histoire évolutive plus plausible.

CHAPITRE 3

CONTEXTE FORMEL DE LA RECHERCHE

3.1 Préliminaires

3.1.1 Arbres

Un des postulats fondamentaux à l'origine de la génomique évolutive est que tous les organismes vivants ont un lien commun de parenté. Il en découle que l'évolution peut être représentée par un arbre (phylogénie), qui permet d'exprimer l'information contenue dans les séquences moléculaires sous forme de hiérarchie d'évolution. Les arbres peuvent être enracinés et dans ce cas il y a une direction d'évolution associée à l'arbre ainsi que des relations de descendance, ou bien non enracinés et la direction de l'évolution ancêtre - descendant n'est pas spécifiée. Les sommets terminaux ou les feuilles de l'arbre représentent les espèces actuelles ou pour lesquelles on possède des données biologiques, tandis que les sommets internes représentent les ancêtres hypothétiques à partir desquels ces derniers ont pu évoluer. Dans le cas d'un arbre enraciné, la racine est l'ancêtre commun de toutes les espèces représentées dans l'arbre. Dans tout ce qui suit, nous ne considérons que des arbres d'évolution enracinés.

Formellement, soit $\mathscr{G} = \{1, 2, \cdots, g\}$ un ensemble de g espèces. Un ***arbre ou une phylogénie d'espèces*** S sur $\mathscr{G}$ est un arbre binaire enraciné représentant l'histoire évolutive des espèces : les feuilles représentent les espèces actuelles (étiquetées par des éléments de G), et les sommets internes représentent leur évolution par spéciations. Il contient exactement g feuilles et pour chaque $i \in \mathscr{G}$, l'étiquette i se retrouve une seule fois. Un ***arbre de gènes*** T est un arbre binaire enraciné, tel que chaque feuille est étiquetée par un élément de G. Une feuille étiquetée i représente une copie de gène présente dans le génome i. Une étiquette donnée peut apparaître plusieurs fois dans T. La figure 3.1. est une illustration des notations introduites.

Soit T un arbre. On note par $|T|$ ***la taille de*** T, c'est à dire le nombre de feuilles de T. On note également $\mathscr{G}(\mathrm{T})$ ***l'ensemble de génomes de*** T, qui est le sous-ensemble de

$\mathscr{G}$ défini par les étiquettes des feuilles de T.

Soit x un sommet de T. Alors T_x ***est le sous-arbre de*** T de racine x et l'ensemble de génomes de x, noté par $\mathscr{G}(x)$, est le sous-ensemble de $\mathscr{G}$ défini par les étiquettes des feuilles de T_x . Si x n'est pas une feuille, on note x_g et x_d les deux descendants de x(le fils gauche et le fils droit) . Si x n'est pas la racine de T, on appelle ancêtre de x un sommet y quelconque qui se trouve sur le chemin de x à la racine.

Une ***suppression de feuille*** de T consiste à enlever une feuille f de T, ainsi que le sommet de degré 2 qui r ésulte de cette suppression (le sommet parent de f). On dit qu'un arbre T' est inclus dans T s'il est obtenu à partir de T par une suite de suppressions de feuilles. On dit qu'un sous-arbre T_x de T est un ***sous-arbre maximal*** vérifiant une propriété P ssi T_x vérifie la propriété P et que pour tout sommet y ancêtre de x, T_y ne vérifie pas P.

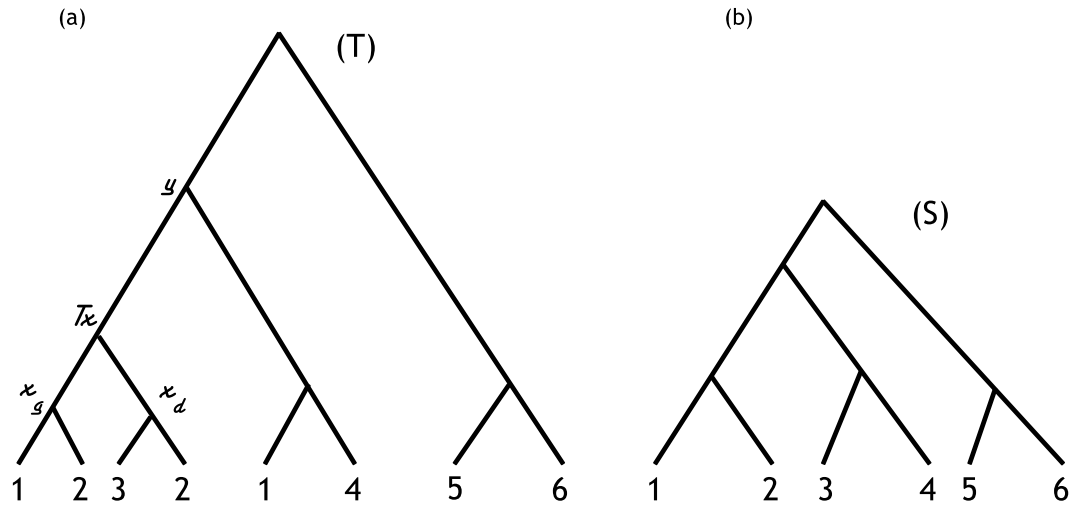

Figure 3.1 – Un arbre de gènes T et un arbre d'espèces S, pour l'ensemble de génomes $\mathscr{G} = \{1,2,3,4,5,6\}$. $|T| = 8$ et $|S| = 6$. T_x est un arbre de racine x et $\mathscr{G}(T_x) = \{1,2,3\}$. T_x est un sous-arbre de T_y. Il est obtenu en supprimant 2 feuilles de T_y : 1 et 4. Tx_g et Tx_d sont les sous-arbres gauche et droit de T_x. T_y est le sous-arbre maximal de T sur l'ensemble de génomes $\mathscr{G} = \{1,2,3,4\}$.

3.1.2 Réconciliation

3.1.2.1 Définition

Lorsqu'une phylogénie est construite pour une famille de gènes homologues en utilisant une des méthodes décrites au Chapitre 2, Section 2.2, le critère principal considéré pour inférer la topologie de l'arbre est la similarité de séquence. Pour inférer une phylogénie des espèces, on étudie plusieurs familles de gènes orthologues, ce qui offre une plus grande confiance dans la topologie obtenue.

En considérant que les deux topologies, celle des gènes et celle des espèces sont correctement représentées, on observe généralement des contradictions au niveau de un ou plusieurs triplets. Lorsqu'un gène est dupliqué, on en retrouve plusieurs copies à l'intérieur du génome d'une espèce et par la suite, chacune des copies peut évoluer de différentes façons. Par conséquent, une espèce peut contenir une ou plusieurs copies du gène ancestral. En plus des duplications, les gènes sont aussi soumis aux pertes et aux transferts horizontaux au cours de leur évolution. Naturellement, lorsqu'on infère une phylogénie pour une famille de gènes homologues, leur façon d'évoluer par des mécanismes autres que les spéciations se reflète directement dans la topologie obtenue. Très souvent, il en résulte une incompatibilité entre cette dernière et l'arbre évolutif des espèces. De plus, il paraît évident que tout scénario évolutif plausible doit prendre en compte l'évolution des gènes par duplications et pertes. Étant donneé cette problématique, les méthodes de construction d'arbres qui reposent sur la similarité de séquence ne sont pas suffisantes pour inférer une histoire évolutive qui exprime directement les différents mécanismes d'évolution des gènes.

Pour expliquer les disparités entre l'arbre des espèces et celui des gènes, on introduit le concept de *réconciliation*. Afin de donner un sens aux différences d'évolution entre la phylogénie des mammifères et celle des hémoglobines appartenant à leur génomes, Goodman [11] définit la réconciliation implicitement comme *l'emboitement de l'arbre des gènes dans l'arbre des espèces*. Il introduit un modèle basé sur l'évolution des gènes par duplication et pertes que Page [23] a développé davantage en introduisant les arbres réconciliés et la fonction de mapping entre les deux phylogénies incongruentes. Ce mo-

dèle basé sur les duplications et les pertes, a été adopté et élaborée par Guigó, Muchnik et Smith [13], Eulenstein [8], Zhang [31], Ma et al. [21], Gorecki [12], Chauve et El-Mabrouk [3]. Un arbre réconcilié permet d'observer l'histoire évolutive de la famille de gènes par duplications et pertes en accord avec l'histoire des spéciations des génomes qui les contiennent. Il est le résultat d'une fonction de couplage entre l'arbre de gènes et celui des espèces qui projette les sommets représentant les gènes vers leurs espèces ancestrales.

Gorecki [12], ainsi que Chauve et El-Mabrouk [3] définissent la réconciliation en terme d'insertions successives de sous-arbres. Avant d'aborder cette définition, introduisons quelques concepts préliminaires :

- Une ***insertion de sous-arbre*** dans un arbre T consiste à insérer un sous-arbre sur une branche de T. Dans la figure 3.2(b), l'arbre T' est obtenu a partir de T par une insertion du sous-arbre $(3,6)$ sur la branche β de ce dernier.

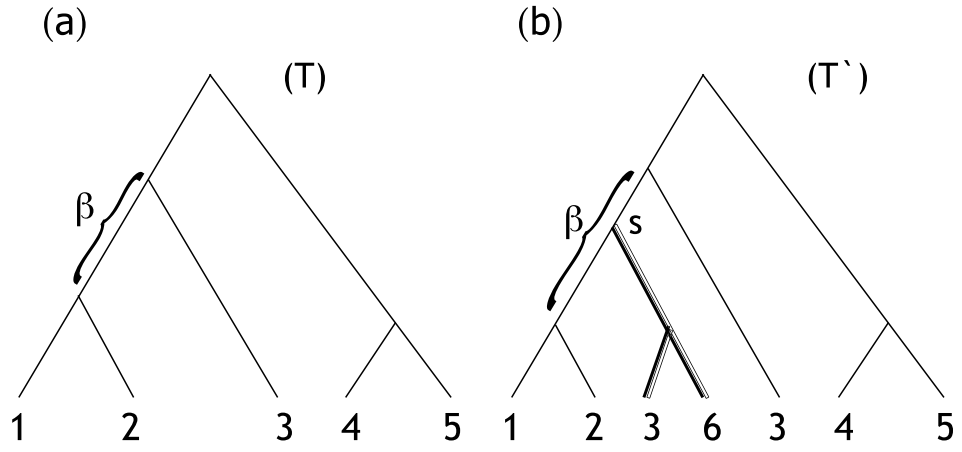

Figure 3.2 – Insertion de sous-arbre. On insère le sous-arbre de racine s sur la branche β de l'arbre T. L'arbre résultant de cette insertion est T'.

- Un arbre T' est appelé une ***extension de*** T si T est un sous-abre de T' et si T' est obtenu à partir de T par des insertions successives de sous-arbres. Dans la figure 3.2, T' est une extension de T.

- Un arbre de gènes T est dit ***DS-consistant avec un arbre d'espèces*** S si T reflète une histoire évolutive sans pertes, donc une histoire d'évolution par des duplications (D) et des spéciations (S). Formellement, T est ***DS-consistant avec*** S si pour chaque sommet t de T tel que $|\mathscr{G}(t)| \geq 2$ il existe un sommet s de S tel que $\mathscr{G}(t) = \mathscr{G}(s)$ et une des conditions suivantes est vérifiées :

  (D) soit $\mathscr{G}(t_g) = \mathscr{G}(t_d)$, indiquant un sommet de duplication,

  (S) ou bien $\mathscr{G}(t_d) = \mathscr{G}(s_d)$ et $\mathscr{G}(t_g) = \mathscr{G}(s_g)$, indiquant sommet de spéciation.

  Définition 1 : Une **réconciliation** entre un arbre de gènes T et un arbre d'espèces S est une ***extension R(T,S) de*** T qui est DS-consistante avec S.

Un exemple de réconciliation est illustré à la Figure 3.3.(c) pour l'arbre de gènes T de la Figure 3.3.(b) et l'arbre d'espèces S de la Figure 3.3.(a). La réconciliation $R(T,S)$ 3.3.(c) représente un scénario d'évolution pour la famille de gènes. La réconciliation est obtenue par une série d'insertions de sous-arbres, representées dans cet exemple par les lignes pointillées.

3.1.2.2 Critère d'optimisation

D'un point de vue théorique, il existe un nombre illimité de réconciliations pouvant expliquer l'incompatibilité d'une phylogénie de gènes par rapport à la phylogénie des espèces. En effet, une multitude d'insertions de sous-arbres peuvent être effectuées afin d'obtenir différents scénarios d'évolution qui peuvent expliquer les différences d'évolution entre une famille de gènes homologues et les espèces respectives. La définition 1 propose un modèle qui ne contraint pas le nombre de pertes et par conséquent, il ne contraint pas le nombre de scénarios d'évolution.

La plupart des méthodes pour la reconstruction d'histoire évolutive adhèrent au principe de *parcimonie* et par conséquent choisissent un scénario qui minimise le nombre

d'événements nécessaire pour expliquer la divergence entre les deux phylogénies. Plusieurs critères d'optimisation ont été considérés : le nombre de duplications (coût en duplications - "duplication cost"), le nombre de pertes (coût en pertes - "loss cost") ou les deux (coût en mutations - "mutation cost"). Page [23] introduit le concept *d'arbre réconcilié* et propose une définition équivalente à celle de la section précédente, soit une réconciliation qui minimise le nombre de feuilles dans l'arbre inféré pour représenter le scénario d'évolution. Conséquemment on minimise aussi le nombre de duplications nécessaires pour arriver à une congruence des deux arbres. Eulensien, Mirkin et Vingron [13] et Page [25] proposent un algorithme de réconciliation basé sur le couplage LCA ("least common ancestor", abrégé LCA) , notion approfondie dans la section suivante. Cet algorithme est élaboré et utilisé dans [1, 8, 12, 21, 24, 30, 31]

3.1.2.3 Le mapping LCA

Étant donné un arbre de gènes T et un arbre d'espèces S, le mapping LCA de T vers S, noté M, relie chaque sommet t de T au plus récent ancêtre commun de $\mathscr{G}(t)$ dans S. La fonction de couplage LCA relie un gène à la plus récente espèce susceptible de le contenir. Lorsqu'un sommet x et au moins un de ses enfants, noté $c(x)$, sont tous les deux reliés à la même espèce ancestrale, on conclut que cette dernière contenait au moins deux copies du gène x. Par conséquent x est un sommet de duplication. Plus précisément chaque sommet t de T est dit ***sommet de duplication*** de T par rapport à S si et seulement si $M(t_\ell) = M(t)$ et/ou $M(t_r) = M(t)$. Inversement, un sommet n est dit ***sommet de spéciation*** si ses deux descendants immédiats sont reliés a des ancêtres distincts dans S.

Le mapping LCA entre T et S induit une réconciliation R(T,S) obtenue par une extension de T, telle que un sommet de duplication de T correspond à un sommet de duplication de R(T,S), et un sommet de spéciation de T correspond à un sommet de spéciation de R(T,S).

L'arbre de la Figure 3.3.(c) est une réconciliation par mapping LCA entre l'arbre de gènes T de la Figure 3.3.(b) et l'arbre d'espèces S de la Figure 3.3.(a). La réconciliation $R(T,S)$ 3.3.(c) represente un scénario d'évolution pour la famille de gènes.

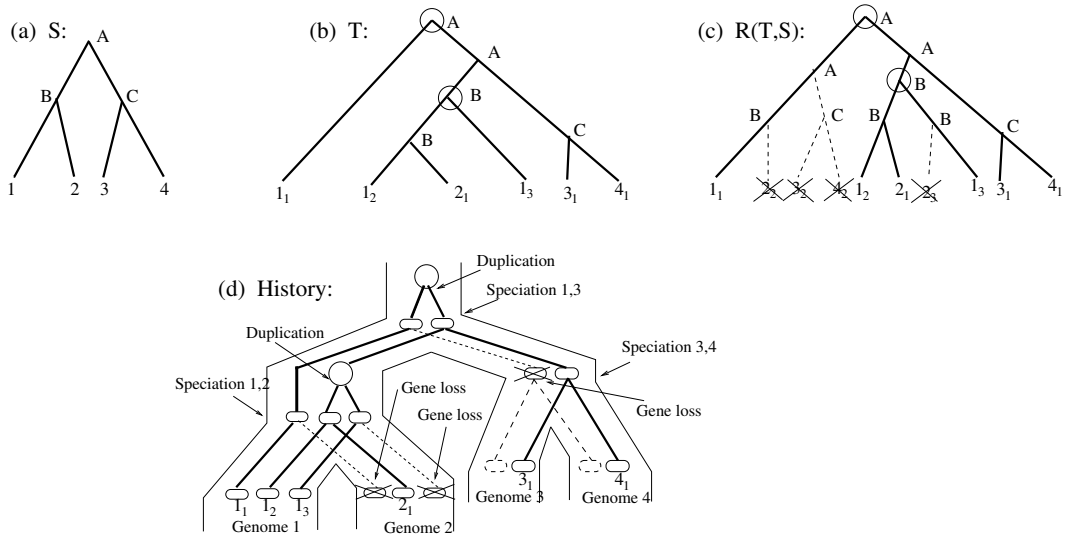

Figure 3.3 – (a) Un arbre d'espèces S pour $\mathscr{G} = \{1,2,3,4\}$. Les sommets internes de S sont A, B and C ; (b) Un arbre de gènes T. L'étiquette de feuille x représente une copie du gène dans le génome x. Les étiquettes des sommets internes sont obtenus en fonction du couplage LCA entre T and S. Les sommets de duplication de T par rapport à S sont marqués par des cercles. (Section Mapping LCA) ; (c) Une réconciliation $R(T,S)$ de T et S. La ligne pointillé représente une insertion de sous-arbre. La correspondance entre les sommets de $R(T,S)$ et S est indiquée par les étiquettes des sommets. Les sommets de $R(T,S)$ marqués sont des sommets de duplication ; les autres sont des sommets de spéciation. Cette réconciliation reflète une histoire évolutive de la famille de gènes ayant deux duplications et deux pertes.

Le ***coût en duplications*** associé à la réconciliation $M(T,S)$, noté par $\mathbf{d}(\mathbf{T},\mathbf{S})$ est le nombre de sommets de duplication de T par rapport à S. Zhang [31] et Eulenstein [8] proposent des algorithmes linéaires pour calculer la réconciliation par couplage LCA. GeneTree [24] implémente l'algorithme proposé par Zhang. Il existe aussi un algorithme quadratique, développé par Eddy et Zmasek [32] qui est plus facile à implémenter.

Il a été montré dans [3] que la réconciliation par mapping LCA minimise le coût de duplications, de pertes et par conséquent de mutations. De plus, on prouve que la réconciliation par mapping LCA est la seule qui minimise les pertes. Implicitement on déduit que minimiser les pertes minimise aussi les duplications. Dans le même article, Chauve et El-Mabrouk proposent également un algorithme linéaire pour calculer cette unique réconciliation qui minimise les pertes. L'approche vise le minimum d'insertions de sous-arbres et nous conduit vers le même arbre réconcilié obtenu par la méthode du mapping LCA.

Quand l'arbre d'espèces est inconnu, le réconciliation constitue un moyen de l'inférer à partir d'un ensemble d'arbres de gènes. Ce problème consiste à trouver l'arbre d'espèces qui minimise un certain critère pour les arbres de gènes donnés. [4, 16, 21] proposent des algorithmes pour le modèle des duplications et des mutations. A noter que l'inférence d'un arbre d'espèces optimal pour ces deux critères est un problème NP-difficile [21].

3.1.3 Noeuds de duplication et arbres MD

Chauve et El - Mabrouk [3] introduisent une nouvelle dimension dans la caractérisation des sommets de duplication. Dans le cas où les deux copies de gènes existent encore dans les espèces actuelles, ces deux copies peuvent être considérées comme un témoin de la duplication. Ceci est la raison pour laquelle on distingue entre deux catégories de sommets de duplications : les duplications apparentes et les duplications non-apparentes.

Formellement, soit T un arbre de gènes et S une arbre d'espèces. Tout sommet t de T tel que $\mathscr{G}(t_g) \cap \mathscr{G}(t_d) \neq \emptyset$ (c.a.d que les sous-arbres gauche et droit de l'arbre de racine t contiennent chacun une copie de gène du même génome) est un sommet de duplication, quelle que soit la réconciliation de T et S. On dit que t est un ***sommet de duplication ap-***

parente de T, abrégé ***sommet AD*** . Tel qu'introduit dans [3], on dit que T est *un arbre de duplication minimal (abrégé arbre-MD) en accord avec S*, ou *un arbre MD-consistant avec S*, si et seulement si le coût de duplication $d(T,S)$ est égal au nombre de sommets de duplications apparentes. En d'autres mots tout sommet de duplication de T est un sommet AD. Un sommet de duplication de T par rapport à S qui n'est pas un sommet AD est dit ***sommet de duplication non-apparente***, abrégé ***sommet NAD***.

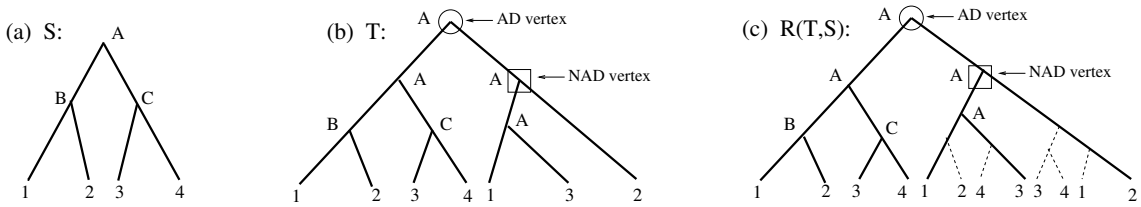

Figure 3.4 – (a) Un arbre d'espèces S pour $\mathscr{G} = \{1,2,3,4\}$. Les sommets internes de S sont A, B and C ; (b) Un arbre de gènes T. L'étiquette de feuille x représente une copie du gène dans le génome x. Les étiquettes des sommets internes sont obtenus en fonction du couplage LCA entre T and S. Les sommets AD de T par rapport à S sont marqués par des cercles et les sommets NAD par des carrés. (c) La réconciliation $R(T,S)$ de T et S. Les lignes pointillées représentent les insertions de sous-arbres nécéssaires pour trouver un accord entre T et S. Dans ce cas 4 insertions sont nécessaires.

La plupart du temps, l'approche des méthodes de réconciliation consiste à effectuer un minimum d'insertions de sous-arbres dans l'arbre de gènes afin d'obtenir un super-arbre qui est en accord avec la phylogénie des espèces. Les sommets NAD qui marquent les branchements problématiques d'un ou plusieurs triplets, peuvent être expliqués soit par des pertes de gènes, par une mauvaise identification des gènes dans le génome ou par des erreurs de construction de l'arbre de gènes. Quoi qu'il en soit, tout scénario d'évolution obtenu par la réconciliation d'un arbre de gènes et d'un arbre d'espèces est fortement dépendant des deux phylogénies. Les méthodes de réconciliation supposent que les deux arbres sont corrects. Si un des deux arbres contient des erreurs, des insertions supplémentaires doivent être effectuées afin d'arriver à un consensus. Dans ce dernier cas, Hahn [14] montre que la tendance des méthodes de réconciliation est d'inférer des duplications proches de la racine de l'arbre et des pertes vers les feuilles, si

l'arbre de gènes est incorrect. Dans ce travail j'introduis une approche permettant la correction des arbres de gènes, plus susceptibles de contenir des erreurs que les arbres d'espèces, comme une étape préliminaire à la réconciliation. On considère les sommets de duplication comme un signal d'erreur et on procède à l'élimination de feuilles qui sont possiblement mal placées.

3.1.4 Arbre consensus maximal (Maximum Agreement Subtree)

La comparaison de phylogénies joue un rôle important en biologie évolutive. L'inférence d'arbres évolutifs pour un ensemble d'espèces est réalisée en utilisant les différentes méthodes discutées au chapitre 2. Elles sont appliquées à des groupes de gènes homologues et souvent les topologies obtenues sont en désaccord.

Une méthode pour évaluer la congruence entre plusieurs arbres est d'obtenir l'ensemble maximal d'espèces pour lequel tous les arbres présentent une topologie commune. C'est un problème connu dans la littérature sous le nom ***d'arbre consensus maximal ("Maximum Agreement Subtree" - MAST)*** et il est rencontré en biologie moléculaire et en linguistique [5]. Finden et Gordon [10] introduisent une définition formelle :

DÉFINITION MAST

Soit $\mathcal{G} = \{1, 2, \cdots, n\}$ un ensemble d'étiquettes. On dit que T est un *arbre défini sur* $\mathcal{G}$ si chaque feuille de T est étiquetée par un élément de $\mathcal{G}$. On note par $I(T, \mathcal{G})$ un arbre *isomorphe* à T. Soit $\mathcal{G}'$ un sous-ensemble de $\mathcal{G}$.

Pour un ensemble d'arbres $T_1 \cdots T_k$ dèfinis sur $\mathcal{G}$, ***l'arbre consensus maximal*** MAST($T_1 \cdots T_k$) est le plus grand ensemble $\mathcal{G}' \in \mathcal{G}$, tel que MAST($T_1 \cdots T_k$) = $I(T_1, \mathcal{G}')$ = I (T_2, $\mathcal{G}'$)=...= I (T_k, $\mathcal{G}'$).

Dans le présent travail on s'intéresse au problème MAST, dans le contexte biologique, pour deux arbres binaires, soit l'arbre de gènes T et l'arbre d'espèces S respectivement. Voici la définition de ce problème :

PROBLÈME MAST POUR UN ARBRE DE GÈNES ET UN ARBRE D'ESPÈCES

Entrée : Un arbre de gènes T sur $\mathcal{G}$, à feuilles uniques et un arbre d'espèces S pour $\mathcal{G}$.

Sortie : Un arbre pondéré T_{MAX} inclus dans T et MD-consistant avec S de taille maxi-

male.

Pour le cas de deux arbres binaires, il existe un algorithme polynomial introduit par Steel et Warnow [26] pour résoudre ce problème. L'algorithme est basée sur une méthode de programmation dynamique qui calcule la liste de tous les sous-arbres pour chacun des arbres donné. Par la suite, on calcule le MAST pour chaque paire de sous-arbres des deux listes. Cette méthode est discutée en détail au Chapitre 4, Section 4.1.2.

Farach, Przytycka et Thorup [9] proposent un algorithme plus efficace, de complexité $O(n \log^3 n)$ et par la suite, Cole et al. [5] améliorent ce résultat et développent un algorithme de complexité $O(n \log n)$. Il montrent que le problème MAST dans le cas de deux arbres binaires enracinés se réduit au problème de la plus longue sous-séquence croissante ("longest increasing subsequence").

3.2 Problème considéré et motivation

Contrairement aux duplications apparentes, les duplications non-apparentes ne sont pas appuyées par la présence d'un gène en plusieurs copies dans le même génome. Un sommet NAD pointe vers des éventuels désaccords entre l'arbre des gènes et celui des espèces. Formellement, un sommet x de T *partage* trois espèces $\{a,b,c\}$ en $\{a,b;c\}$ si l'ensemble de génomes d'un enfant de x contient a et b, mais pas c et que l'ensemble de génomes de l'autre contient c, mais ni a, ni b. Pour tout sommet NAD x de T, il existe un triplet $\{a,b,c\}$ qui est partagé différemment par x que par l'image de x dans S donnée par le mapping LCA. Dans la figure Figure 3.3 le triplet (x,y,z) est partagé différemment par S que par T.

Selon des tests effectuées dans [3] sur des données simulées, basées sur 12 espèces *Drosophila* étudiées dans [15] on observe que 95% des duplications mènent vers des sommets AD. Ce fait suggère que les sommets NAD sont probablement dus à des gènes mal placés dans l'arbre de gènes. À noter que les sommets NAD pointent vers un sous-ensemble de la totalité des gènes mal placés, en considérant qu'un gène placé de manière aléatoire ne mène pas obligatoirement vers une contradiction. Par exemple, si pour un sommet AD t de T on rajoute une branche à l'un de ses descendants immédiats, ceci ne

transformera pas t en sommet NAD. Donc un ajout aléatoire de branche ne mènera pas nécessairement vers un sommet NAD.

Considérons le cas d'un gène mal placé qui mène vers un sommet NAD. Pour tout arbre réconcilié, la contradiction qu'engendre ce sommet aura comme effet direct l'augmentation du coût de mutation. Plus précisément des pertes additionnelles doivent être rajoutées pour expliquer l'incongruence 3.4. Par conséquent, pour un arbre de gènes T qui n'est pas MD-consistant avec S, on peut éliminer les triplets incongruents dévoilés par les sommets NAD en effectuant un certain nombre de suppressions de feuilles et transformer ainsi T en arbre MD-consistant avec S. À noter, qu'un arbre qui n'a que deux feuilles, est par définition MD-consistant avec tout arbre d'espèces. Il est en effet possible d'obtenir un arbre de gènes MD-consistant avec n'importe quel arbre d'espèces. Le présent travail exploite ces propriétés des sommets NAD et introduit un problème d'optimisation :

PROBLÈME DU MINIMUM DE SUPPRESSIONS DE FEUILLES
Soit T un arbre de gènes défini sur $\mathscr{G}$ et un arbre d'espèces S pour les génomes de $\mathscr{G}$.
En effectuant un nombre minimal de suppressions de feuilles dans T, on veut obtenir le plus grand arbre, noté T_{MAX}, inclus dans T et MD-consistant avec S.

Le Chapitre 4 décrit en détail l'algorithme permettant la correction d'un arbre de gènes. J'introduis d'abord l'algorithme Steel et Warnow qui permet de résoudre le problème *MAST*. Par la suite, je montre que le PROBLÈME DU MINIMUM DE SUPPRESSIONS DE FEUILLES pour certaines classes d'arbres, se réduit à une extension du PROBLÈME MAST. Je présente également une extension de cette technique pour le cas d'un arbre quelconque.

CHAPITRE 4

MÉTHODE POUR LA CORRECTION D'UN ARBRE DE GÈNES

La réconciliation est une méthode permettant d'expliquer le problème de désaccord entre un arbre d'espèces et un arbre inféré pour une famille de gènes homologues issus de ces espèces. Le scénario d'évolution inféré décrit l'évolution des gènes par duplications et pertes et intègre ces mécanismes dans l'histoire évolutive de leurs espèces. Toutefois, l'inconvénient majeur des méthodes de réconciliation est que tout scénario inféré pour l'évolution des gènes par duplications et pertes est fortement dépendant de la structure de l'arbre de gènes. Un seul gène mal placé peut mener à un scénario d'évolution totalement différent. Dans ce mémoire je suppose que l'arbre d'espèces est correct et que l'arbre de gènes est susceptible de contenir des erreurs. J'introduis une méthode permettant d'évaluer un arbre de gènes par rapport au scénario d'évolution obtenu par une méthode de réconciliation pour le modèle de duplications et pertes. En exploitant la notion de sommet NAD, introduite au chapitre précédant, on élimine les désaccords entre l'arbre de gènes et l'arbre d'espèces, par la suppression de feuilles de l'arbre de gènes. Ceci permet d'obtenir un arbre MD et minimiser les événements de duplications et pertes induits par la réconciliation.

On considère $\mathscr{G} = \{1, 2, \cdots, g\}$ un ensemble de g espèces et un arbre d'espèces S pour $\mathscr{G}$. Soit T un arbre pour une famille de gènes homologues de $\mathscr{G}$. On suppose que T n'est pas un arbre MD, c'est à dire qu'il existe au moins un sommet de duplication de T qui soit un sommet NAD. Les prochaines sections de ce chapitre décrivent deux types d'arbres pour lesquelles un algorithme exact a été développé ainsi que la méthode qui permet de corriger ces deux types d'arbres. La dernière section introduit une méthode heuristique pour le cas général.

4.1 Arbres à feuilles uniques

4.1.1 Problème du minimum de suppressions de feuilles pour un arbre à feuilles uniques

Definition 1. *Soit $\mathscr{G} = \{1, 2, \cdots, n\}$ un ensemble d'étiquettes et soit T un arbre défini sur $\mathscr{G}$. On dit que T est* ***un arbre à feuilles uniques*** *si T contient exactement n feuilles et chaque feuille de T est étiquetée par un élément distinct de $\mathscr{G}$.*

Soit T un arbre de gènes à feuilles uniques sur $\mathscr{G} = \{1, 2, \cdots, n\}$. Il représente l'évolution d'une famille de gènes contenant une seule copie par génome. Dans ce cas tout sommet de duplication t de T est un sommet NAD. De plus, il existe dans T au moins un sommet NAD puisque T n'est pas un arbre MD. On pose le problème de trouver l'ensemble minimal de feuilles à supprimer de T afin d'obtenir une topologie MD-consistante avec S. Ce problème est équivalent à trouver l'ensemble maximal de feuilles pour lequel T et S exhibent une topologie isomorphe. Par conséquent résoudre le PROBLÈME DU MINIMUM DE SUPPRESSIONS DE FEUILLES POUR UN ARBRE À FEUILLES UNIQUES sur $\mathscr{G}$ est équivalent à résoudre le PROBLÈME MAST pour deux arbres binaires définis sur $\mathscr{G}$. Voici une définition formelle :

PROBLÈME MAST
Entrée : Un arbre de gènes T sur $\mathscr{G}$, à feuilles uniques et un arbre d'espèces S pour $\mathscr{G}$.
Sortie : Un arbre pondéré T_{MAX} inclus dans T et MD-consistant avec S de taille maximale.

La figure 4.1 montre un exemple du problème du minimum de suppressions de feuilles pour un arbre à feuilles uniques sur $\mathscr{G} = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

4.1.2 Algorithme de correction d'un arbre à feuilles uniques

Dans cette section on décrit l'algorithme de programmation dynamique introduit par Steel et Warnow [26] qui permet de résoudre le problème MAST pour deux arbres binaires S et T de taille n, définis sur $\mathscr{G}$ et implicitement le PROBLÈME DU MINIMUM

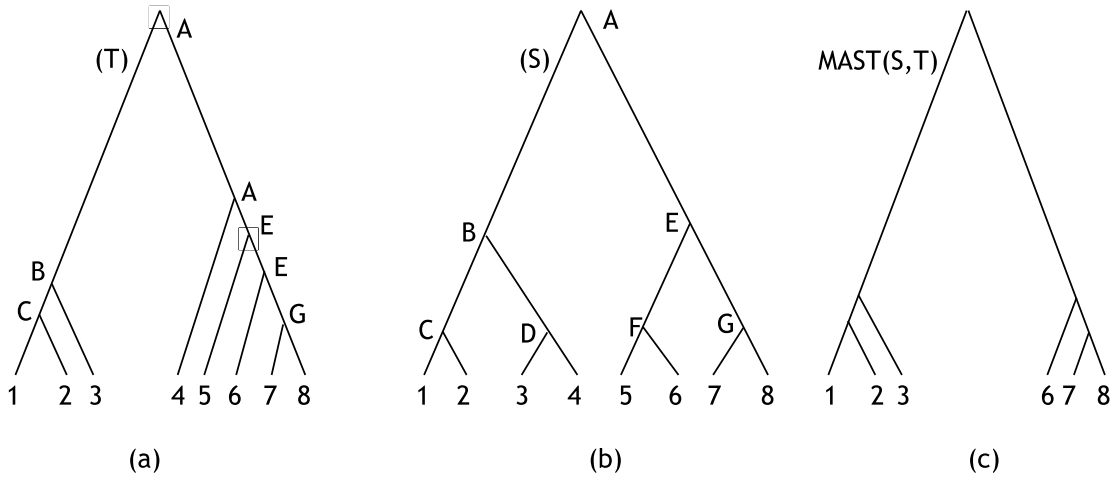

Figure 4.1 – (a) Un arbre de gènes T à feuilles uniques. (b) Un arbre d'espèces S pour $\mathscr{G} = \{1,2,3,4,5,6,7,8\}$. Les sommets internes de S sont A, B, C, D, E, F, G. Les étiquettes des sommets internes de T sont obtenus en fonction du couplage LCA entre T et S (Section 3.1.2.3). L'arbre T contient deux sommets *NAD* marqués par des *carrés*. (c) $MAST(S,T)$ est l'arbre d'accord maximal entre S et T. Deux feuilles sont supprimées de T afin d'obtenir un arbre de consensus maximal : {4, 5}.

DE SUPPRESSIONS DE FEUILLES POUR UN ARBRES À FEUILLES UNIQUES sur $\mathscr{G}$. À noter que l'algorithme Steel-Warnow concerne le cas général de deux arbres binaires non-enracinés. Dans le présent travail, cet algorithme est appliqué au cas particulier des arbres binaires enracinés.

Dans ce qui suit, nous adoptons les notations de [26]. Soit T un arbre binaire enraciné et soit $\mathscr{E}(T)$ l'ensemble des arcs de T. Un sous-arbre de T est obtenu de T par la suppression d'un arc $e \in \mathscr{E}$. Pour chaque sous-arbre $t \in T$ obtenu en supprimant $e_t \in \mathscr{E}(T)$, on note t_1 et t_2 les deux sous-arbres issus de la suppression du sommet incident à e_t (figure 4.2).

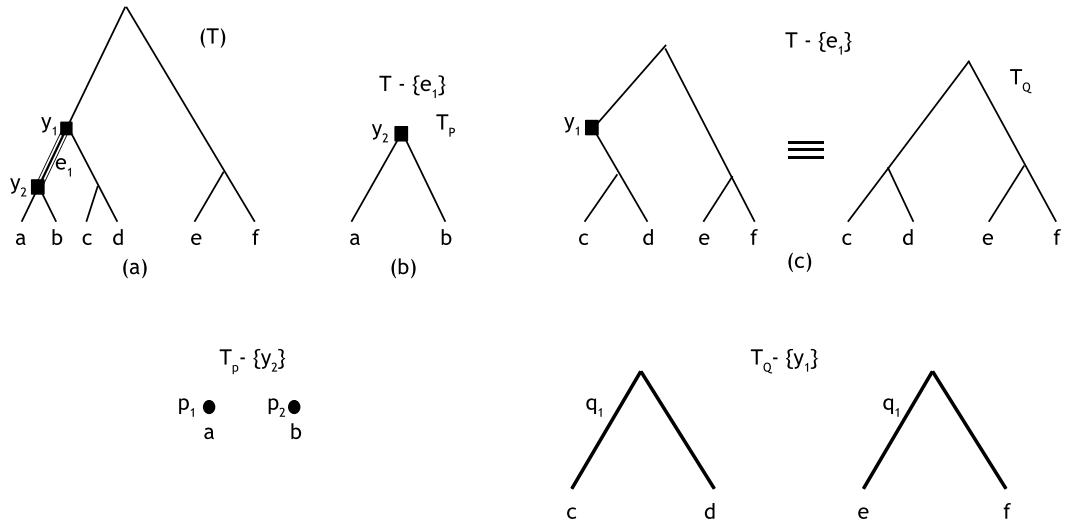

Figure 4.2 – (a) Un arbre T. (b) et (c) - représentent T_p et T_q les deux sous-arbres issus de la suppression de l'arc e_1 de$T$. (d) et (e) représentent les sous-arbres résultant de la suppression des deux sommets incidents à l'arc e_1, soit y_2 et y_1 des arbres T_p et T_q respectivement.

La première étape de l'algorithme consiste à construire la liste de tous les sous-arbres s, t de S et T respectivement, soit $\mathscr{O}(S)$ et $\mathscr{O}(T)$, ayant chacune $O(n)$ éléments. Ces deux listes sont ordonnées par ordre d'inclusion des sous-arbres : pour t_a et t_b - deux sous-arbres de T, si t_a est un sous-arbre de t_b, alors t_a précède t_b dans la liste $\mathscr{O}(T)$. On note également que chaque arbre est par définition sous-arbre de lui-même, donc $T \in \mathscr{O}(T)$

et $S \in \mathscr{O}(S)$. La prochaine étape consiste à calculer le produit cartésien $\mathscr{O}(T) \times \mathscr{O}(S)$, noté $\mathscr{L}$, qui consiste en une liste de paires de sous-arbres (s,t), où $s \in \mathscr{O}(S)$ et $t \in \mathscr{O}(T)$. On calcule ensuite $MAST(s,t)$ pour chaque paire $(s,t) \in \mathscr{L}$ selon la récurrence 4.1 :

$$\begin{aligned} MAST(s,t) &= |\mathscr{G}(s) \cap \mathscr{G}(t)| \text{ si } s \text{ ou } t \text{ est une feuille.} \\ MAST(s,t) &= max\{MAST(s_1,t_1) + MAST(s_2,t_2), \\ &\quad MAST(s_1,t_2) + MAST(s_2,t_1)\} \text{sinon.} \end{aligned} \tag{4.1}$$

Par cette approche de *programmation dynamique*, $MAST(S,T)$ est obtenu en évaluant les combinaisons des paires de tous les sous-arbres de S et T pour trouver celle ayant le maximum de feuilles, tout en étant conforme aux topologies des deux arbres. Une fois que le MAST des sous-arbres de s et t a été évalué, calculer $MAST(s,t)$ se fait en temps constant. Calculer le MAST de toutes les paires de sous-arbres possibles se fait donc en temps $O(n^2)$. Le coût total de l'algorithme Steel-Warnow et $O(n^2)$. On obtient l'ensemble maximal de feuilles pour lequel S et T exhibent une topologie commune. Ce résultat est équivalent au minimum de suppressions de feuilles à effectuer dans T afin d'obtenir une même topologie pour S et T. À noter que l'ensemble de feuilles à supprimer peut être obtenu en même temps que la cardinalité de l'ensemble de feuilles. Ceci permet de retrouver la topologie du MAST(S,T).

4.2 Arbres AD : Arbres de sommets AD inférieurs aux sommets NAD

Cette section présente une méthode exacte pour une deuxième classe d'arbres, soit les arbres AD. Je propose une généralisation de l'algorithme Steel et Warnow qui permet de résoudre le problème MAST pour ce type d'arbres. J'introduis aussi une méthode permettant d'obtenir des arbres à feuilles uniques à partir d'un arbre AD. Dans [7] on montre que le problème MAST pour cette classe d'arbres est équivalent au problème du minimum de suppressions de feuilles et dans cette section je résume cette preuve.

Soit S un arbre d'espèces pour $\mathscr{G}$. On s'intéresse ici seulement aux arbres de gènes qui ont la propriété AD suivante : aucun sommet AD de l'arbre n'est ancêtre d'un sommet NAD (voir à la figure 4.3(a)).

Definition 2. *Soit T un arbre de gènes sur $\mathscr{G}$. T est dit* arbre AD *ssi pour tout sommet NAD x de T, si y est un sommet de duplication, ancêtre de x, alors y est nécessairement un sommet NAD.*

Ci-dessous, j'explique une méthode de transformation permettant d'obtenir, à partir d'un arbre AD à feuilles multiples (i.e. la même étiquette potentiellement présente à plusieurs feuilles), un arbre pondéré à feuilles uniques, c'est-à-dire un arbre à étiquettes uniques avec un score attribué à chaque feuille. Je présente ensuite une généralisation du MAST à un arbre pondéré à feuilles uniques, puis une méthode basée sur ce MAST généralisé pour résoudre le PROBLÈME DU MINIMUM DE SUPPRESSIONS DE FEUILLES. Je discute brièvement de la preuve (preuve complète [7]) que la méthode développée est exacte pour le PROBLÈME DU MINIMUM DE SUPPRESSIONS DE FEUILLES. Introduisons d'abord notre méthode de transformation en un arbre pondéré à feuilles uniques.

Definition 3. *Soit U un arbre de gènes et S un arbre d'espèces sur $\mathscr{G}$.* ***Un arbre pondéré induit par*** *(S,U), noté U^I est obtenu à partir de S en supprimant toutes les feuilles qui n'appartiennent pas à $\mathscr{G}(U)$. Pour chaque feuille s restante dans S on associe un poids qui représente le nombre d'occurrences de s dans U, c'est à dire le nombre de feuilles de U qui sont étiquetées s.*

L'arbre pondéré induit par (S,U) est un sous-arbre de S et il respecte sa topologie. De plus, U^I est un arbre à étiquettes uniques. Pour un arbre AD T, soit $\{t_1,t_2,\cdots,t_k\}$ l'ensemble des plus hauts sommets AD de T et $\{T_1, T_2, ...T_k\}$ l'ensemble des sous-arbres enracinés à ces sommets respectifs. Puisque T est un arbre AD, alors les sous arbres $T_1,T_2,\cdots T_k$ ne contiennent pas de sommet NAD, donc chaque T_i, pour $1 \leq i \leq k$, est MD-consistant avec le sous-arbre de S induit par $\mathscr{G}(T_i)$. On remplace dans T chaque sous-arbre T_i, pour $1 \leq i \leq k$, par le sous-arbre pondéré T_i^I induit par $(S(lca(i),T_i)$. L'arbre T^I obtenu est un arbre à feuilles uniques. La figure 4.3(c) montre un arbre pondéré T^I induit par (S,T) (arbres 4.3(a,b)).

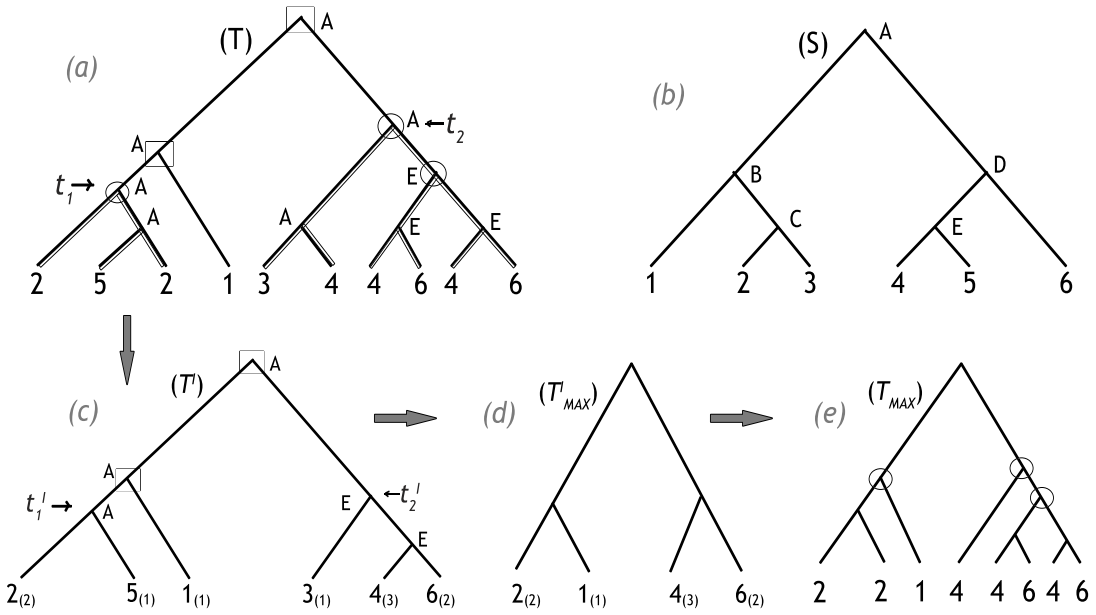

Figure 4.3 – (a) Un arbre de gènes AD, T. (b) Un arbre d'espèces S pour $\mathscr{G} = \{1,2,3,4,5,6\}$. Les sommets internes de S sont A, B, C, D, E. Les étiquettes des sommets internes de T sont obtenus en fonction du couplage LCA entre T et S (Section : 3.1.2.3). Les sommets AD de T sont marqués par des *cercles* et les sommets NAD par des *carrés*.$\{t_1,t_2\}$ est l'ensemble des plus hauts sommets AD de T (c) L'arbre pondéré T^I induit par (S,T), où chaque sous-arbre t_i de T enraciné au plus haut sommet AD est remplacé par t_i^I, le sous-arbre pondéré équivalent. (d) T^I_{MAX} est l'arbre pondéré induit par (S,T)de taille maximale en accord avec S et T^I - le $MAST_p(S,T^I)$. (e) T_{MAX} est l'arbre induit par T^I_{MAX} sur T - le $MAST(S,T)$ et un arbre MD-consistant avec S, obtenu par un minimum de suppressions de feuilles de T.

4.2.1 Généralisation de Steel et Warnow pour arbres pondérés à feuilles uniques

L'algorithme de programmation dynamique présenté à la section 4.2 peut être adapté aux arbres à feuilles uniques pondérées. On considère S un arbre d'espèces pour $\mathscr{G}$ et T un arbre AD sur $\mathscr{G}$. Soit T^I l'arbre pondéré induit par (S,T). À noter que S est par définition un arbre à feuilles uniques et on consid̀ere que ses feuilles sont pondérées à 1. Introduisons $MAST_p$, la généralisation de Steel et Warnow qui permet de résoudre le PROBLÈME MAST PONDÉRÉ, pour S et T^I. La *valeur d'un arbre pondéré, notée V*, est la somme des pondérations des feuilles de l'arbre.

MAST PONDÉRÉ ($MAST_p$) :
Entrée : Un arbre pondéré T sur $\mathscr{G}$ et un arbre d'espèces S pour $\mathscr{G}$.
Sortie : Un arbre pondéré T_{MAX} inclus dans T et MD-consistant avec S de valeur maximale.

La première étape consiste à calculer de la même façon que dans la section précédente, les listes $\mathscr{O}(S)$ et $\mathscr{O}(T^I)$ de tous les sous-arbres s, et t^I de S et T^I respectivement, $O(n)$ éléments. Par la suite, on calcule $MAST_p(s,t^I)$ pour chaque $s \in \mathscr{O}(S)$ et $t^I \in \mathscr{O}(T^I)$, de la façon suivante :

Pour t^I ou s, *feuille* de T, S respectivement, on a
- si $\mathscr{G}(s) \cap \mathscr{G}(t^I) \neq \phi$, soit $x \in \mathscr{G}(s) \cap \mathscr{G}(t^I)$ et soit i la pondération de x dans S et j la pondération de x dans t^I, alors
$MAST_p(s,t^I) = max\{i,j\}$ - si $\mathscr{G}(s) \cap \mathscr{G}(t^I) = \phi$, alors
$MAST_p(s,t^I) = \phi$.

Pour t^I et s, deux *sous-arbres* de T et S respectivement, on a
MAST(s,t) = *max* { MAST(s_1,t_1) + MAST(s_2,t_2), MAST(s_1,t_2) + MAST(s_2,t_1) }

La figure 4.4 présente l'algorithme permettant de résoudre le PROBLÈME MAST

POUR ARBRES PONDÉRÉS, T^I et S.

ALGORITHME $MAST_p(S, T)$
1. Calculer $\mathscr{O}(T^I)$ - la liste de sous-arbres de T^I ;
2. Calculer $\mathscr{O}(S)$ - la liste de sous-arbres de S ;
3. Construire $\mathscr{L} = \mathscr{O}(S)$ x $\mathscr{O}(T^I)$, la liste des paires (s,t^I) de sous-arbres $s \in S$ et $t^I \in T^I$
4. POUR TOUT $(s,t^I) \in \mathscr{L}$ FAIRE
5. si $\mathscr{G}(s) \cap \mathscr{G}(t^I) \neq \phi$ alors
6. soit $x \in \mathscr{G}(s) \cap \mathscr{G}(t^I)$ et
7. soit i la ponderation de x dans S
8. soit j la ponderation de x dans t^I
9. $MAST_p(s,t^I) = max\{i, j\}$
10. si $\mathscr{G}(s) \cap \mathscr{G}(t^I) = \phi$, alors
11. $MAST_p(s,t^I) = \phi$.
12. FIN POUR

Figure 4.4 – Algorithme qui calcule le $MAST_p$ pour S et T^I, deux arbres pondérés à feuilles uniques.

4.2.2 Problème du minimum de suppressions de feuilles pour un arbre AD

Soit S un arbre d'espèces pour $\mathscr{G}$ et soit T un arbre de gènes de type AD sur $\mathscr{G}$. *Le problème du Minimum de suppression de feuilles* pour T et S est équivalent au $MAST_p$ pour S et T^I [7](Theorem 1). L'algorithme 4.5 permet de résoudre le PROBLÈME DE MINIMUM DE SUPPRESSIONS DE FEUILLES pour un arbre AD.

On peut montrer que résoudre la généralisation du problème MAST aux arbres pondérés est équivalent à résoudre le PROBLÈME DU MINIMUM DE SUPPRESSIONS DE FEUILLES pour un arbre AD. Le théorème principal de [7] stipule cette équivalence :

Théorème 1 **:**
Soit $\mathscr{G} = \{1, 2, \cdots, g\}$ un ensemble de g espèces et S un arbre d'espèces pour $\mathscr{G}$. Soit T un arbre de gènes AD sur $\mathscr{G}$. Soit T^I_{MAX} la solution de la généralisation de MAST PONDÉRÉ *pour T^I et S, et T_{MAX} le sous-arbre de T induit par T^I_{MAX}. Alors T_{MAX} est une solution du* PROBLÈME DU MINIMUM DE SUPPRESSIONS DE FEUILLES *pour T et S.*

ALGORITHME *CorrigerArbreAD*(S, T)
1. Calculer l'arbre pondere T^I induit par (S,T) :
2. POUR TOUT sommet AD $i \in T$ FAIRE
3. Calculer l'arbre pondere t_i^I induit par $(S_l ca(t_i), t_i)$;
4. Remplacer dans T chaque t_i par l'arbre pondere t_i^I ;
5. FIN POUR
6. Calculer l'arbre T^I_{MAX}= $MAST_p(T^I, S)$
7. Construire T_{MAX} l'arbre induit sur T par T^I_{MAX}
8. Retourner($|T|$ - $|T_{MAX}|$).

Figure 4.5 – Algorithme qui résoud PROBLÈME DU MINIMUM DE SUPPRESSIONS DE FEUILLES POUR UN ARBRE AD. On calcule l'arbre pondéré T^I induit par (S,T) d'abord et on résoud $MAST_p$ pour S et T^I. Ceci nous permet d'obtenir le nombre de feuilles à supprimer de T pour obtenir un arbre MD-consistant avec S et "corriger" ainsi T.

Dans [7], le *Lemme 1* montre d'abord que T_{MAX}, l'arbre induit sur T par T^I_{MAX} = $MAST_p(S, T^I)$(4.3(d)) est un arbre MD-consistant avec S. On suppose le contraire, que T_{MAX} contient un sommet NAD x et donc un triplet (a,b,c) qui contredit la phylogénie de S. Aucun sommet ne peut être transformé en sommet de duplication par des suppressions de feuilles. Alors il n'y a que deux possibilités pour x dans T : x est un sommet AD ou bien x est un sommet NAD. Dans le cas ou x est NAD, comme T^I_{MAX} ne contient pas de duplications (il est un arbre pondéré), alors il en résulte que a, b ou c est absent de $T^I_{MAX}(x)$. Ceci contredit l'hypothèse que (a,b,c) appartient à T_{MAX}.

Dans le cas ou x est un sommet AD, il existe une feuille d présente à la fois dans Tx_l et dans Tx_r. Soit $s = lca(x)$. Alors $d \in Ss_l$ ou bien $s \in Ss_r$. Également on a a,b ou $c \in Ss_l$ ou Ss_r aussi. Comme on procède à une suppression de feuille dans le sous-arbre T_x, il en résulte que le parent de x dans T^I, noté y, est un sommet NAD qu'on veut éliminer. Implicitement y et x ont le même LCA s dans S. Et par une suppression de feuille de T^I_x on obtient un LCA différent pour x et y, afin de s'assurer qu'on élimine le sommet NAD de T^I. Donc, si après une ou plusieurs suppressions de feuilles, T^I_x contient toujours a, b, c et d, il en résulte que S exhibe la phylogénie $((a,b,c),d)$ ce qui est une contradiction.

Par le suite, *Lemme 2* montre que T_{MAX} est l'arbre de taille maximale inclus dans T

qui est MD-consistant avec S. Si une feuille i d'étiquette s est supprimée de T afin de résoudre un sommet NAD, alors toutes les feuilles d'étiquettes s seront supprimées de T. Ceci est dû au fait que T est un arbre AD est les sommets AD de T appartiennent à des sous-arbres ayant comme racine un sommet NAD s'il sont concernés par une suppression de feuille.

4.3 Algorithme pour le cas général.

Cette section présente une méthode générale permettant de corriger un arbre de gènes par rapport à un arbre d'espèces. Dans le cas d'un arbre de gènes T satisfaisant la contrainte AD, l'algorithme présenté revient à celui présenté à la section précédente. C'est donc un algorithme exact dans ce cas. Cependant, dans le cas général d'un arbre ne satisfaisant pas la contrainte AD, alors l'algorithme est une heuristique, qui n'est pas garantie de donner la solution optimale.

Après avoir effectué le mapping LCA de T par rapport à S, on cherche dans T des sous-arbres U (pondérés ou non) ayant deux propriétés particulières en ce qui concerne la distribution des sommets AD et NAD.

Propriété NAD : U ne contient pas de sommet AD ; U est donc un arbre à feuilles uniques qui ne contient que des sommets NAD.

Propriété AD : La racine de U est un sommet AD et U ne contient pas de sommet NAD.

Plus précisément, un sous-arbre maximal de T qui vérifie la propriété *Propriété NAD* est un sous-arbre maximal à feuilles uniques de T. Les sous-arbres maximaux de T vérifiant la *Propriété AD* sont les sous-arbres maximaux de T qui peuvent être remplacés par les sous-arbres pondérés équivalents, tel que présenté à la section 4.2.

On définit deux notions : ***la frontière-NAD*** de T est l'ensemble des sommets de T des sous-arbres maximaux de T vérifiant la *Propriété NAD* ; ***la frontière-AD*** de T est l'ensemble des sommets de T qui constituent des sous-arbres maximaux de T vérifiant

la *Propriété AD*.

```
ALGORITHM CORRIGERARBRE (T, S,)
 1.  nbSuppressions = 0 ;
 2.  SI T est un arbre MD-consistant avec S ALORS
 3.      RETOURNE(nbSuppressions)
 4.  FIN SI
 5.  T^I = T ;
 6.  POUR TOUT x ∈ frontiere-AD(T) FAIRE
 7.      Remplacer T^I(x) par l'arbre pondere equivalent dans T^I
 8.  FIN POUR
 9.  POUR TOUT x ∈ frontiere-NAD(T^I) FAIRE
10.      T^I_MAX(x) = MAST_p(T^I_x, S) ;
11.      Remplacer T_x dans T par le sous-arbre induit par T^I_MAX(x) ;
12.      nbSuppressions = nbSuppressions + |T^I_x| − |T^I_MAX(x)| ;
13.  FIN POUR
14.  RETOURNE(nbSuppressions +CorreigerArbre(T, S))
```

Figure 4.6 – L'algorithme *CorrigerArbre* prend en entrée un arbre de gènes et un arbre d'espèces et retourne le nombre de suppressions de feuilles nécessaire afin de transformer T en un arbre MD-consistant avec S.

L'algorithme *CorrigerArbre* présenté à la figure 5.1 procède en deux étapes :

La condition d'arrêt - Ligne 2 à 4 :

Si T est un arbre MD-consistant avec S, alors T ne nécessite aucune suppression de feuilles et l'algorithme termine.

La récurrence - Ligne 6 à 13 :

Cette étape consiste à résoudre les sous-arbres maximaux de T qui sont des sous-arbres AD, en suivant la procédure décrite à la section 4.2. D'abord on pondère les sous-arbres de T qui ne contiennent que des sommets AD, c'est à dire les sous-arbres enracinés aux sommets de la frontière-AD, pour obtenir T^I. Par la suite, pour chaque sommet x $\in$ frontière-NAD, on résout le sous-arbre AD maximal t_x de T par $MAST_p(S, t_x^I)$. t_x est remplacé dans T par l'arbre équivalent induit par $MAST_p(S, t_x^I)$ et le nombre de feuilles supprimées est enregistré.

Si T est un arbre à feuilles uniques alors la frontière-AD ne contient aucun sommet et la frontière-NAD est la racine de T. T sera résolu par la méthode présentée à la section 4.1 et la récurrence sera exécutée une seule fois.

Si T est un arbre AD, il sera résolu par la procédure présentée à la la section 4.2. La frontière-NAD devient la racine de T et la récurrence sera exécutée une seule fois.

Si T est un arbre quelconque, la récurrence peut être exécutée plusieurs fois avant d'obtenir un arbre MD-consistant avec S et dans ce cas un résultat optimal n'est pas garanti. Si après une exécution l'arbre contient toujours des sommets NAD, l'algorithme *CorrigerArbre* est appliqué de nouveau jusqu'à l'obtention d'un arbre MD. Ce dernier n'est pas nécessairement l'arbre consensus maximal, donc le nombre de suppressions de feuilles obtenu n'est peut-être pas le minimum. Cette problématique ouvre la question sur l'efficacité de l'algorithme, discutée plus en détail dans le chapitre suivant.

La figure 4.7 montre un exemple d'exécution dans le cas général d'un arbre quelconque (i.e un arbre qui n'est pas AD). On commence par la pondération de chaque arbre enraciné aux sommets de la frontière-AD et par la suite on résout avec $MAST_p$ chaque sous-arbre enraciné aux sommets NAD de la frontière-NAD. L'arbre induit sur T après la résolution des sous-arbres AD maximaux par $MAST_p$, n'est pas un arbre MD-consistant avec S et le processus recommence. Dans cet exemple deux exécutions de la récurrence sont nécessaires afin d'obtenir un arbre MD-consistant avec S en effectuant deux suppressions de feuilles - ce qui est le résultat optimal pour S et T.

4.3.1 Complexité de l'algorithme

Soit n la taille de l'arbre de gènes T. Déterminer le type d'arbre (MD ou non), ainsi que les sommets NAD et les sommets AD (Lignes 2 à 4) nécessite le calcul du mapping LCA de T vers S pour lequel il existe un algorithme linéaire [33]. Vérifier la *condition d'arrêt* ainsi que déterminer la frontière-NAD (Ligne 2 à 4) prend un temps O(n). La frontière-NAD peut contenir au plus $O(n)$ sommets, donc l'algorithme $MAST_p$ peut être exécuté $O(n)$ fois, dans le pire des cas. La complexité de l'algorithme $MAST$, ainsi que la généralisation $MAST_p$ est en $O(n^2)$. Il existe toutefois un algorithme de complexité $O(nlogn)$ [5] pour résoudre le problème $MAST$. Dans [27] on montre qu'il est possible

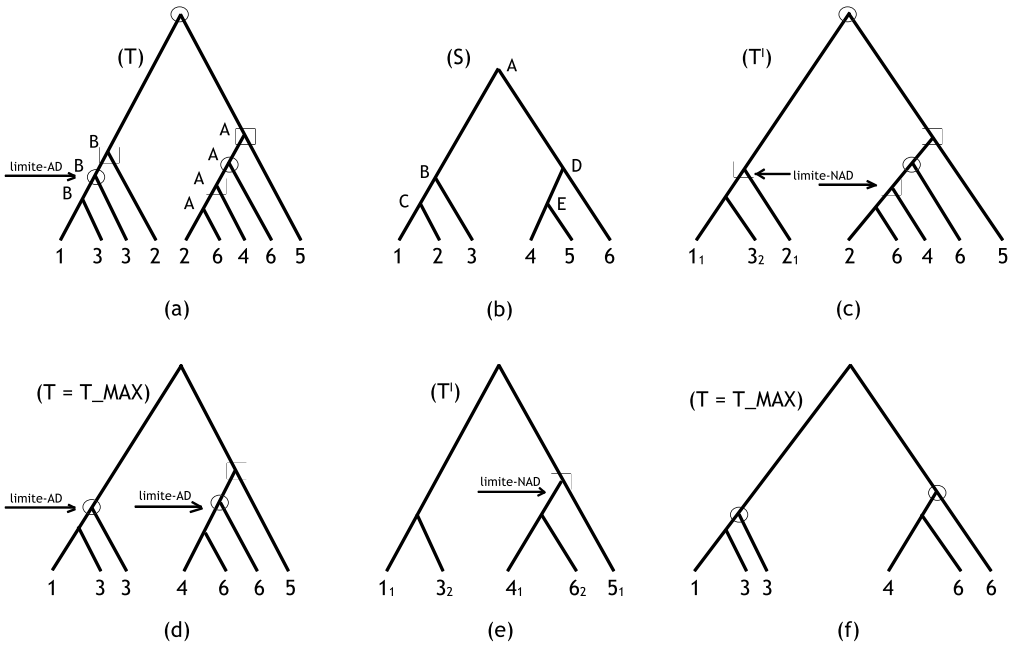

Figure 4.7 – (a) Un arbre de gènes T. (b) Un arbre d'espèces S pour $\mathscr{G} = \{1,2,3,4,5,6\}$. Les sommets internes de S sont A, B, C, D, E et étiquettes des sommets internes de T représentent le LCA du sommet de T dans S.Les sommets AD de T sont marqués par des *cercles* et les sommets NAD par des *carrés*. La première etape (a) consiste à trouver la frontière-NAD de T et remplacer le sous-arbre $T(x)$ pour chaque sommet x *in* frontière-AD avec le sous-arbre pondéré induit par $(T(x), S)$. On obtient ainsi T^I. Deuxièmement (c), on trouve la limite-NAD de T^Iet par la suite (d) pour chaque sommet $y \in$ frontière-NAD on résout le $MAST_p(T^I(y), S) = T^I(y)_{MAX}$. Chaque sous-arbre $T^I(y)$ de T^I est remplacé par l'arbre induit par $T^I(y)_{MAX}$, et on obtient un arbre qui nécessite de nouveau une évaluation (e). Le processus est repris avec la récurrence (Algorithme CorrigerArbre - Ligne 6-13) et s'arrête dans ce cas-ci après la deuxième boucle, avec l'arbre T qui est MD-consistant avec S obtenu en (f).

de le généraliser au problème $MAST_p$, alors la complexité de l'algorithme *CorrigerArbre* devient O($n^2 logn$).

CHAPITRE 5

RÉSULTATS

Tel que discuté au chapitre précédent, dans le cas d'un arbre de gènes à feuilles uniques ou bien d'un arbre de gènes *AD*, l'algorithme *CorrigerArbre* donne un résultat optimal pour le PROBLÈME DU MINIMUM DE SUPPRESSIONS DE FEUILLES. En ce qui concerne le arbres *NAD*, qui contiennent au moins un sommet *AD* au-dessus d'un sommet *NAD*, il n'est par certain que l'algorithme *CorrigerArbre* donne une réponse optimale. Dans cette section on s'intéresse au cas des arbres *NAD* et on présente les méthodes qui nous ont permis d'évaluer la performance de l'algorithme heuristique ainsi que les résultats obtenus.

Plus précisément, on cherche à évaluer la performance de la méthode heuristique qui permet de corriger un arbre *NAD* par comparaison avec le résultat optimal. À cette fin, nous avons comparé les résultats de *CorrigerArbre* avec un algorithme naïf qui essaie toutes les combinaisons de suppressions de feuilles afin de déterminer le minimum de feuilles à supprimer. Supposons que le nombre de feuilles à supprimer obtenu en appliquant *CorrigerArbre* sur un arbre *NAD* soit $NbObtenu$. L'algorithme naïf essaie toutes les combinaisons de $NbObtenu - 1$ feuilles sur $\mathscr{G}$ afin de déterminer s'il existe un arbre consensus entre T et S de taille plus grande que celui obtenu. Si on trouve un tel arbre ceci est équivalent à dire que *CorrigerArbre* n'a pas trouvé la réponse optimale puisqu'il existe un arbre consensus de taille plus grande donc un plus petit ensemble de feuilles à supprimer de T pour le transformer en arbre $MD - consistant$ avec S. On continue de tester toutes les combinaison avec $NbObtenu - 2$, $NbObtenu - 3 \cdots$ 1 tant qu'on trouve un arbre consensus possible et on arrête lorsqu'on n'en trouve plus. La Figure 5.1 donne les détails de cet algorithme naïf qui trouve l'erreur de notre algorithme pour un arbre *AD* en comparant à la réponse optimale.

La complexité exponentielle de l'algorithme naïf $TrouveOptimum$ fait en sorte que les recherches sur l'erreur sont limitées par la taille de l'arbre de gènes. Nous avons considérés un ensemble de 5 génomes et des arbres de gènes de tailles t allant de 6 à 24

```
Algorithme TrouveOptimum(S, T, NbObtenu)
1.  k = NbObtenu − 1 ;
2.  TANT QUE k > 0 ;
3.      SOIT 𝒮((T_k) la liste de tous les sous-arbres de T
        obtenus par la suppression de k feuilles ;
4.      POUR TOUT T_i ∈ 𝒮(T_k) FAIRE
5.          Calculer MAST(S, T_i) ;
6.          SI ( MAST(S, T_i)) == k) ALORS
7.              k = k-1 ;
8.              return (Algorithme TrouveOptimum(S, T ,k)) ;
9.          SINON
10.             NbOptimal = k+1 ;
11.         FIN SI
12.     FIN POUR
13. FIN TANT QUE
14. NbOptimal = k ;
15. return NbOptimal ;
```

Figure 5.1 – Algorithme naïf qui trouve *NbOptimal*, le minimum de suppressions de feuilles à effectuer dans T afin d'obtenir un arbre consensus. *L'erreur* de l'algorithme *CorrigerArbre*, est égale à $NbObtenu - NbOptimal$ et vaut 0 si $NbObtenu = NbOptimal$.

(avec un saut de 2). Pour chaque taille t, 500 arbres *NAD* ont été aléatoirement générés. Le diagramme 5.2 montre que pour plus de 65% des arbres *NAD* générés *CorrigerArbre* trouve la réponse optimale, ce qui est équivalent à dire qu'il y a 0 erreurs pour plus de 65% des arbres générés.

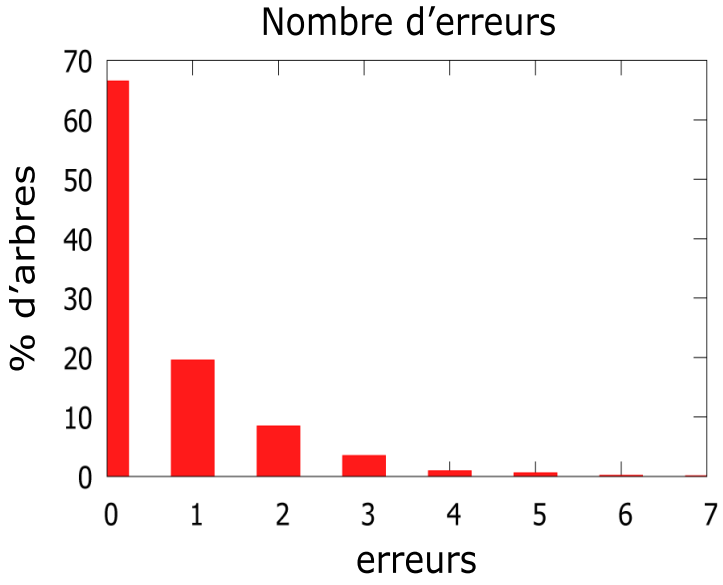

Figure 5.2 – *NbObtenu* - le nombre de suppressions obtenu par l'algorithme *CorrigerArbre versus NbOptimal* - le nombre optimal de suppressions à effectuer obtenu par l'algorithme *TrouveOptimum*. L'erreur est définie comme $NbObtenu - NbOptimal$. Dans plus de 65% des cas l'erreur est 0 et l'algorithme *CorrigerArbre* trouve la réponse optimale. On remarque que si la réponse de notre algorithme n'est pas optimale, c'est-à-dire $NbObtenu \neq NbOptimal$, la plupart du temps la réponse diffère de 1.

La figure 5.3 montre que le taux d'erreur, $(NbObtenu - NbOptima)l/NbObtenu$, est indépendant de la taille de l'arbre et il est inférieur à 0.15. Ce résultat est obtenu en effectuant la moyenne du taux d'erreur pour tous les arbres d'une certaine taille. Des tests additionnels montrent que le taux d'erreur est indépendant du nombre de sommets *AD* ou du nombre de sommets *NAD*. En fait, le taux d'erreur dépend du nombre de fois que la *récurrence* de *CorrigerArbre* est effectuée, c'est-à-dire du nombre de niveaux

intercalants de relation AD au-dessus des NAD (propriété d'un arbre NAD). (exemple figure 5.4)

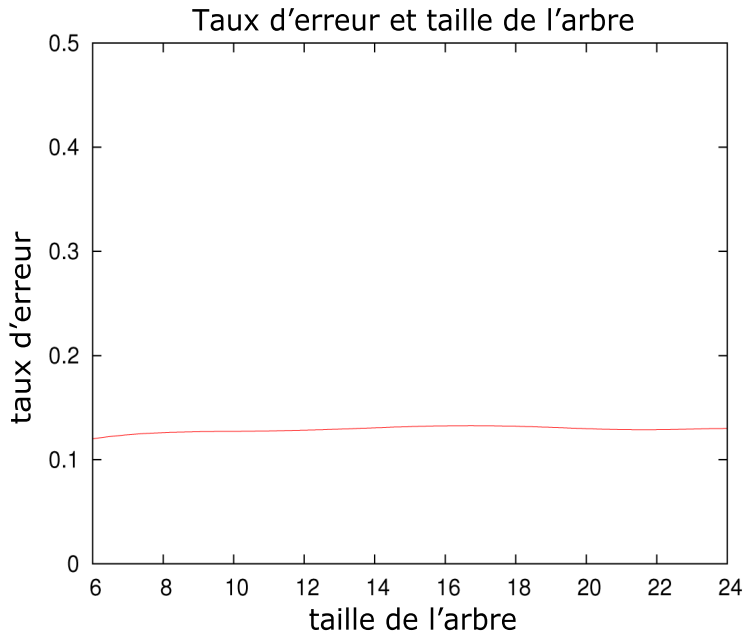

Figure 5.3 – Le taux d'erreur, défini comme $NbObtenu - NbOptimal/NbObtenu$ est inférieur à 0.15 et ne dépend pas de la taille de l'arbre de gènes.

Afin d'évaluer la capacité de l'algorithme à détecter les feuilles mal placées, un ensemble de 10 génomes et des arbres de gènes de tailles allant de 10 à 100 (avec un saut de 10) ont été considérés. Pour une topologie d'arbre d'espèces générée aléatoirement et pour un arbre de gènes T de tailles t, allant de 10 à 100, $MD - consistant$ avec S, un nombre de feuilles $NbInsertions = t/10$ a été aléatoirement inséré, c'est-à-dire 10% du nombre total de feuilles de T. Ceci permet d'évaluer le nombre de feuilles mal placées détectées par l'algorithme CorrigerArbre. Pour chaque taille t allant de 10 à 100, le pourcentage de détection défini comme $(NbObtenu/NbInsertions) \times 100$ a été calculé pour 100 arbres différents par taille. La figure 5.5 représentant nos résultats, montre que le pourcentage de détection diminue si la taille de l'arbre augmente. Ceci s'explique surtout par le fait qu'un arbre MD-consistant n'a pas besoin de suppressions de feuilles et son pourcentage de détection est 100% et plus on rajoute de feuilles (1 feuille pour

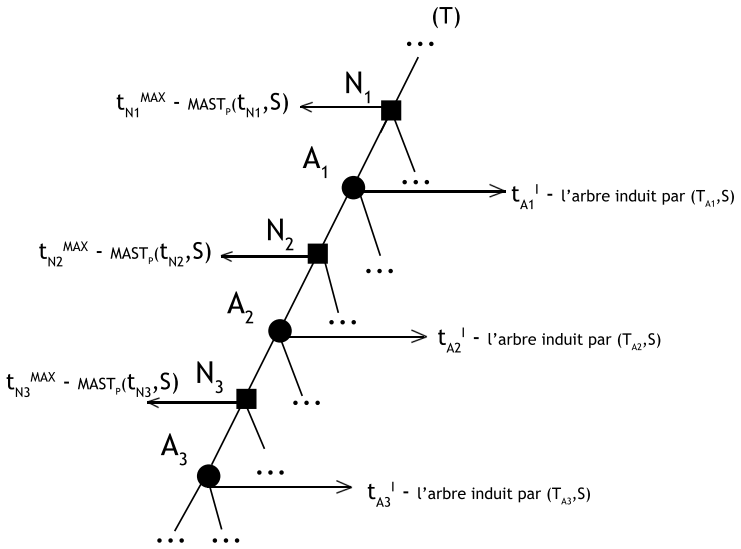

Figure 5.4 – Le taux d'erreur dépend du nombre de fois que la récurrence de l'algorithme *CorrigerArbre* est répétée. Ceci est relié au nombre de niveaux intercalants de relation AD au-dessus des NAD, dans ce cas 3 : $A_1 - N_1$, $A_2 - N_2$, $A_3 - N_3$

$t = 10$ et 10 feuilles pour $t = 100$) plus on a de chances d'obtenir un arbre qui n'est pas MD-consistant avec S. Si on ne considère pas le cas des arbres qui sont toujours MD-consistants après les insertions de feuilles aléatoires, le pourcentage de détection est à 40%.

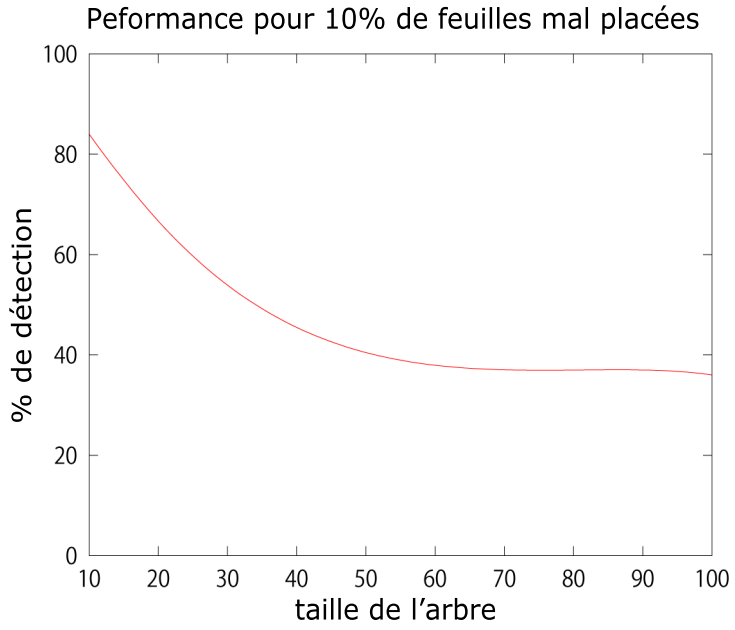

Figure 5.5 – Pourcentage de détection des feuilles mal placées dans T pour l'algorithme CorrigerArbre : $(NbObtenu/NbInsertions)x100$ est dépendent pas de la taille de l'arbre et diminue si cette dernière augmente, considérant un nombre d'insertions de feuilles égal à 10% de la taille de T.

CHAPITRE 6

CONCLUSION

6.1 Discussion

En exploitant la notion de sommet *NAD* qui peut constituer un indicateur de gène mal placé, le présent travail propose l'algorithme polynomial *CorrigerArbre*, qui permet de transformer l'arbre de gènes en arbre $MD-consistant$ avec l'arbre d'espèces par un minimum de suppressions de feuilles, éliminant ainsi tout sommet *NAD*.

L'algorithme *CorrigerArbre* est exact pour un arbre de gènes à feuilles uniques, ou pour un arbre de gènes *AD* c'est à dire un arbre sans sommets *AD* au-dessous des sommets *NAD*. Dans le cas général d'un arbre quelconque, *CorrigerArbre* est une heuristique et nos simulations montrent que ses résultats sont très proches de l'optimum pour ce qui est du nombre minimum de feuilles à supprimer.

Les sommets *NAD* regroupent un ensemble de gènes qui sont en désaccord les uns par rapport aux autres. Ces sommets témoignent d'une partition erronée pour un triplet de gènes $\{a,b,c\}$. Pour la résolution d'un tel sommet, l'élimination de n'importe quel gène appartenant à ce triplet suffit. Ainsi, une limitation de notre méthode qui vise l'élimination des sommets *NAD* afin d'obtenir un arbre de gènes en accord avec la phylogénie des espèces, est qu'elle ne peut détecter précisément les gènes mal placés. Elle offre cependant un bonne estimation de leur nombre. Une extension possible est d'inférer tous les ensembles optimaux de feuilles à supprimer et d'utiliser des valeurs de bootstrapping pour les arcs afin de faire un choix des gènes à supprimer.

Afin de palier à ce problème de détection des gènes mal placés, le coût de réconciliation peut être utilisé. Un sommet *NAD* d'un point de vue biologique, témoigne d'une ou plusieurs pertes ayant suivi une duplication. Tel que mentionné précédemment, l'algorithme *CorrigerArbre* peut être utilisé comme étape préliminaire à la réconciliation afin d'optimiser les coûts de mutation subséquents. De la même façon, la réconciliation peut être utilisée pour trouver les feuilles mal placées en effectuant pour chaque ensemble de

feuilles à supprimer la réconciliation de l'arbre au complet et évaluer le coût de réconciliation.

D'autre part, une feuille insérée aléatoirement ne résulte pas nécessairement en la création d'un sommet *NAD*. Ainsi les sommets *NAD* pointent vers un sous-ensemble de ces feuilles insérées de façon arbitraire et nos simulations montrent que l'algorithme CorrigerArbre détecte en moyenne 40% de ceux-ci. Toutefois, les gènes mal placés entrainent une augmentation significative du coût de mutation de l'arbre, et l'élimination des sommets *NAD* comme une étape préliminaire à la réconciliation contribue à contrecarrer cet aspect. On obtient ainsi un meilleur portrait de l'évolution de la famille de gènes par duplication et pertes et on évite des coûts invraisemblables.

6.2 Extensions et perspectives

L'algorithme *CorrigerArbre* peut être utilisé dans le contexte d'inférence phylogénétique, afin de choisir parmi un ensemble d'arbres équiprobables obtenus par une méthode d'inférence phylogénétique celui qui peut être transformé en arbre *MD* en un minimum de suppressions de feuilles.

Il peut également contribuer au choix de l'arbre d'espèces pour un ensemble d'arbres de gènes comme complément à la réconciliation afin de minimiser le coût de mutation. On choisit l'arbre d'espèces pour lequel des arbres de gènes *MD*-consistants sont obtenus par un minimum de suppressions de feuilles.

Dans l'article [27], soumis au journal "Algorithms for Molecular Biology", on montre que l'algorithme polynomial permettant de résoudre le PROBLÈME DU MINIMUM DE SUPPRESSIONS DE FEUILLES pour un arbre d'espèces et un arbre de gènes peut être généralisé pour une forêt d'arbres de gènes. De plus, dans le contexte d'inférence d'arbres d'espèces, on montre qu'une heuristique en temps polynomial nous permet de résoudre le PROBLÈME DU MINIMUM DE SUPPRESSIONS D'ESPÈCES afin de retrouver un ensemble d'arbres de gènes pour lesquels il existe au moins un arbre d'espèces *MD*-consistant avec eux. Le problème d'inférence d'arbre d'espèces optimal pour un coût de duplication ou de mutation est NP-difficile.

BIBLIOGRAPHIE

[1] P. Bonizzoni, G. Della Vedova et R. Dondi. Reconciling a gene tree to a species tree under the duplication cost model. *Theoretical Computer Science*, 347:36–53, 2005.

[2] W.C. Chang et O. Eulenstein. Reconciling gene trees with apparent polytomies. Dans D.Z. Chen et D. T. Lee, éditeurs, *Proceedings of the 12th Conference on Computing and Combinatorics (COCOON)*, volume 4112 de *Lecture Notes in Computer Science*, pages 235–244, 2006.

[3] C. Chauve et N. El-Mabrouk. New perspectives on gene family evolution : losses in reconciliation and a link with supertrees. Dans S. Batzoglou, éditeur, *Research in Molecular Biology (RECOMB 2009)*, volume 5541 de *Lecture Notes in Computer Science*, pages 46–58. Springer, 2009.

[4] K. Chen, D. Durand et M. Farach-Colton. Notung : Dating gene duplications using gene family trees. *Journal of Computational Biology*, 7:429–447, 2000.

[5] R. Cole, M. Farach-Colton, R. Hariharan, T. Przytycka et M. Thorup. An $o(n \log n)$ algorithm for the maximum agreement subtree problem for binary trees. *SIAM Journal of Computing*, 30(5):1385-1404, 2000.

[6] J.A. Cotton et R.D.M. Page. Rates and patterns of gene duplication and loss in the human genome. *Proceedings of the Royal Society of London. Series B*, 272: 277–283, 2005.

[7] Andrea Doroftei et Nadia El-Mabrouk. Removing noise from gene trees. Dans *Proceedings of the 11th international conference on Algorithms in bioinformatics*, WABI'11, pages 76–91, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-23037-0. URL `http://dl.acm.org/citation.cfm?id=2039945.2039953`.

[8] O. Eulenstein, B. Mirkin et M. Vingron. Duplication-based measures of difference between gene and species trees. *Journal of Computational Biology*, 5:135– 148, 1998.

[9] Martin Farach, Teresa M Przytycka et Mikkel Thorup. On the agreement of many trees. *Information Processing Letters*, 55(6):297 – 301, 1995.

[10] C.R. Finden et A.D. Gordon. Obtaining common pruned trees. *J. Classification*, 2: 255- 276, 1985.

[11] M. Goodman, J. Czelusniak, G.W. Moore, A.E. Romero-Herrera et G. Matsuda. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Systematic Zoology*, 28:132–163, 1979.

[12] P. Gorecki et J. Tiuryn. DLS-trees : a model of evolutionary scenarios. *Theoretical Computer Science*, 359:378–399, 2006.

[13] R. Guigó, I. Muchnik et T.F. Smith. Reconstruction of ancient molecular phylogeny. *Molecular Phylogenetics and Evolution*, 6:189–213, 1996.

[14] M.W. Hahn. Bias in phylogenetic tree reconciliation methods : implications for vertebrate genome evolution. *Genome Biology*, 8(R141), 2007.

[15] M.W. Hahn, M.V. Han et S.-G. Han. Gene family evolution across 12 *drosophilia* genomes. *PLoS Genetics*, 3 :e197, 2007.

[16] M.T. Hallett et J. Lagergren. New algorithms for the duplication-loss model. Dans R. Shamir, S. Miyano, S. Istrail, P. Pevzner et M. S. Waterman, éditeurs, *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology*, RECOMB, pages 138–146, New York, 2000. ACM.

[17] FELSENSTEIN J. Confidence limits on phylogenies : an approach using the bootstrap. *Evolution*, 39:783–791, 1985.

[18] Li Li, Christian J. Stoeckert et David S. Roos. Orthomcl : Identification of ortholog groups for eukaryotic genomes. 13(9):2178–2189, 2003.

[19] W.H. Li, Z. Gu, H. Wang et A. Nekrutenko. Evolutionary analysis of the human genome. *Nature*, 409:847–849, 2001.

[20] M. Lynch et J.S. Conery. The evolutionary fate and consequences of duplicate genes. *Science*, 290:1151-1155, 2000.

[21] B. Ma, M. Li et L. Zhang. From gene trees to species trees. *SIAM Journal on Computing*, 30:729–752, 2000.

[22] Kevin P. O'Brien, Maido Remm et Erik L. L. Sonnhammer. Inparanoid : a comprehensive database of eukaryotic orthologs. 33(suppl 1):D476–D480, 2005.

[23] R.D.M. Page. Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. *Systematic Biology*, 43:58–77, 1994.

[24] R.D.M. Page. Genetree : comparing gene and species phylogenies using reconciled trees. *Bioinformatics*, 14:819–820, 1998.

[25] R.D.M Page et M.A. Charleston. Reconciled trees and incongruent gene and species trees. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 37:57–70, 1997.

[26] M. Steel et T. Warnow. Kaikoura tree theorems :computing the maximum agreement subtree. *Inform. Process. Lett.*, 48:77-82, 1993.

[27] K.M. Swenson, A. Doroftei et N. El-Mabrouk. Gene tree correction for reconciliation and species tree inference. *Algorithms for Mol. Biol.*, (submitted), 2012.

[28] Roman Tatusov, Natalie Fedorova, John Jackson, Aviva Jacobs, Boris Kiryutin, Eugene Koonin, Dmitri Krylov, Raja Mazumder, Sergei Mekhedov, Anastasia Nikolskaya, B Sridhar Rao, Sergei Smirnov, Alexander Sverdlov, Sona Vasudevan,

Yuri Wolf, Jodie Yin et Darren Natale. The cog database : an updated version includes eukaryotes. *BMC Bioinformatics*, 4(1):41, 2003. ISSN 1471-2105. URL `http://www.biomedcentral.com/1471-2105/4/41`.

[29] Roman L. Tatusov, Eugene V. Koonin et David J. Lipman. A genomic perspective on protein families. 278(5338):631–637, 1997.

[30] B. Vernot, M. Stolzer, A. Goldman et D. Durand. Reconciliation with non-binary species trees. *Journal of Computational Biology*, 15:981–1006, 2008.

[31] L.X. Zhang. On Mirkin-Muchnik-Smith conjecture for comparing molecular phylogenies. *Journal of Computational Biology*, 4:177–188., 1997.

[32] C. M. Zmasek et S. R. Eddy. A simple algorithm to infer gene duplication and speciiation events on a gene tree. *Bioinformatics*, 17:821– 828, 2001.

[33] Christian M. Zmasek et Sean R. Eddy. A simple algorithm to infer gene duplication and speciation events on a gene tree. 17(9):821–828, 2001.

Annexe I

Annexe 1

Removing Noise from Gene Trees

Andrea Doroftei[1] and Nadia El-Mabrouk[2]

[1] DIRO, Université de Montréal, H3C 3J7, Canada, [redacted]
[2] DIRO[redacted]

Abstract. Reconciliation is the commonly used method for inferring the evolutionary scenario for a gene family. It consists in "embedding" an inferred gene tree into a knwo species tree, revealing the evolution of the gene family by duplications and losses. The main complaint about reconciliation is that the inferred evolutionary scenario is strongly dependant on the considered gene tree, as few misplaced leaves may lead to a completely different history, with significantly more duplications and losses. As using different phylogenetic methods with different parameters may lead to different gene trees, it is essential to have criteria to choose, among those, the appropriate one for reconciliation. In this paper, following the conclusion of a previous paper, we flag certain duplication vertices of a gene tree, the "non-apparent duplication" (NAD) vertices, as resulting from the misplacement of leaves, and consider the optimization problem of removing the minimum number of leaves leading to a tree without any NAD vertex. We develop a polynomial-time algorithm that is exact for two special classes of gene trees and show, by simulations, that it is very close to optimality in general.

1 Introduction

Almost all genomes which have been studied contain genes that are present in two or more copies. As an example, duplicated genes account for about 15% of the proteins genes in the human genome [21]. In operational practice, homologous gene copies, e.g. copies in one genome or amongst different genomes that are descendant from the same ancestral gene, are identified through sequence similarity. For example, using a BLAST-like method [1], all gene copies with a similarity score above a certain threshold would be grouped into the same *gene family*. Using a classical phylogenetic method, a *gene tree*, representing the evolution of the gene family by local mutations, can then be constructed based on the similarity scores.

From a functional point of view, grouping genes by sequence similarity is not sufficient to infer a common function for genes. Indeed, it is important to distinguish between two kinds of homologs: *orthologs* which are copies in different species related through speciation, and thus likely to have similar functions, and *paralogs*, which are copies that have evolved by duplication, and more likely to have acquired new functions. Duplication is, indeed, a major source of gene innovation and creation of new functions [24]. In addition, gene losses, arising through the pseudogenization of previously functional genes, also play a key role in the evolution of gene families [3, 9, 10, 12, 19, 22, 24]. Understanding the evolution of gene families through speciation, duplication and loss is thus a fundamental question in functional genomics, evolutionary biology and phylogenomics [28, 30].

The most commonly used methods to infer evolutionary scenarios for gene families are based on the *reconciliation* approach that compares the species tree S (describing the relationships among taxa) to the gene tree T. Indeed, assuming no sequencing errors and a "correct" gene tree, the incongruence between the two trees can be seen as a footprint of the evolution of the gene family through processes other than speciation, such as duplication and loss. The concept of reconciling a gene tree to a species tree under the duplication-loss model was pioneered by Goodman [15] and then widely accepted, utilized and also generalized to models of other processes, for example horizontal gene transfer [20]. Several definitions of reconciliation exist in the literature, one of them expressed in term of "tree extension" [6]. More precisely, a *reconciliation* R between T and S is an extension of

T (obtained by grafting new subtrees onto existing branches of T) *consistant* with the species tree, i.e. reflecting the same phylogeny. A duplication and loss history for the gene family is then directly deduced from R. As many reconciliations exist, a natural approach is to select the one that optimizes a given criterion. Natural combinatorial criteria are the number of duplications (duplication cost), losses (loss cost) or both combined (mutation cost) [7, 23]. The so called Lowest Common Ancestor (LCA) mapping between a gene tree and a species tree, formulated in [17, 27] and widely used [4, 11, 13, 16, 23, 25–27, 31], defines a reconciliation that minimizes both the duplication and mutation costs.

The main complaint about reconciliation methods is that the inferred duplication and loss history for a gene family is strongly dependant on the gene tree considered for this family. Indeed, a few misplaced leaves in the gene tree can lead to a completely different history, possibly with significantly more duplications and losses [18]. Reconciliation can therefore inspire confidence only in the case of a well-supported gene tree. Typically bootstrapping values are used as a measure of confidence in each edge of a phylogeny. How should the weak edges of a gene tree be handled? A strategy adopted in [7] is to explore the space of gene trees obtained from the original gene tree T by performing Nearest Neighbor Interchanges (NNI's) around weakly-supported edges. The problem is then to select, from this space, the tree giving rise to the minimum reconciliation cost.

In this paper, we explore a different strategy for correcting a gene tree that consists in removing a minimum number of "misleading" gene copies. Criteria for identifying, in the gene tree, potentially misplaced leaves were given in a previous paper [6], where "non-apparent duplication vertices", were flagged as potentially resulting from the misplacement of leaves in the gene tree. The reason is that each one of these vertices x reflects a phylogenetic contradiction with the species tree. More precisely, there is at least one triplet of species (a, b, c) that has a different phylogeny on the subtree of T rooted at x than in S. We develop algorithmic methods for removing the minimum number of leaves allowing to obtain a gene tree T without any non-apparent duplication vertex.

In the next section, we begin by formally introducing our concepts. We then motivate and state our problem in Section 3. Section 4 is dedicated to the algorithmic developments. We first describe two special classes of gene trees which lead to an exact polynomial-time algorithm. We then present a heuristic algorithm for the general case. In Section 5, we test the optimality of our algorithm, and the ability of the presented approach to identify misplaced genes. We finally conclude in Section 6 and identify our short term perspectives for improvements.

2 Definitions

2.1 Trees

Let $\mathcal{G} = \{1, 2, \cdots, g\}$ be a set of integers representing g different species (genomes). A ***species tree*** on $\mathcal{G}$ is a binary tree with exactly g leaves, where each $i \in \mathcal{G}$ is the label of a single leaf (Figure 1.(a)). A ***gene tree*** on $\mathcal{G}$ is a binary tree where each leaf is labeled by an integer from $\mathcal{G}$, with possibly repeated leaves (Figure 1.(b)). A gene tree represents a gene family, where each leaf labeled i represents a gene copy located on genome i. In the case of a species tree or a *uniquely leaf-labeled gene tree*, i.e. no leaf-label occurs more than once, we will make no difference between a leaf and its label.

Given a tree U, the ***size of*** U, denoted $|U|$, is the number of leaves of U, and the ***genome set of*** U, denoted by $\mathcal{G}(U)$, is the subset of $\mathcal{G}$ defined by the labels of the leaves of U. Given a vertex x of U, U_x is the subtree of U rooted at x, and the ***genome set of*** x, denoted by $\mathcal{G}(x)$, is the subset of $\mathcal{G}$ defined by the labels of the leaves of U_x (for example, in the tree of Figure 1.(a), $\mathcal{G}(B) = \{1, 2\}$). If x is not a leaf, we denote by x_l and x_r the two children of x. Finally, if x is not the root, any vertex y on a path from x to the root is an *ancestor* of x.

Given a tree U, a ***leaf removal*** consists in removing a given leaf i from U, and its parent vertex. A tree U' obtained from U through a sequence of leaf removals is said to be ***included in*** U.

Finally, a subtree U_x of U, for a given vertex x, is said to be a ***maximum subtree*** of U verifying a given property P iff U_x verifies property P and, for any vertex y that is an ancestor of x, U_y does not verify property P.

2.2 Reconciliation

Applying a classical phylogenetic method to the sequences of a family of genes leads to a gene tree T that is different from the species tree, mainly due to the presence of multiple gene copies in T, and that may reflect a divergence history different from S. The reconciliation approach consists in "embedding" the gene tree into the species tree, revealing the evolution of the gene family by duplications and losses.

There are several definitions of reconciliation between a gene tree and a species tree [4, 11, 16, 17, 23, 25, 27]. Here we define reconciliation in terms of subtree insertions, following the notation used in [5, 16]. We begin by introducing some definitions:

- A *subtree insertion* in a tree T consists in grafting a new subtree onto an existing branch of T.
- A tree T' is said to be an *extension* of T if it can be obtained from T by subtree insertions on the branches of T.
- The gene tree T is said to be *DS-consistent with* S (DS holding for Duplication/Speciation) if T reflects a history with no loss, i.e. if for every vertex t of T such that $|\mathcal{G}(t)| \geq 2$, there exists a vertex s of S such that $\mathcal{G}(t) = \mathcal{G}(s)$ and one of the two following conditions holds:
  (D) either $\mathcal{G}(t_r) = \mathcal{G}(t_\ell)$ (indicating a Duplication),
  (S) or $\mathcal{G}(t_r) = \mathcal{G}(s_r)$ and $\mathcal{G}(t_\ell) = \mathcal{G}(s_\ell)$ (indicating a Speciation).

Definition 1. *A* **reconciliation** *between a gene tree T and a species tree S on $\mathcal{G}$ is an extension $R(T, S)$ of T that is DS-consistent with S.*

For example, the tree of Figure 1.(c) is a reconciliation between the gene tree T of Figure 1.(b) and the species tree of Figure 1.(a). Such a reconciliation between T and S implies an unambiguous evolution scenario for the gene family, where a vertex of $R(T, S)$ that satisfies property (D) represents a duplication (duplication vertex), a vertex that satisfies property (S) represents a speciation (speciation vertex), and an inserted subtree represents a gene loss. The number of duplication vertices of $R(T, S)$ is called the *duplication cost* of $R(T, S)$.

2.3 LCA Mapping

The LCA mapping between T and S, denoted by M, maps every vertex t of T towards the Lowest Common Ancestor (LCA) of $\mathcal{G}(t)$ in S. A vertex t of T is called a ***duplication vertex*** of T with respect to S if and only if $M(t_\ell) = M(t)$ and/or $M(t_r) = M(t)$ (see Figure 1.(b)). We denote by $\mathbf{d}(\mathbf{T}, \mathbf{S})$ the number of duplication vertices of T with respect to S.

This mapping induces a reconciliation $M(T, S)$ between T and S, where an internal vertex t of T leads to a duplication vertex in $M(T, S)$ iff t is a duplication vertex of T with respect to S. In other words, the duplication cost of $M(T, S)$ is $d(T, S)$ (see for example [4, 23, 25] for more details on the construction of a reconciliation based on the LCA mapping). Moreover, $M(T, S)$ is a reconciliation that minimizes all of the duplication, loss and mutation costs [6, 16]. In particular, $d(T, S)$ is the minimum duplication cost of any reconciliation between T and S.

2.4 Duplication Vertices and MD-trees.

Let T be a gene tree and S be a species tree. It is immediate to see that any vertex t of T such that $\mathcal{G}(t_\ell) \cap \mathcal{G}(t_r) \neq \emptyset$ (i.e. the left and right subtrees rooted at t contain a gene copy in the same

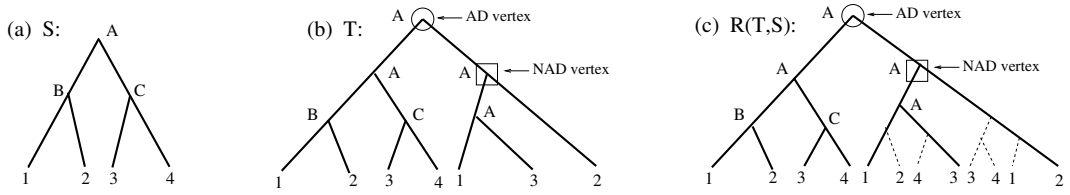

Fig. 1. (a) A species tree S for $\mathcal{G} = \{1, 2, 3, 4\}$. The three internal vertices of S are named A, B and C; (b) A gene tree T. A leaf label x indicates a gene copy in genome x. Internal vertices' labels are attributed according to the LCA mapping between T and S. Flagged vertices are duplication vertices of T with respect to S (Section 2.3); (c) A reconciliation $R(T, S)$ of T and S. Dotted lines represent subtree insertions. The correspondence between vertices of $R(T, S)$ and S is indicated by vertices' labels. Flagged vertices are duplication vertices. All other internal vertices of $R(T, S)$ are speciation vertices. This reconciliation reflects a history of the gene family with two gene duplication preceding the first speciation event, and 4 losses.

genome) will always be a duplication vertex in any reconciliation between T and S, in particular in $M(T, S)$. Such a vertex is called an ***apparent duplication vertex*** (***AD vertex*** for short) of T. In the tree of Figure 1.(b), the root is an AD vertex as its left and right subtree both contain a gene copy in genome 1. Following our notations in [6], we say that T is a *Minimum-Duplication tree consistent with* S, or equivalently a ***tree that is MD-consistent with*** S, iff the duplication cost $d(T, S)$ is equal to the number of apparent duplications of T. In other words, all duplication vertices of T with respect to S are AD vertices.

However, this is not always true, in other words, a duplication vertex of T with respect to S is not necessarily an AD vertex. We call such a duplication vertex that is not an AD vertex a ***non-apparent duplication vertex***, or simply a ***NAD vertex***. For example, the tree of Figure 1.(b) contains one NAD vertex, indicated by a square, and thus T is not MD-consistent with S.

3 Motivation and Problem Statement

Non-apparent duplication vertices point to disagreements between a gene tree and a species tree that are not due to the presence of repeated leaf labels, i.e. multiple copies in the same genome. More precisely, we say that a vertex x of T *splits* three species $\{a, b, c\}$ into $\{a, b; c\}$ if the genome set of one of its children contains a and b but not c, and the genome set of its other child contains c but neither a nor b. Then for any NAD x of T, there is a triplet of species $\{a, b, c\}$ that are split differently by x and by the LCA mapping of x in S. For example, in Figure 1, the triplet of species $\{1, 2, 3\}$ is split into $\{1, 3; 2\}$ by the NAD vertex of T, and into $\{1, 2; 3\}$ by the vertex A in S. It has therefore been suggested that NAD vertices may point at gene copies that are erroneously placed in the gene tree.

Different observations made in [6] tend to support this hypothesis. In particular, using simulated datasets based on the species tree of 12 *Drosophila* species given in [19] and a birth-and-death process, starting from a single ancestral gene, and with different gene gain/loss rates, it has been found that 95% of gene duplications lead to an AD vertex.

Notice however that a misplaced gene in a gene tree T, in other words, a gene randomly placed in T, does not necessarily lead to a NAD vertex. In other words, NAD vertices can only point to a subset of misplaced leaves. However, in the context of reconciliation, the additional damage caused by a misplaced leaf leading to a NAD vertex is the fact that it significantly increases the real mutation-cost of the tree, as shown in Figure 2.

Following the later observations, we exploit the properties of NAD vertices for gene tree correction. If T is not MD-consistent with S, then a tree that is MD-consistent with S can always be obtained from T by performing a certain number of leaf removals. Indeed, a gene tree with only two

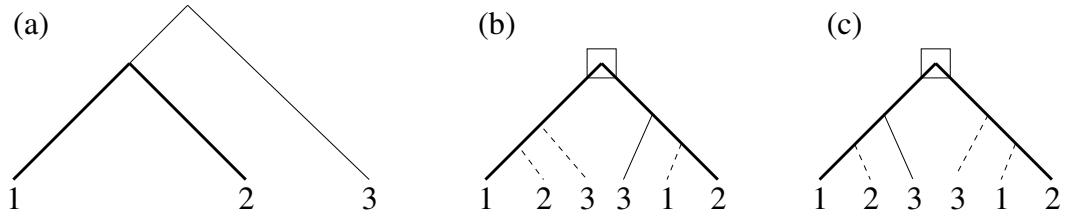

Fig. 2. Let $S = ((1,2),3)$ (the tree in (a)) be the phylogenetic tree for the three species $\{1,2,3\}$. Let $T = (1,2)$ be a gene tree. (a), (b) and (c) are the three possibilities for T after a random insertion of a leaf labeled 3. (a) is the only case leading to a tree without any NAD vertex. It reflects a history of the three gene copies without any duplication or loss; (b) and (c) each contains a NAD vertex, and can be explained by a duplication-loss history of minimum mutation cost of 5: 1 duplication and 4 losses.

leaves is always MD-consistent with any species tree. The optimization problem considered in this paper is therefore the following:

Minimum Leaf Removal Problem:
Input: A gene tree T on $\mathcal{G}$ and a species tree S for $\mathcal{G}$;
Output: A tree T^{MAX} included in T and MD-consistent with S of maximum size (i.e. obtained form T by a minimum number of leaf removals).

4 Method

In the rest of this section, we assume that the set of genomes $\mathcal{G}$ and the species tree S for $\mathcal{G}$ are fixed.

Let T be a gene tree for a gene family on $\mathcal{G}$. We suppose that T is not an MD-tree consistent with S, in other words there is at least one duplication vertex of T that is a NAD vertex. We begin by describing special classes of gene trees for which exact polynomial-time algorithms have been developed.

4.1 Uniquely leaf-labeled gene trees

When the considered gene family contains at most a unique gene copy per genome, the gene tree T is uniquely leaf-labeled. In this case, minimizing the number of leaves that should be removed from T to obtain an MD-tree consistent with S is equivalent to finding the maximum number of genes that lead to the same phylogeny in T and S. In other words, it is immediate to see that the Minimum Leaf Removal Problem reduces, in this case, to the Maximum Agreement Subtree Problem given below.

Maximum Agreement Subtree Problem:
Input: A uniquely leaf-labeled gene tree T on $\mathcal{G}$ and a species tree S for $\mathcal{G}$;
Output: A tree T^{MAX} included in T and MD-consistent with S of maximum size.

A more general definition is given in the literature, where the Maximum Agreement Subtree is defined on a set of uniquely leaf-labeled trees as the largest tree included in each tree of the set. This definition is equivalent to ours in the case of a gene tree T and a species tree S,

The Maximum Agreement Subtree Problem (or MAST problem for short) arises naturally in biology and linguistics as a measure of consistency between two evolutionary trees over species or languages, respectively [8]. In the evolutionary study of genomes, usually different methods and different gene families are used to infer a phylogenetic tree for a set of species, usually yielding to different trees. In such context, one have to find a consensus of the various obtained trees. The

MAST is one method of arriving at such a consensus. The MAST problem was introduced by Finden and Gordon [14]. Amir *et al.* showed that computing a maximum agreement subtree of three trees with unbounded degree is NP-hard [2]. However, in the case of two binary trees (which is the case of interest in this paper), the problem is polynomial. The first polynomial-time algorithm for this problem was given by Steel and Warnow [29]; it had a running time of $O(n^2)$. Later, Cole *et al.* [8] developed an $O(n \log n)$ time algorithm, which, as far as we know, is the most efficient algorithm for solving the MAST problem on two binary trees.

4.2 No AD above NAD

In this section, we consider a tree T containing no AD vertex above a NAD vertex (Figure 3.(a)). More precisely, T satisfies CONSTRAINT C below:

CONSTRAINT C: For each NAD vertex x of T, if y is an ancestor of x that is a duplication vertex, then y is a NAD vertex.

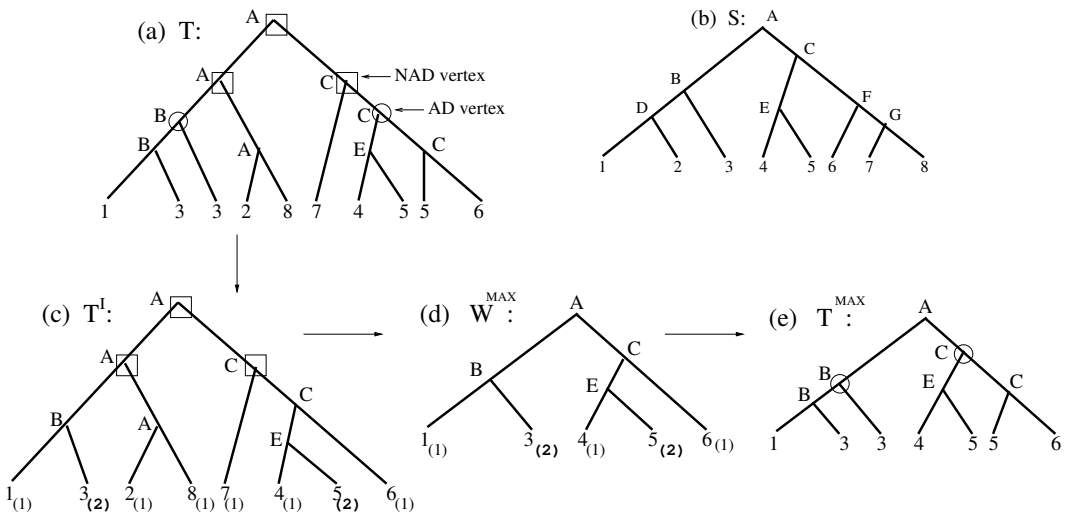

Fig. 3. Solving the MINIMUM LEAF REMOVAL PROBLEM for a tree satistying CONSTRAINT C; (a) A gene tree T on $\mathcal{G} = \{1, 2, 3, 4, 5, 6, 7, 8\}$; (b) A species tree S for $\mathcal{G}$. Internal vertices of S are identified with different characters. Labels of internal vertices of T are attributed according to the LCA mapping between T and S. T contains 5 duplication vertices with respect to S: two AD vertices (surrounded by a circle) and three NAD vertices (surrounded by a square); (c) The tree T^I obtained by replacing the two subtrees of T rooted at each of the two AD vertices by their weighted induced trees. Leaves' weights are given in brackets; (d) The weighted agreement subtree W^{MAX} of T^I and S of maximum value. $v(W^{MAX}) = 6$; (e) The subtree T^{MAX} of T induced by W^{MAX}. T^{MAX} is an MD-tree consistent with S.

We show, in what follows, that the MINIMUM LEAF REMOVAL PROBLEM reduces, in this case, to a "generalization" of the MAXIMUM AGREEMENT SUBTREE PROBLEM to weighted trees, where a *weighted tree* is a uniquely leaf-labeled tree with weighted leaves.

Definition 2. *Let U be a tree on $\mathcal{G}$. The* weighted tree U^I induced by (U, S) *is the tree included in S obtained from S by removing all leaves that are not in $\mathcal{G}(U)$, such that a weight is attributed to*

each leaf s, representing the number of occurrences of s in U (i.e. the number of leaves of U labeled s).

Let $T_1, T_2, \cdots T_m$ be the maximum subtrees of T rooted at an AD vertex (i.e. subtrees of T rooted at the highest AD vertices). Then, the tree T^I obtained by replacing each T_i, for $1 \leq i \leq m$, by the weighted tree T_i^I induced by (T_i, S), is a uniquely leaf-labeled tree, i.e. a weighted tree (Figure 3.(c)). This result directly follows from the fact that T satisfies Constraint C. Let ρ_s be the leaf removal operation that creates the subtree obtained by removing the weighted leaf s from T^I. Then the *corresponding removals* in T consists in removing from T all leaves labeled s.

Finally, we formulate the generalization of the MAST problem to weighted trees as follows, where the value of a weighted tree W is the sum of its leaves' weights.

Maximum Weighted Agreement Subtree Problem:
Input: A weighted tree W on $\mathcal{G}$ and a species tree S for $\mathcal{G}$;
Output: A weighted tree W^{MAX} included in W and MD-consistent with S of maximum value.

We are now ready for the main theorem.

Theorem 1. *Let T be a gene tree satisfying* Constraint C. *Let W^{MAX} be a solution of the* Maximum Weighted Agreement Subtree Problem *on T^I and S, and T^{MAX} be the subtree of T induced by W^{MAX}. Then T^{MAX} is a solution of the* Minimum Leaf Removal Problem *on T and S.*

A complete example of the algorithmic methodology used for solving the Minimum Leaf Removal Problem on T and S inspired by Theorem 1 is given in Figure 3.

In other words, Theorem 1 states that solving the Minimum Leaf Removal Problem on T is equivalent to solving the Maximum Weighted Agreement Subtree Problem on T^I. The proof of Theorem 1 is subdivided into the proofs of the two following lemmas.

Lemma 1. *The tree T^{MAX} is MD-consistent with S.*

Proof. We show, by contradiction, that T^{MAX} does not contain any NAD vertex. Suppose that T^{MAX} contains a NAD vertex x. Then x maps to the same vertex s of S than one of its child, let say the left child. Then there exist two leaves of $T_{x_l}^{MAX}$, labeled a and b, and one leaf of $T_{x_r}^{MAX}$ labeled c such that the triplet (a, b, c) exhibits a wrong phylogeny. As a non-duplication vertex in T can not become a duplication vertex after leaf removals, we have only two possibilities for x in T:

1. *x is a NAD vertex in T.* Then the genome sets of T_{x_l} and T_{x_r} are disjoint. Moreover, the genome set of $W_{x_l}^{MAX}$ (respec. $W_{x_r}^{MAX}$) is a subset of the genome set of T_{x_l} (respec. T_{x_r}). On the other hand, as x is not a duplication vertex in W^{MAX}, one of the three genes a, b and c should be adsent in W_x^{MAX}. And thus, $\{a, b, c\}$ can not be a subset of the genome set of T_x^{MAX}: contradiction.
2. *x is an AD vertex in T.* Then the subtree T_x of T rooted at x contains at least two leaves labeled with the same label d (different from a, b and c), one in T_{x_l} and one in T_{x_r}. Moreover the leaf labeled d in S should belong to the subtree of S rooted at s, and thus to the subtree S_i rooted at the left or right child of s. Such subtree S_i contains at least one leaf labeled a or b or c.
   On the other hand, let y be the parent of x in T^I. As an optimal solution of the Maximum Weighted Agreement Subtree Problem on T^I removes leaves from the subtree T_x^I, such operation should result in removing the duplication vertex y. In other words, x and y should map to the same vertex s in S. Moreover the result of the leaf removal from T_x^I should result in a different LCA mapping for x and y. Indeed, otherwise removing leaves from the corresponding subtree in T^I does not contribute to eliminate any NAD from T^I. It follows that S should exhibit the phylogeny $((a, b, c); d)$, which is a contradiction with the result of the last paragraph □

Lemma 2. *Let T' be a tree included in T that is MD-consistent with S. Then $|T'| \leq |T^{MAX}|$.*

Proof. We will show that, for any $s \in \mathcal{G}$, if a leaf i labeled s is removed from T (i.e. i is not a leaf in T'), then all leaves of T labeled s are removed from T.

Suppose this is not the case. Let y be the vertex of T representing the least common ancestor of all leaves labeled s in T. Then y is an AD node. As a leaf i labeled s is removed from T, such removal should contribute in resolving a NAD vertex x of T. From Constraint C, such vertex should be outside the subtree of T rooted at y. Moreover, it should clearly be an ancestor of y (otherwise removing i will have no effect on x).

As x is a NAD vertex, it maps to the same vertex s of S than one of its child, let say the left child. Then, there exist two leaves of T_{x_l} labeled a and b, and one leaf of T_{x_r} labeled c such that the triplet (a, b, c) exhibits a wrong phylogeny. Moreover, as removing leaf i labeled s contributes in solving x, we can assume that $a = s$. However, from our assumption, it remains, in T', a leaf labeled s. Thus: (1) either it remains, in T', at least one leaf labeled b and one leaf labeld c, or (2) all leaves labeled b, or all leaves labeled c are removed. In case (1), the wrong phylogeny exhibited by the triplet (a, b, c) is still present, preventing vertex x from being a non-duplication vertex. In case (2), as all copies of b (or equivalently c) are removed, there is no need of removing leaf i labeled s for correcting the wrong phylogeny exhibited by the triple (a, b, c).

Therefore, the weighted tree W' induced by T' is obtained from T^I through a sequence of leaf removals. Now, as W^{MAX} is the solution of the Maximum Weighted Agreement Subtree Problem on T^I, then $v(W^{MAX}) \geq v(W')$, and thus $|T^{MAX}| \geq |T'|$ □

4.3 An Algorithm for the General Case

In this section, we present a general algorithm, that is exact in the case of a uniquely leaf-labeled gene tree (Section 4.1) or a gene tree satistying Constraint C (Section 4.2), and a heuristic in the general case.

We first introduce preliminary definitions. For a given tree U (weighted or not), consider the two following properties on U:

Property ONLY-NAD: U has no AD nodes;
Property ONLY-AD: U is rooted at an AD node and contains no NAD node.

We define the ***NAD-border*** of U as the set of roots of the subtrees of U verifying Property ONLY-NAD, and the ***AD-border*** of U as the set of roots of the subtrees of U verifying Property ONLY-AD.

Algorithm Correct-Tree (Figure 4) is a recursive algorithm that takes as input a gene tree T and a species tree S, and outputs a number of leaf removals transforming T into a tree that is MD-consistent with S. It proceeds as follows:

• Stop condition - Lines 2 to 4: If T is MD-consistent with S, then no leaf removal is performed, and the algorithm terminates.

• Recurrence Loop - Lines 6 to 13: Resolve all maximum subtrees of T verifying Constraint C as described in Section 4.2, that is:

1. Construct the weighted tree T^I (Lines 6-8);
2. For each root x of a maximum subtree T^I_x of T^I satisfying Constraint C (Line 9), solve the Maximum Weighted Agreement Subtree Problem on T^I_x, which leads to the weighted tree W^{MAX}_x (Line 10), compute the induced tree T_x (Line 11) and store the number of performed leaf removals (Line 12).

```
Algorithm Correct-Tree (T, S)
1.  LeafRemoval=0;
2.  IF T is a tree MD-consistent with S THEN
3.      RETURN(LeafRemoval)
4.  END IF
5.  T^I = T;
6.  FOR ALL x ∈ AD-border(T) DO
7.      Replace T_x^I by its induced weighted tree;
8.  END FOR
9.  FOR ALL x ∈ NAD-border(T^I) DO
10.     W_x^MAX = WMAST(T_x^I);
11.     Replace T_x by the subtree induced by W_x^MAX;
12.     LeafRemoval = LeafRemoval + (v(T_x^I) − v(W_x^MAX));
13. END FOR
14. RETURN(LeafRemoval+Correct-Tree(T, S))
```

Fig. 4. An algorithm that takes as input a gene tree T and a species tree S, and outputs the number of leaf removals "LeafRemoval" performed to transforming T into a tree that is MD-consistent with S.

5 Results

We only test the optimality of Algorithm Correct-Tree in the case of a gene tree satisfying Property AD-above-NAD, i.e. containing at least one AD vertex above a NAD vertex. Indeed, the algorithm is guaranteed to give the optimal solution otherwise (i.e. for trees satisfying the constraints of Section 4.1 or Section 4.2). We compared the number *NbObtained* of leaf-removal obtained by Algorithm Correct-Tree with the number *NbOptimal* obtained by an exact naive algorithm that tries all possible leaf-subset removals. More precisely, if the minimum number of leaf-removal output by Algorithm Correct-Tree is r, then, we try all subsets of $r-1$, $r-2$ $\cdots$ $r-i$ leaf removals, and stop as soon as a tree that is MD-consistent with S is obtained. As the naive algorithm has clearly an exponential-time complexity, tests are performed on trees of limited size.

We considered a genome set of fixed size, and gene trees with 6 to 24 leaves. For each size s (from 6 to 24 with steps of 2), we performed 500 runs, each with a random species tree S of size 5, and a random gene tree T of size s. Results in Figure 5 are obtained by averaging, for each size s, the results obtained for gene trees satisfying Property AD-above-NAD. The left diagram of Figure 5 shows that Algorithm Correct-Tree gives an exact solution for more than 65% of the trees. Moreover, when *NbOptimal* differs from *NbOptimal*, in mot cases the difference is 1. The right diagram shows that the error-rate, computed as $(NbObtained - NbOptimal)/NbObtained$, is independant from the size of the tree, and does not exceed 0.15. After testing other dependency factors (unshown results), it appears that the error-rate only depends on the number of times the loop 9- 13 of Algorithm Correct-Tree is executed, which is not directly related to the number of NADs or ADs in the tree.

Finally, we tested the ability of the presented approach to identify misplaced genes. To do so, we considered a genome set of fixed size 10, and gene trees of size s varying from 10 to 100 (with a step of 10). For a random species tree S of size 10, and a random tree T MD-consistent with S of size s, we incorporate randomly $NbAdded = (s.10)/100$ leaves with randomly chosen labels. We then test how many "misplaced" leaves our method is able to detect. For each size s, results are averaged over 100 trees. Figure 6 shows the detection percentage of Algorithm Correct-Tree, which is computed as $(NbObtained/NbAdded).100$. This detection percentage decreases with increasing size of the gene tree. This is mainly due to the fact that, as a tree that an MD-consistent tree needs no leaf removal, its detection percentage is always 100%, and that the more leaves we add (1 for a gene tree of size 10, but 10 for a gene tree of size 100) the less chance we have to end up with

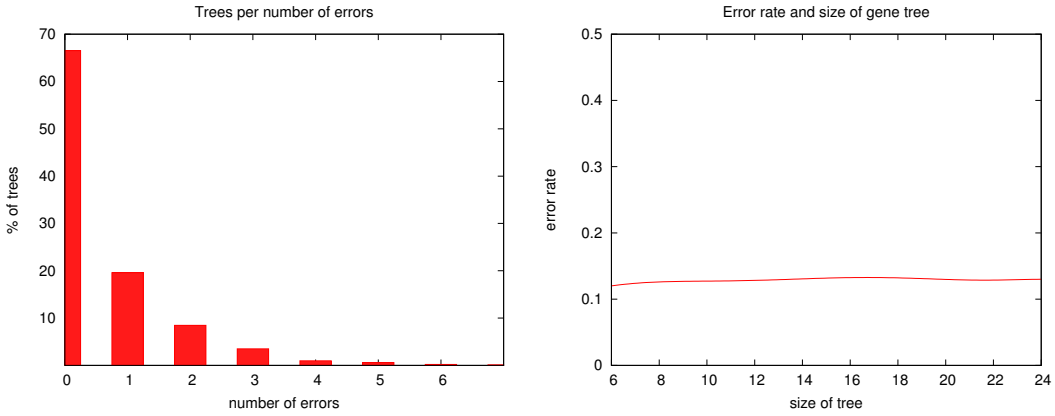

Fig. 5. Comparison of the number *NbObtained* of leaf-removal obtained by Algorithm Correct-Tree with the optimal number *NbOptimal* obtained by an exact algorithm. Left: Percentage of trees leading to a given number $NbObtained - NbOptimal$ of errors (see text for more details on the experiment parameters). Right: The error rate, computed as $(NbObtained - NbOptimal)/NbObtained$, depending on the size of the gene tree (number of leaves).

an MD-consistent tree. Removing the cases of MD-consistent trees lead to a detection percentage around 40%.

6 Conclusion

Based on observations pointing to NAD vertices of a gene tree as indications of potentially misplaced genes, we developed a polynomial-time algorithm for inferring the minimum number of leaf-removals required to transform a gene tree into an MD-tree, i.e. a tree with no NAD vertices. The algorithm is exact in the case of a uniquely leaf-labeled gene tree, or in the case of a gene tree that does not contain any AD vertex above a NAD vertex. Our experimental results show that, in practice, the algorithm is close to optimality in the general case. Unfortunately, NAD vertices can only reveal a subset of misplaced genes, as a randomly placed gene does not necessarily lead to a NAD vertex. Our experiments show that, on average, we are able to infer 40% of misplaced genes. However, the additional damage caused by a misplaced leaf leading to a NAD is an excessive increase of the real mutation-cost of the tree. Therefore, removing NADs can be seen as a preprocessing of the gene tree preceding a reconciliation approach, in order to obtain a better view of the duplication-loss history of the gene family.

Another use of our method would be to choose, among a set of equally supported gene trees output by a given phylogenetic method, the one that can be transformed to an MD-consistent tree by a minimum number of leaf removals.

A limitation of our approach is that a NAD resulting from a wrong bipartition $\{a, b; c\}$ can be, a priori, solved by removing any gene from this bipartition. Our present approach is able to detect a number of misplaced genes but, in general, it is unsufficient to detect precisely the genes that have been erroneously added in the tree. An extension would be to infer all optimal subsets of leaf removals, and to use bootstrapping values on the edges of the tree for a judicious choice of the genes to be removed.

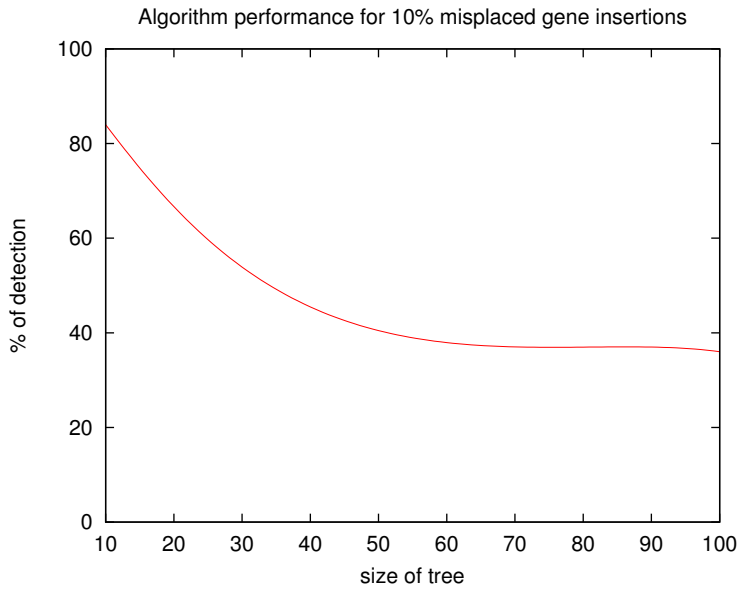

Fig. 6. Percentage of misplaced leaf detection, computed as (*NbObtained*/*NbAdded*).100, where *NbAdded* is the number of randomly added leaves, and *NbObtained* is the number of leaf removals obtained by Algorithm Correct-Tree (see text for more details).

References

1. S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *J.Mol.Biol.*, 215(3):403-410, 1990.
2. A. Amir and D. Keselman. Maximum agreement subtree in a set of evolutionary trees: matrics and efficient algorithms. *SIAM J. Computing*, 26:1656- 1669, 1997.
3. T. Blomme, K. Vandepoele, S. De Bodt, C. Silmillion, S. Maere, and Y. van de Peer. The gain and loss of genes during 600 millions years of vertebrate evolution. *Genome Biology*, 7:R43, 2006.
4. P. Bonizzoni, G. Della Vedova, and R. Dondi. Reconciling a gene tree to a species tree under the duplication cost model. *Theoretical Computer Science*, 347:36–53, 2005.
5. C. Chauve, J.-P. Doyon, and N. El-Mabrouk. Gene family evolution by duplication, speciation and loss. *J. Comput. Biol.*, 15:1043-1062, 2008.
6. C. Chauve and N. El-Mabrouk. New perspectives on gene family evolution: losses in reconciliation and a link with supertrees. In S. Batzoglou, editor, *Research in Molecular Biology (RECOMB 2009)*, volume 5541 of *Lecture Notes in Computer Science*, pages 46–58. Springer, 2009.
7. K. Chen, D. Durand, and M. Farach-Colton. Notung: Dating gene duplications using gene family trees. *Journal of Computational Biology*, 7:429–447, 2000.
8. R. Cole, M. Farach-Colton, R. Hariharan, T. Przytycka, and M. Thorup. An $o(n \log n)$ algorithm for the maximum agreement subtree problem for binary trees. *SIAM Journal of Computing*, 30(5):1385-1404, 2000.
9. J.A. Cotton and R.D.M. Page. Rates and patterns of gene duplication and loss in the human genome. *Proceedings of the Royal Society of London. Series B*, 272:277–283, 2005.
10. J.P. Demuth, T. De Bie, J. Stajich, N. Cristianini, and M.W. Hahn. The evolution of mammalian gene families. *PLoS ONE*, 1:e85, 2006.
11. D. Durand, B.V. Haldórsson, and B. Vernot. A hybrid micro-macroevolutionary approach to gene tree reconstruction. *Journal of Computational Biology*, 13:320–335, 2006.
12. E.E. Eichler and D. Sankoff. Structural dynamics of eukaryotic chromosome evolution. *Science*, 301:793–797, 2003.

13. O. Eulenstein, B. Mirkin, and M. Vingron. Duplication-based measures of difference between gene and species trees. *Journal of Computational Biology*, 5:135– 148, 1998.
14. C.R. Finden and A.D. Gordon. Obtaining common pruned trees. *J. Classification*, 2:255- 276, 1985.
15. M. Goodman, J. Czelusniak, G.W. Moore, A.E. Romero-Herrera, and G. Matsuda. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Systematic Zoology*, 28:132–163, 1979.
16. P. Gorecki and J. Tiuryn. DLS-trees: a model of evolutionary scenarios. *Theoretical Computer Science*, 359:378–399, 2006.
17. R. Guigó, I. Muchnik, and T.F. Smith. Reconstruction of ancient molecular phylogeny. *Molecular Phylogenetics and Evolution*, 6:189–213, 1996.
18. M.W. Hahn. Bias in phylogenetic tree reconciliation methods: implications for vertebrate genome evolution. *Genome Biology*, 8(R141), 2007.
19. M.W. Hahn, M.V. Han, and S.-G. Han. Gene family evolution across 12 *drosophilia* genomes. *PLoS Genetics*, 3:e197, 2007.
20. M. T. Hallett and J. Lagergren. Efficient algorithms for lateral gene transfer problems. In *Proceedings of the Fifth Annual International Conference on Computational Biology (RECOMB '01)*, pages 149–156, New York, 2001. ACM.
21. W.H. Li, Z. Gu, H. Wang, and A. Nekrutenko. Evolutionary analysis of the human genome. *Nature*, 409:847–849, 2001.
22. M. Lynch and J.S. Conery. The evolutionary fate and consequences of duplicate genes. *Science*, 290:1151-1155, 2000.
23. B. Ma, M. Li, and L. Zhang. From gene trees to species trees. *SIAM Journal on Computing*, 30:729–752, 2000.
24. S. Ohno. *Evolution by gene duplication*. Springer, Berlin, 1970.
25. R.D.M. Page. Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. *Systematic Biology*, 43:58–77, 1994.
26. R.D.M. Page. Genetree: comparing gene and species phylogenies using reconciled trees. *Bioinformatics*, 14:819–820, 1998.
27. R.D.M Page and M.A. Charleston. Reconciled trees and incongruent gene and species trees. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 37:57–70, 1997.
28. M.J. Sanderson and M.M. McMahon. Inferring angiosperm phylogeny from EST data with widespread gene duplication. *BMC Evolutionary Biology*, 7:S3, 2007.
29. M. Steel and T. Warnow. Kaikoura tree theorems:computing the maximum agreement subtree. *Inform. Process. Lett.*, 48:77-82, 1993.
30. I. Wapinski, A. Pfeffer, N. Friedman, and A. Regev. Natural history and evolutionary principles of gene duplication in fungi. *Nature*, 449:54–61, 2007.
31. L.X. Zhang. On Mirkin-Muchnik-Smith conjecture for comparing molecular phylogenies. *Journal of Computational Biology*, 4:177–188., 1997.