Université de Montréal

Algorithmes pour la réconciliation d'un arbre de gènes avec un arbre d'espèces

par
Jean-Philippe Doyon

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

Thèse présentée à la Faculté des arts et des sciences
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en informatique

Décembre, 2009

Université de Montréal
Faculté des arts et des sciences

Cette thèse intitulée:

Algorithmes pour la réconciliation d'un arbre de gènes avec un arbre d'espèces

présentée par:

Jean-Philippe Doyon

a été évaluée par un jury composé des personnes suivantes:

Miklós Csűrös,	président-rapporteur
Sylvie Hamel,	directrice de recherche
Hervé Philippe,	codirecteur
Cedric Chauve,	codirecteur
Philippe Langlais,	membre du jury
Alain Denise,	examinateur externe
François-Joseph Lapointe,	représentant du doyen de la FES

Thèse acceptée le: 13 avril 2010

RÉSUMÉ

Une réconciliation entre un arbre de gènes et un arbre d'espèces décrit une histoire d'évolution des gènes homologues en termes de duplications et pertes de gènes. Pour inférer une réconciliation pour un arbre de gènes et un arbre d'espèces, la parcimonie est généralement utilisée selon le nombre de duplications et/ou de pertes. Les modèles de réconciliation sont basés sur des critères probabilistes ou combinatoires.

Le **premier article** définit un modèle combinatoire simple et général où les duplications et les pertes sont clairement identifiées et la réconciliation parcimonieuse n'est pas la seule considérée. Une architecture de toutes les réconciliations est définie et des algorithmes efficaces (soit de dénombrement, de génération aléatoire et d'exploration) sont développés pour étudier les propriétés combinatoires de l'espace de toutes les réconciliations ou seulement les plus parcimonieuses.

Basée sur le processus classique nommé *naissance-et-mort*, un algorithme qui calcule la vraisemblance d'une réconciliation a récemment été proposé. Le **deuxième article** utilise cet algorithme avec les outils combinatoires décrits ci-haut pour calculer efficacement (soit approximativement ou exactement) les probabilités postérieures des réconciliations localisées dans le sous-espace considéré.

Basé sur des taux réalistes (selon un modèle probabiliste) de duplication et de perte et sur des données réelles/simulées de familles de champignons, nos résultats suggèrent que la masse probabiliste de toute l'espace des réconciliations est principalement localisée autour des réconciliations parcimonieuses. Dans un contexte d'approximation de la probabilité d'une réconciliation, notre approche est une alternative intéressante face aux méthodes MCMC et peut être meilleure qu'une approche sophistiquée, efficace et exacte pour calculer la probabilité d'une réconciliation donnée.

Le problème nommé *Gene Tree Parsimony* (GTP) est d'inférer un arbre d'espèces qui minimise le nombre de duplications et/ou de pertes pour un ensemble d'arbres de gènes. Basé sur une approche qui explore tout l'espace des arbres d'espèces pour les génomes

considérés et un calcul efficace des coûts de réconciliation, le **troisième article** décrit un algorithme de *Branch-and-Bound* pour résoudre de façon exacte le problème GTP. Lorsque le nombre de taxa est trop grand, notre algorithme peut facilement considérer des relations prédéfinies entre ensembles de taxa. Nous avons testé notre algorithme sur des familles de gènes de 29 eucaryotes.

Mots clés : Famille de gènes, duplication de gène, perte de gène, homologie, génomique comparative, arbre d'espèces, arbre de gènes, coévolution, probabilité, réconciliation, parcimonie, évolution et phylogénétique.

ABSTRACT

A reconciliation between a gene tree and a species tree depicts an evolutionary scenario of the homologous genes in terms of gene duplications and gene losses. To infer such a reconciliation given a gene tree and a species tree, parsimony is generally used according to the number of gene duplications and/or losses. The combinatorial models of reconciliation are based on probabilistic or combinatorial criteria.

The **first paper** defines a simple and more general combinatorial model of reconciliation which clearly identifies duplication and loss events and does not only induce the most parsimonious reconciliation. An architecture of all possible reconciliations is developed together with efficient algorithms (that is counting, randomization, and exploration) to study combinatorial properties of the space of all reconciliations or only the most parsimonious ones.

Based on the classical birth-death process, an algorithm that computes the likelihood of a reconciliation has recently been proposed. The **second paper** uses this algorithm together with the combinatorial tools described above to compute efficiently, either exactly or approximately, the posterior probability of the reconciliations located in the considered subspace. Based on realistic gene duplication and loss rates and on real/simulated datasets of fungal gene families, our results suggest that the probability mass of the whole space of reconciliations is mostly located around the most parsimonious ones. In the context of posterior probability approximation, our approach is a valuable alternative to a MCMC method and can competes against a sophisticated, efficient, and exact computation of the probability of a given reconciliation.

The Gene Tree Parsimony (GTP) problem is to infer a species tree that minimizes the number of duplications and/or losses over a set of gene family trees. Based on a new approch that explores the whole species tree space for the considered taxa and an efficient computation of the reconciliation cost, the **third paper** describes a Branch-and-Bound algorithm that solves exactly the GTP problem. When the considered number of

taxa is too large, our algorithm can naturally take into account predefined relationships between sets of taxa. We test our algorithm on a dataset of eukaryotic gene families spanning 29 taxa.

Keywords: Gene family, gene duplication, gene loss, homology, comparative genomics, species tree, gene tree, coevolution, probability, reconciliation, parsimony, evolution and phylogenetic.

TABLE DES MATIÈRES

LISTE DES FIGURES

REMERCIEMENTS

Je remercie Cedric Chauve, Sylvie Hamel et Hervé Philippe pour m'avoir donné la possibilité de faire un doctorat dans le domaine de la bio-informatique et pour m'avoir encouragé durant ces longues années. Je remercie aussi Cedric pour les deux séjours de recherche à Vancouver et pour m'avoir encadré dans mes travaux.

Merci aussi à mes parents pour avoir fait ce que je suis et m'avoir soutenu durant toutes mes études. Un gros merci à tous les membres du labo : Denis, Mathieu (he oui ! ! !), Olivier, Tamas, Sébastien, etc. Surtout un gros merci à Denis pour avoir déposé ma thèse (ha ha ! ! !) et pour les nombreuses bières que nous avons bues ensemble. Aussi à Laure pour son amitié et pour m'avoir enfin convaincu d'aller au Robin. Ha oui, un gros merci à Véro pour son amitié, son écoute et les bons restos et le vin. Merci aussi à Manu pour m'avoir enduré comme colocataire et à Naiara pour les années passées ensemble.

Merci aussi à mes nouveaux collègues du LIRMM, surtout à Fabio et Ralouca pour m'avoir enduré durant ce pénible mois de janvier. Ha ha ! ! !

CHAPITRE 1

BIOLOGIE ET BIO-INFORMATIQUE

1.1 Génomique évolutive

Toute espèce est caractérisée par son génome dans lequel on trouve son matériel génétique qui la différencie des autres espèces. Celui-ci contient notamment un ensemble de gènes, chacun étant un segment d'acide désoxyribonucléique (ADN) formé d'une séquence ordonnée d'acides nucléiques : l'adénine (A), la guanine (G), la thymine (T) et la cytosine (C). Un gène codant sert de modèle à la synthèse d'une protéine par un processus biologique qui comprend deux étapes : la transcription, où l'ARN [1] polymérase crée une copie nommée ARN messager (ARNm) du gène, et la traduction, où l'ARN de transfert (ARNt) et les ribosomes décodent les acides nucléiques de l'ARNm en acides aminés (20 possibilités). Une fois que les protéines ont été fabriquées par ce processus de synthèse, elles interagissent et forment une deuxième structure biologique nommée réseau de protéines, celui-ci est propre à toute espèce et définit ses propriétés fonctionnelles.

La *sélection naturelle* est le processus à partir duquel quelques individus d'une espèce vivent jusqu'à l'âge de la maturité sexuelle et réussissent à reproduire de nouveaux individus pour la génération suivante. Si cette nouvelle génération est suffisamment grande, ce processus de sélection est répété soit jusqu'à l'extinction de l'espèce, soit jusqu'à nos jours, ayant ainsi pour résultat une espèce contemporaine. Si des populations d'une même espèce sont isolées durant une période suffisamment longue pour limiter l'échange de matériels génétiques, ici les barrières géographiques ont un rôle important, cette espèce va donner naissance à de nouvelles espèces. Ce processus est nommé *spéciation* et est le principal acteur de l'histoire d'évolution d'une espèce ances-

[1] L'acide ribonucléique.

trale ayant donnée naissance aux espèces contemporaines.

Un problème fondamental en génomique évolutive est de connaître les relations évolutives entre différents organismes (c'est-à-dire des espèces contemporaines) et ce domaine est nommé phylogénie [50]. L'objectif est d'inférer un arbre qui indique les relations de parenté entre différentes espèces contemporaines et leurs espèces ancestrales (représentant des événements de spéciation). Un exemple d'un tel arbre est donné à la figure 1.1.

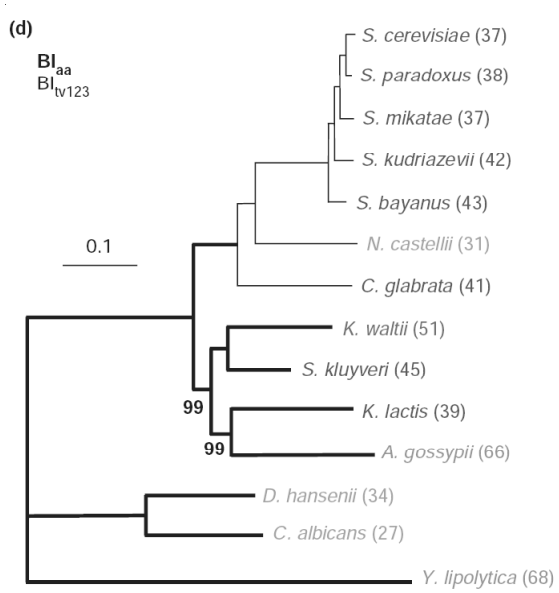

Figure 1.1: Phylogénie de 14 levures (figure importée de [50]).

Les principaux mécanismes influançant l'évolution du génome agissent soit au niveau de sa séquence moléculaire, soit au niveau de ses gènes. La *mutation* remplace un nucléotide d'une position donnée par un autre et les mécanismes agissant au niveau des gènes sont définis aux sections 1.1.1 et 1.1.2.

1.1.1 Duplication de gènes

Cette section décrit les trois principaux processus de duplication de gènes et les devenirs des deux copies. Ces mécanismes se différencient principalement par la taille de

la séquence dupliquée et l'implication ou non d'une molécule d'ARN [96, 49].

Lors du processus de division cellulaire nommé *méiose*, la recombinaison entre deux "chromatides non sœurs" mal appariées peut produire une duplication en tandem sur un des chromosomes et supprimer le fragment d'ADN correspondant sur l'autre chromosome. Un aspect important de ce mécanisme, nommé *recombinaison inégale*, est que les introns des gènes (séquence nucléique chez les eucaryotes pour la traduction d'un gène en protéine) seront présents dans les gènes copies à condition qu'ils l'étaient dans la séquence dupliquée. Un deuxième mécanisme de duplication est nommée *rétroposition*, où l'ARN messager est inversement transcrit en ADN complémentaire puis inséré dans le génome. Contrairement à la recombinaison inégale, ni les introns ni les séquences de régulation sont dupliqués et le gène n'est plus traduit en protéine. Le dernier mécanisme est nommé *duplication complète d'un chromosome* (ou d'un génome) et crée une nouvelle copie pour chaque gène impliqué. L'inconvénient de telles duplications à grande échelle repose sur la fidélité de la transmission du segment dupliqué à la prochaine génération (c'est-à-dire dans la population). Malgré certaines hypothèses d'anciennes duplications complètes de génome au début de l'évolution des vertébrés [68, 88], les signes clairs de tels événements sont débattus [83].

Dans le contexte d'évolution génomique, une des principales causes favorisant la conservation d'une espèces est la création de nouveaux matériels génétiques suite aux duplications de gènes, à partir desquelles de nouvelles fonctions peuvent apparaître par sélection naturelle. Ceci a pour conséquence d'accroître la divergence entre espèces, d'observer l'apparition d'attributs propres à une espèce et d'augmenter la capacité d'adaptation d'une espèce face à son milieu naturel [96]. Suite à une duplication de gène dans un seul génome, la nouvelle copie doit survivre au processus de sélection naturelle afin qu'il soit observable après quelques milliers d'années. Nous décrivons ci-dessous les principaux devenirs pour les deux copies du gène [96].

Immédiatement après une duplication de gène, les deux copies ont la même séquence moléculaire et s'ils sont tous les deux traduits, il y aura un surplus du produit

protéique. Par conséquent, la séquence moléculaire de l'une des copies risque d'être moins contrainte et plus vulnérable aux mutations, la structure du gène en sera négativement affectée, le gène perdra sa fonction et ne sera plus traduit en protéine. Un tel gène est dit *pseudogène* et, après un certain temps d'évolution, sa séquence aura tellement divergée qu'il ne sera plus reconnaissable comme descendant du gène parent (*pseudogénisation*). Une autre possibilité pour les deux copies est qu'elles se partagent les différentes fonctions "originales" du gène père, une telle coopération accroît la capacité d'adaptation de l'organisme par l'élimination de conflits entre ces fonctions [49] (*sous-fonctionnalisation*). Il est aussi possible qu'une des copies du gène soit meilleure dans l'accomplissement de la fonction originale (*spécialisation fonctionnelle*). Un devenir moins fréquent pour les deux copies du gène est que l'une d'elles acquiert une nouvelle fonction généralement similaire à la fonction originale (*néo-fonctionnalisation*). Enfin, si le produit protéique du gène dupliqué est en grande demande chez l'organisme, une dernière alternative est que les deux copies conservent la fonction originale [58].

1.1.2 Autres mécanismes affectant l'évolution des gènes

La *perte* d'un gène survient lorsqu'il devient un pseudogène ou est supprimé par des réarrangements génomiques [41]. Le *transfert horizontal de gène* échange un gène entre un génome dit donneur et un génome dit receveur selon différents mécanismes qui permettent à un ADN étranger d'accéder au génome. Dû à la fréquence des transferts entre bactéries [75], l'existence même d'une topologie de ces espèces est actuellement débattue dans la communauté scientifique [42]. La *désactivation* d'un gène est causée par des substitutions de sa séquence de régulation et l'empêche d'être traduit en protéine. Le processus inverse est nommé *activation* [38]. La *conversion génique* survient lors de la méiose, modifie la séquence d'un gène suite à une erreur durant la recombinaison et est généralement causée par une similarité de séquence entre locus distincts. La *fission* est le processus par lequel un gène est découpé en deux gènes et la *fusion* joint deux gènes adjacents en un seul gène [92].

1.1.3 Familles de gènes homologues

Les gènes appartenant aux espèces actuelles, dont l'histoire d'évolution est représentée par une phylogénie d'espèces (voir la Figure 1.1), ont aussi leur propre histoire, possiblement différente de celle des espèces et à partir desquelles des relations entre les gènes sont définies [35]. En particulier, si deux gènes descendent d'un gène ancestral commun, ils sont dits *homologues*. Ils sont soit *orthologues* si la divergence de leur séquence suit une spéciation, soit *paralogues* si elle suit une duplication, soit *xénologues* si elle suit un transfert. Enfin, la divergence entre deux gènes orthologues suit la spéciation du plus récent ancêtre commun aux deux espèces d'où proviennent ces gènes. Une telle divergence simultanée de cette espèce avec le plus récent ancêtre des deux gènes est nommée *cospéciation* [72]. Ces différentes relations entre gènes (sauf les xénologues) sont représentées à la figure 1.2.

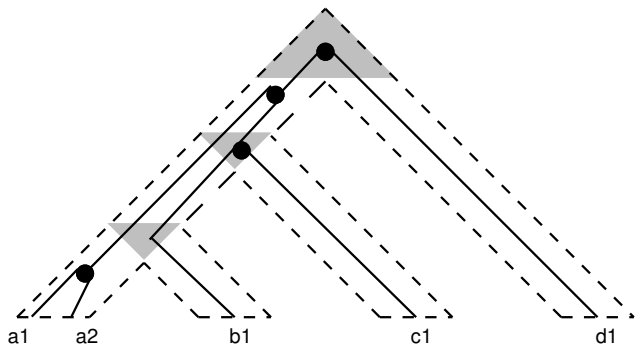

Figure 1.2: Familles de gènes homologues. Les branches de l'arbre d'espèces (resp. gènes) sont représentées par des tubes en pointillés (resp. lignes pleines) et ses nœuds en gris (noir). Un gène est nommé par une lettre correspondant à l'espèce auquel il appartient et d'un indice (les gènes a_1 et a_2 appartiennent à l'espèce A). Les nœuds de cospéciation de l'arbre des gènes sont couverts par un nœud de l'arbre d'espèces et inversement pour les nœuds de duplication.

Les gènes homologues forment une famille et ont généralement des fonctions similaires (métabolisme, olfactifs, contrôle des cycles d'une cellule, etc.). La différence d'un phénotype (caractère observable) entre deux organismes peut parfois s'expliquer par des familles de gènes de tailles différentes. À titre d'exemple, l'efficacité du système visuel de l'humain en comparaison avec celui de la souris est dûe aux duplications répétées

d'un gène particulier (les opsines) et d'une famille de gènes plus grande [49].

Une des principales motivations pour étudier les familles de gènes est de différencier les gènes orthologues des paralogues et xénologues. Des gènes orthologues, dont l'histoire d'évolution correspond à celle des espèces (par définition), ont généralement la même fonction et ils peuvent être utilisés pour prédire la fonction d'un gène (annotation de gène) en se basant sur les fonctions connues des autres orthologues. Ils sont aussi utilisés pour inférer l'arbre phylogénétique des espèces (phylogénomique) et définir une relation de couplage entre les gènes de deux génomes [16]. Ensuite, les famillles de gènes servent à estimer les taux de duplications et de pertes de gènes [14, 44] et établir le contenu en gènes des génomes ancestraux [64].

1.2 Aspects de bio-informatique

Étudier l'histoire d'évolution d'une famille de gènes se fait selon les trois étapes suivantes : regrouper les gènes en familles (section 1.2.1), inférer une phylogénie pour chaque famille (section 1.2.2) et comparer la phylogénie des gènes avec celle des espèces (section 1.3).

1.2.1 Inférer les familles de gènes homologues

Plusieurs méthodes pour regrouper les gènes en familles sont basées sur un graphe de similarité de séquence, où les nœuds correspondent aux séquences moléculaires (nucléiques ou protéiques) et la valuation d'un arc entre deux séquences est définie selon leur degré de similarité. Pour une séquence dite "cible" et un ensemble de séquences donnés en paramètre, une heuristique de comparaison de séquence, nommée BLAST [2], identifie les séquences dont la similarité avec la séquence cible est au-dessus d'un degré minimum. Ce programme est successivement utilisé pour chaque séquence cible et contre toutes les séquences du jeux de données, une comparaison deux-à-deux est ainsi calculée. Ensuite, deux gènes sont reliés par un arc s'ils respectent la règle nommé "Best

Reciprocal Hit" [53] et définie comme suit : parmi toutes les séquences, le premier gène est celui qui est le plus similaire au second et vice-versa. Ce graphe est ensuite utilisé pour regrouper les gènes en familles et nous présentons ci-dessous différentes méthodes proposées.

Le regroupement de Markov [89] [2] est basé sur la simulation d'une marche aléatoire dans le graphe des gènes, où la valeur d'un arc est remplacée par une probabilité de transition entre les deux gènes. Selon un processus de Markov (stochastique), les probabilités des chemins entre toutes paires de nœuds sont calculées en utilisant deux opérations nommées expansion et inflation. L'expansion du graphe calcule la probabilité d'un chemin de longueur deux entre toutes les paires de nœuds et l'inflation accroît la "proximité" (c'est-à-dire la probabilité) de deux nœuds proches au détriment de ceux éloignés. L'algorithme alterne entre ces deux opérations jusqu'à ce que le graphe soit partitionné en amas et qu'il n'existe plus de chemin entre deux amas.

Les méthodes de regroupement dites hiérarchiques [22, 47] fonctionnent de façon itérative en couplant les deux amas les plus proches pour en former un nouveau, où la distance utilisée correspond à la distance minimale ("Single Linkage") ou maximale ("Complete Linkage") entre deux éléments des deux amas. Ces méthodes sont hiérarchiques car elles créent différents niveaux superposés de regroupements. Le regroupement nommé k-moyennes [65, 60], où l'objectif est de partitionner n données en k amas de façon à ce que chaque donnée soit dans l'amas dont la moyenne est la plus proche, est NP-difficile lorsque k n'est pas fixé [67]. Une dernière méthode de regroupement, nommée "Affinity Propagation" [36], repose sur l'idée de rechercher un exemple type pour chaque amas. Pour l'identification des complexes du réseau d'interactions de protéines de *Saccharomyces cerevisiae*, des expériences [91] montrent que cette méthode a moins de succès que le regroupement de Markov.

[2] "Markov Clustering Algorithm"

1.2.2 Inférence phylogénétique

L'inférence d'une phylogénie pour un ensemble de séquences moléculaires débute par aligner [56, 18] les positions homologues, c'est-à-dire les acides qui descendent d'un ancêtre commun. Cet alignement est ensuite donné à une méthode d'inférence phylogénétique afin de reconstruire une topologie.

Les méthodes de distance, notamment le *Neighbour Joining* [78] et *l'Évolution Minimum* [77], infèrent une phylogénie en se basant sur une matrice de distances évolutives calculée à partir de l'alignement des séquences. Selon une phylogénie donnée, la *Parcimonie Maximale* infère les états ancestraux pour chacune des positions homologues de l'alignement et calcule l'arbre pour lequel la somme des substitutions est minimale. Les méthodes probabilistes, soit le *Maximum de Vraisemblance* [33] (MV) et *l'Inférence Bayésienne* [48] (IB), se basent sur un modèle probabiliste d'évolution de séquence [51]. Le MV calcule la topologie qui maximise la vraisemblance, soit la probabilité d'observer les séquences selon cet arbre. Le IB estime les probabilités postérieures des phylogénies en combinant, selon le théorème de Bayes, la fonction de vraisemblance et des probabilités dites *a priori* sur les topologies et les paramètres du modèle d'évolution.

1.3 Introduction à la réconciliation d'un arbre de gènes et d'un arbre d'espèces

Goodman *et al.* [38] sont les premiers à utiliser et décrire le concept de réconciliation pour expliquer l'incongruence entre la phylogénie de vertébrés basée sur des données morphologiques avec l'arbre de gènes des globines. Ici, l'objectif est de trouver l'arbre de gènes G le plus similaire à la phylogénie d'espèces S pour lequel l'histoire d'évolution de G selon S est parcimonieuse selon le nombre total d'événements suivants : mutation, duplication et perte d'un gène, activation et désactivation d'un gène (voir la figure 1.3). Ils ont appliqué leur méthode sur des séquences d'hémoglobines de 28 animaux. L'arbre de gènes calculé par leur heuristique selon la parcimonie classique (seulement avec les mutations) est en désaccord avec plusieurs aspects de la phylogénie

d'espèces, contrairement à celui où les trois catégories d'événements données ci-dessus sont considérées. En particulier, selon cet arbre de gènes et l'arbre d'espèces, ils observent que la plupart des paires de gènes contemporains sont orthologues.

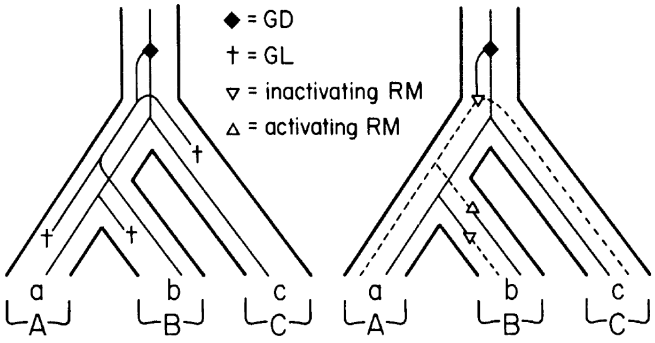

Figure 1.3: Deux réconciliations selon le modèle de [38], où l'arbre de gènes a trois gènes notés a, b et c (idem pour l'arbre d'espèces). Les quatre événements possibles sont représentés comme suit : duplication (⧫), perte (une croix), activation (△) et désactivation (▽) d'un gène. La réconciliation de gauche a une duplication et trois pertes. Celle de droite a une duplication, deux désactivations et une activation de gènes. (Figure importée de [38]).

Dans ce modèle d'évolution d'une famille de gènes, la possibilité d'activer et de désactiver un gène permet de diminuer le nombre minimum d'événements qui doivent être postulés suite à une duplication. Autrement dit, selon un modèle où ni l'activation, ni la désactivation ne sont considérées, l'histoire d'évolution de G selon S est moins parcimonieuse. L'inconvénient du modèle de Goodman *et al.* est qu'il ne prend pas en considération l'intervalle de temps d'évolution, le long de la phylogénie d'espèces, entre la désactivation d'un gène et sa réactivation : une longue période implique beaucoup de mutations et une séquence vraisemblablement non-codante. Enfin, Goodman *et al.* sont les seuls jusqu'à ce jour à considérer ce type d'événements.

1.4 Problèmes étudiés et plan des prochains chapitres

Deux problèmes sont étudiés dans cette thèse. Le premier est de proposer une histoire d'évolution d'un arbre de gènes avec un arbre d'espèces selon un modèle de réconcilia-

tion soit combinatoire, soit probabiliste. Pour un ensemble d'arbres de gènes donné, le deuxième problème est de proposer un arbre d'espèces dont l'ensemble des réconciliations est optimal selon un modèle combinatoire. Nous considérons les aspects suivants pour ces deux problèmes : les arbres de gènes et l'arbre d'espèces sont binaires et enracinés ; les spéciations, les duplications et les pertes sont les événements permis par les modèles de réconciliation.

Les trois prochains chapitres présentent les deux problèmes et les différentes méthodes proposées. Le chapitre 2 décrit les modèles combinatoires de réconciliation, dont celui utilisé pour calculer la probabilité d'une réconciliation. Le modèle probabiliste et les algorithmes pour calculer cette probabilité sont présentés au chapitre 3. Le chapitre 4 introduit le lecteur aux principales méthodes pour inférer un arbre d'espèces selon un ensemble d'arbres de gènes.

Les contributions de cette thèse sont introduites au chapitre 5 et développées aux trois chapitres suivants. Le premier article (chapitre 6) décrit un nouveau modèle combinatoire de réconciliation. Le deuxième article (chapitre 7) donne une méthode efficace, basée sur le contenu des chapitres 3 et 6, pour estimer les probabilités des réconciliations. Le troisième article (chapitre 8) développe une approche de type "Branch-and-Bound" pour inférer un arbre d'espèces selon un ou plusieurs arbres de gènes.

CHAPITRE 2

MODÈLES COMBINATOIRES DE RÉCONCILIATION

Ce chapitre présente les différents modèles combinatoires proposés dans la littérature pour la réconciliation d'un arbre de gènes avec un arbre d'espèces. La section 2.1 donne les définitions basiques utilisées dans cette thèse et la section 2.2 décrit les différents modèles. La section 2.3 introduit nos contributions à un de ces modèles, soit celui d'Arvestad *et al.* [4, 3, 5, 6] qui est une des composantes clés de cette thèse.

2.1 Préliminaires

Excepté lorsque c'est indiqué, tous les arbres considérés dans cette thèse sont binaires et enracinés. Pour un arbre étiqueté T, la racine est notée $r(T)$ et l'ensemble des nœuds, l'ensemble des feuilles et l'ensemble des étiquettes de ses feuilles sont respectivement notés $V(T)$, $L(T)$ et $\Lambda(T)$. L'ensemble des nœuds internes, soit $V(T) \setminus L(T)$, est noté $I(T)$. Les deux fils d'un nœud interne $u \in I(T)$ sont notés u_1 et u_2, le père d'un nœud u (où $u \neq r(T)$) est noté $p(u)$, son frère $s(u)$ et l'arbre enraciné en $u \in V(T)$ est noté T_u. Pour deux nœuds $u, v \in V(T)$, on note $v \leq_T u$ ($v <_T u$) lorsque u est sur le chemin de v à $r(T)$ (resp. et $v \neq u$) et on dit que u est un ancêtre (resp. strict) de v. Pour un nœud $u \in V(T)$, $\Lambda(T_u) \subseteq \Lambda(T)$ est l'ensemble, dit clade, des étiquettes des feuilles de T_u et l'ensemble des clades présents dans T est noté $\mathcal{C}(T)$. Si T est un arbre non-ordonné, le seul fils d'un nœud $u \in I(T)$ qui respecte une contrainte donnée est arbitrairement noté soit u_1, soit u_2. Nous définissons ci-dessous la fonction nommée *Last Common Ancestor* (ou *Lowest Common Ancestor*) d'un arbre T.

Définition 2.1. *Le fonction nommée LCA définie sur un arbre T est notée $lca : V(T) \times V(T) \to V(T)$, où pour $u, v \in V(T)$, $lca(u, v)$ est l'unique nœud $w \in V(T)$ tel que (i) $u \leq_T w$ et $v \leq_T w$ et (ii) aucun descendant strict de w vérifie la propriété (i).*

Dorénavant, G (resp. S) dénote un arbre de gènes (resp. d'espèces) de taille n_G (resp. n_S). G et S sont binaires, enracinés et non-ordonnés. Chaque feuille de S est étiquetée par une espèce (i.e. génome) de l'ensemble $\Lambda(S)$ et chaque espèce de $\Lambda(S)$ étiquette une et une seule feuille de S (il existe une bijection entre $L(S)$ et $\Lambda(S)$). Chaque feuille de G est étiquetée par une espèce de $\Lambda(S)$ et plusieurs feuilles de G peuvent avoir la même étiquette (de $\Lambda(S)$). Autrement dit, deux gènes feuilles de G peuvent provenir d'une même espèce feuille de S. Aussi, nous considérons des arbres G et S où $\Lambda(G) \subseteq \Lambda(S)$. La correspondance entre les gènes feuilles de G et des espèces feuilles de S est définie ci-dessous.

Définition 2.2. *Une fonction de couplage des feuilles de G avec les feuilles de S est une fonction surjective notée $\sigma : L(G) \rightarrow L(S)$, où pour $u \in L(G)$, $\sigma(u)$ est la feuille de S correspondant à l'espèce d'où le gène u provient.*

Il est important de préciser que chaque nœud feuille de G représente un gène contemporain et pour cette raison les termes "gène feuille" et "gène contemporain" sont considérés des synonymes dans cette thèse. Le même principe s'applique pour l'arbre d'espèces S.

2.2 Description des modèles proposés

2.2.1 La réconciliation parcimonieuse

Le concept de réconciliation a premièrement été défini selon la réconciliation dite de parcimonie [71], c'est-à-dire celle qui minimise le nombre total d'événements de duplication et/ou de perte permettant de réconcilier un arbre de gènes G et un arbre d'espèces S. Une réconciliation parcimonieuse est calculée en localisant tous les gènes qui peuvent être une cospéciation selon les contraintes propres au processus de coévolution de G et de S. Dans une telle histoire d'évolution, un nœud interne u de G est une cospéciation lorsque l'évolution de ses deux lignées u_1 et u_2 est conforme à la spéciation d'une espèce x : chaque gène feuille descendant de u_1 (resp. u_2) provient d'une espèce feuille

descendante de x_1 (resp. x_2). Notons aussi que chaque gène feuille $u \in L(G)$ est un nœud de cospéciation avec l'espèce correspondante $\sigma(u)$.

La détection des nœuds de cospéciation se fait selon un couplage des nœuds de G avec les nœuds de S qui est communément appelé le *Couplage LCA* (*Last Common Ancestor mapping*), est noté $M_S : V(G) \to V(S)$ et sera défini ultérieurement. Le nœud $M_S(u) \in V(S)$ associé à un nœud $u \in V(G)$ est l'unique espèce pour laquelle il est possible au gène de correspondre à une cospéciation selon toutes les histoires de coévolution des deux arbres. Si le nœud u a au moins un fils u' tel que $M_S(u) = M_S(u')$, c'est-à-dire que ces deux gènes d'une même lignée sont couplés avec le même événement de spéciation, alors le gène u ne peut être une cospéciaton et est donc une duplication [1]. Un tel nœud de duplication est nommé duplication forcée [27]. La figure 2.1 donne un exemple du couplage LCA.

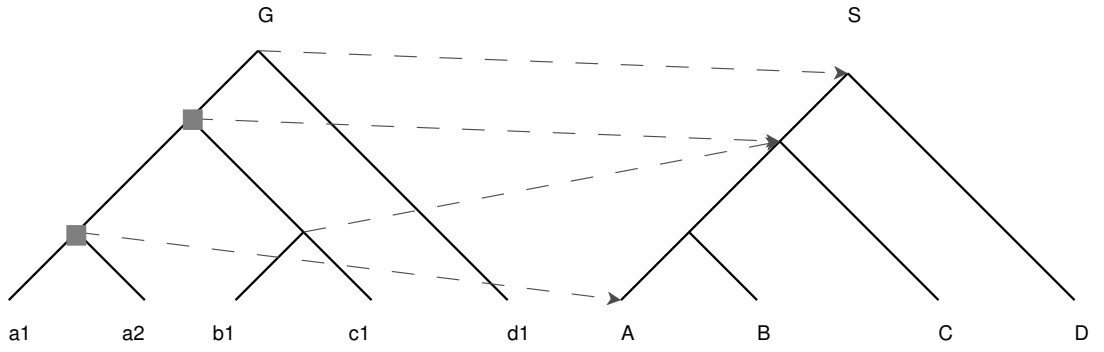

Figure 2.1: Un arbre de gènes G et un arbre d'espèces S reliés par le couplage LCA. Chaque gène est nommé par une lettre correspondant à l'espèce auquel il appartient et d'un indice pour différencier les gènes d'une même espèce (les gènes a_1 et a_2 appartiennent à l'espèce A). Les nœuds de duplication de G sont indiqués en gris et tous les autres sont des cospéciations.

Le couplage LCA $M_S : V(G) \to V(S)$ est décrit comme suit : chaque gène u de G est couplé avec l'espèce $M_S(u) = x$ la plus récente (la plus proche des feuilles de S) de telle sorte que chaque gène contemporain et descendant de u provient d'une espèce contemporaine et descendante de x. Cette fonction entre G et S ainsi que les duplications forcées et les cospéciations qui en résultent sont définies ci-dessous.

[1] Rappelons que nous ne considérons pas les transferts horizontaux de gène.

Définition 2.3. Le couplage LCA, noté $M_S : V(G) \rightarrow V(S)$, couple chaque nœud u de G à l'unique nœud $M_S(u)$ de S tel que $\Lambda(S_{M_S(u)})$ est le plus petit clade de S qui contient $\Lambda(G_u)$.

Définition 2.4. *Un nœud interne $u \in I(G)$ est une duplication forcée si $M_S(u) = M_S(u_1)$ et/ou $M_S(u) = M_S(u_2)$, sinon il est une cospéciation.*

La propriété ci-dessous permet d'expliquer pourquoi la fonction $M_S : V(G) \rightarrow V(S)$ induit une réconciliation parcimonieuse. En particulier, toute fonction de couplage entre G et S qui est utilisée pour définir une histoire de coévolution cohérente des deux arbres doit respecter cette propriété.

Propriété 2.5. *("*Inclusion Preserving Mapping*" [17]). Une fonction $M : V(G) \rightarrow V(S)$ est dite* préserver l'inclusion *lorsque pour chaque paire de nœuds $(u, v) \in V(G) \times V(G)$, si $\Lambda(G_u) \subseteq \Lambda(G_v)$, alors $\Lambda(G_u) \subseteq \Lambda(S_{M(u)}) \subseteq \Lambda(S_{M(v)})$.*

Les raisons pour lesquelles la fonction $M_S : V(G) \rightarrow V(S)$ induit une réconciliation parcimonieuse selon le nombre de duplications et/ou de pertes sont les suivantes : (i) chaque duplication induite est forcée et (ii) parmi l'ensemble des fonctions qui préservent l'inclusion, la distance entre les deux nœuds $M_S(u)$ et $M_S(u')$ est minimale pour chaque nœud $u \in I(G)$ et un de ses fils noté u'.

Plusieurs algorithmes efficaces ont été proposés pour calculer le couplage LCA entre G et S. L'algorithme le plus simple est de type "Brute-force" [71] et a une complexité en temps de $O({n_S}^3)$. Une méthode basée sur des calculs arithméthiques permet de calculer en temps constant le $lca(u, v)$ de deux nœuds u et v d'un arbre T [80], et deux algorithmes [97, 21] utilisent cette méthode afin de calculer le couplage M_S en temps $O(n_G)$. Zmasek et Eddy [98] donnent un algorithme simple et de complexité $O({n_G}^2)$ en temps (assumant que $n_S \in O(n_G)$) et $O(n_G)$ en espace. Celui-ci est basé sur un parcours postfixe de G et sur les observations suivantes : pour un nœud interne u de G, $M_S(u)$ est obligatoirement localisé plus haut que $M_S(u_1)$ et $M_S(u_2)$ dans S et est égal à $lca(M_S(u_1), M_S(u_2))$. Aussi, un algorithme en temps $O(n_G)$ est donné en [30].

Le nombre de pertes et le nombre de duplications induites par la réconciliation parcimonieuse sont calculés selon le couplage LCA entre G et S. Les trois critères combinatoires utilisés pour valuer cette réconciliation parcimonieuse sont définis ci-dessous.

Définition 2.6. Le *coût de duplication* de la réconciliation de parcimonie entre G et S est $d(G,S) = \sum_{u \in I(G)} d(u,S)$, où $d(u,S)$ est 1 si et seulement si $u \in I(G)$ est une duplication forcée et 0 sinon.

Définition 2.7. Le *coût de perte* de la réconciliation de parcimonie entre G et S est $l(G,S) = \sum_{u \in I(G)} l(u,S)$, où $l(u,S)$ est défini comme suit

$$l(u,S) = \begin{cases} 0 & \text{si } M_S(u_1) = M_S(u); \\ & \text{et } M_S(u_2) = M_S(u); \\ d_S(M_S(u_1), M_S(u)) + 1 & \text{si } M_S(u_1) \neq M_S(u) \\ & \text{et } M_S(u_2) = M_S(u); \\ d_S(M_S(u_1), M_S(u)) + d_S(M_S(u_2), M_S(u)) & \text{si } M_S(u_1) \neq M_S(u) \\ & \text{et } M_S(u_2) \neq M_S(u). \end{cases}$$

Définition 2.8. Le *coût de mutation* de la réconciliation de parcimonie entre G et S est $m(G,S) = l(G,S) + d(G,S)$.

Une fois que le couplage $M_S : V(G) \rightarrow V(S)$ est défini, $d(G,S)$ est calculé en temps $\Theta(n_G)$ et, suite à un prétraitement pour déterminer la profondeur de chaque nœud de S, $l(G,S)$ est aussi calculé en temps $\Theta(n_G)$ [97]. Donc, la complexité pour calculer chacun des trois coûts de la réconciliation parcimonieuse dépend du choix de l'algorithme pour calculer le couplage LCA. Enfin, un algorithme complexe calcule $l(G,S)$ en temps $O(n_G)$[63].

Nous avons vu que la réconciliation parcimonieuse n'est formée que de duplications forcées et que ceci est une conséquence du couplage LCA des nœuds de G avec les nœuds de S. Un couplage moins restrictif [73, 17] qui respecte la propriété 2.5 permet de définir une réconciliation plus générale, où les duplications ne sont pas obligatoirement

forcées. Nous verrons dans les prochaines sections différents modèles combinatoires de réconciliation où celle dite de parcimonie n'est pas la seule considérée.

2.2.2 Arbre de réconciliation

Le concept d'arbre de réconciliation a été introduit en [38] et ensuite formellement défini en [71, 63, 17]. Afin de représenter une histoire de coévolution cohérente des arbres G et S, un arbre de réconciliation noté R doit respecter les deux contraintes suivantes : (1) il doit contenir tous et seulement tous les clades de S et (2) G doit être induit de R par le retrait de zero ou plusieurs sous-arbres. La définition formelle d'un arbre de réconciliation est donnée par les définitions 2.9 et 2.10 et la figure 2.2 donne un exemple.

Définition 2.9. *Soit un arbre binaire T et un sous-ensemble de feuilles $K \subseteq L(T)$. L'arbre homomorphique, noté $T|_K$, de T et induit par K est obtenu en considérant le plus petit sous-arbre de T, noté T', tel que $L(T') = K$ et en contractant tous les nœuds de degré deux (excepté la racine).*

Définition 2.10. *Un arbre de réconciliation R entre G et S est un arbre qui respecte les contraintes suivantes :*

1. *G est un sous-arbre homomorphique de R : $\exists K \subseteq L(R)$ tel que $R|_K$ est isomorphe à G ;*
2. *tous les clades de R sont des clades de S ;*
3. *pour chaque nœud interne u de R, $\Lambda(R_{u_1})$ et $\Lambda(R_{u_2})$ sont soit disjoints, soit égaux à $\Lambda(R_u)$.*

L'histoire d'évolution de G selon S représentée par un arbre de réconciliation R est lue de la façon suivante : un nœud interne u de R correspond à une cospéciation si $\Lambda(R_{u_1})$ et $\Lambda(R_{u_2})$ sont disjoints, à une duplication s'ils sont identiques et à une perte si le sous-arbre R_u est retiré de R durant le processus permettant d'obtenir G. Un aspect important d'un arbre de réconciliation est que sa lecture est ambiguë. Effectivement, u

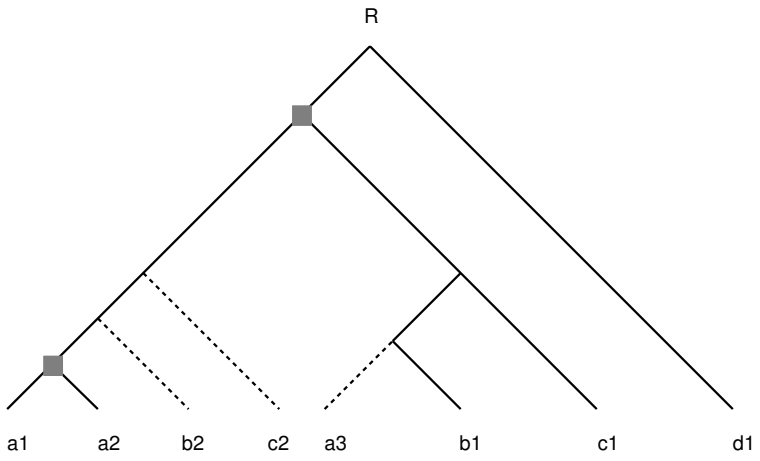

Figure 2.2: Un arbre de réconciliation noté R pour les arbres G et S de la figure 2.1. Les nœuds de duplications de R sont indiqués en gris et les pertes en pointillé.

peut être un nœud de duplication même si $\Lambda(R_{u_1})$ et $\Lambda(R_{u_2})$ sont disjoints, et la perte d'un gène u (soit du sous-arbre R_u) peut être remplacée par un ensemble de pertes à condition que chaque feuille de $L(R_u)$ soit couverte par une de ces pertes. Aussi, un nombre illimité de duplications (et de pertes) peuvent être ajoutées à R et le nombre d'arbres de réconciliation pour les arbres G et S est donc illimité.

Selon la première définition [71] d'un arbre de réconciliation, celui qui est considéré est de taille minimum et est communément appelé *arbre de réconciliation minimum*. Une nouvelle définition récursive [63] permet ensuite d'inférer un unique arbre de réconciliation et les auteurs se demandent si celui-ci est un arbre de réconciliation minimum. Bonizzoni *et al.* [17] donnent une réponse affirmative à cette question et prouvent l'existence d'un seul arbre de réconciliation minimum.

Un algorithme calcule en temps linéaire et sans utiliser le couplage LCA un arbre de réconciliation qui minimise le coût de perte [19]. L'algorithme parcourt simultanément et en débutant aux feuilles les arbres G et S, détecte les pertes induites par le nœud courant de G en comparant avec la spéciation correspondante de S, réinsère ces pertes dans G par l'insertion de sous-arbres de S, contracte les nœuds courants de G et de S et répète ce processus jusqu'à la racine de S.

2.2.3 Arbre DLS

Suite à l'article de Bonizzoni *et al.* [17], plusieurs questions demeurent sans réponse à propos de l'arbre de réconciliation minimum. En particulier, (i) est-il minimal selon les coûts de duplication et de mutation (voir définitions 2.6 et 2.8 respectivement) et (ii) est-ce qu'un tel arbre est unique ? Gorecki et Tiuryn [40] répondent à ces questions en introduisant le concept d'arbre DLS (pour "Duplication, Loss, and Speciation") à partir duquel tous les scénarios d'évolution entre G et S sont considérés. Ensuite, ils définissent des règles combinatoires permettant de transformer un arbre DLS en un nouvel arbre et prouvent qu'une telle architecture est connexe, cohérente (ne contient que des arbres DLS pour G et S) et est pointée par un unique arbre DLS dit de *forme normale*. La définition formelle d'un arbre DLS est donnée ci-dessous et la figure 2.3 donne un exemple.

Définition 2.11. *Soit $\mathcal{I}$ un ensemble d'étiquettes (espèces). Un arbre DLS est soit un arbre vide, soit un arbre binaire noté T, où chaque nœud $v \in V(T)$ est étiqueté par un sous-ensemble non-vide de $\mathcal{I}$ noté $\pi(v)$. L'ensemble de nœuds $V(T)$ est divisé en quatre sous-ensembles disjoints notés $V_\bullet$, $V_\circ$,$V_\square$ et $V_\blacksquare$ et qui respectent les contraintes suivantes :*

1. *si $v \in V_\bullet$, alors $v \in L(T)$ et l'étiquette $\pi(v)$ est composé d'un seul élément de $\mathcal{I}$;*
2. *si $v \in V_\circ$, alors $v \in L(T)$;*
3. *si $v \in V_\square$, alors $v \in V(T) \setminus L(T)$ et $\pi(v) = \pi(v_1) = \pi(v_2)$;*
4. *si $v \in V_\blacksquare$, alors $v \in V(T) \setminus L(T)$ et $\pi(v) = \pi(v_1) \cup \pi(v_2)$ et $\pi(v_1) \cap \pi(v_2) = \emptyset$;*
5. *pour tous nœuds v et w de T, si les deux ensembles $\pi(v)$ et $\pi(w)$ ne sont pas disjoints, alors l'un doit être inclus dans l'autre.*

Ce modèle combinatoire de réconciliation est similaire aux arbres de réconciliation sous les aspects suivants : les clades des deux fils d'un nœud interne sont soit disjoints, soit égaux ; l'arbre de gènes G doit en être extrait par le retrait de tous les nœuds de

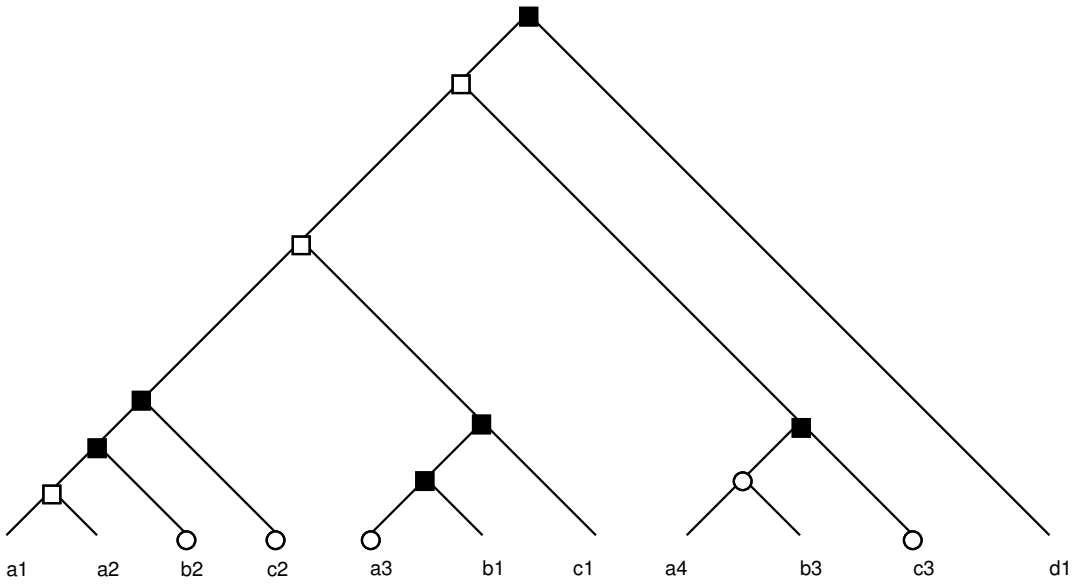

Figure 2.3: Exemple d'un arbre DLS pour les arbres G et S de la figure 2.1, où les symboles suivants (avec leur signification) sont utilisés : ∘ (perte), □ (duplication) et ■ (cospéciation).

pertes ; et le nombre d'arbres DLS est illimité. Contrairement à un arbre de réconciliation, la lecture d'une histoire d'évolution décrite par un arbre DLS n'est pas ambiguë : tous les événements de cospéciation, de duplication et de perte sont formellement indiqués. Enfin, le nombre d'arbres DLS est dénombrable.

Les règles combinatoires appliquées à un arbre DLS pour obtenir un nouvel arbre sont décrites à la figure 2.4. L'aspect important ici est que l'architecture induite par ces opérateurs est de taille non-finie lorsqu'ils sont tous permis alors qu'elle est finie lorsque seulement ceux nommés "TMOVE" et "CLOST" sont permis.

L'architecture des arbres DLS permet de prouver que l'arbre DLS de forme normale est (resp. n'est pas) le seul à minimiser le coût de mutation (resp. duplication) et qu'il est équivalent à l'arbre de réconciliation minimum. Pour les mutations, une preuve plus simple de cette unicité est obtenue grâce aux observations suivantes sur l'arbre de réconciliation calculé par l'algorithme donné en [19] (voir la section 2.2.2) : il est le seul à minimiser le coût de perte et il minimise aussi celui de duplication. Nous pouvons donc conclure que minimiser les pertes donne un arbre de réconciliation qui minimise

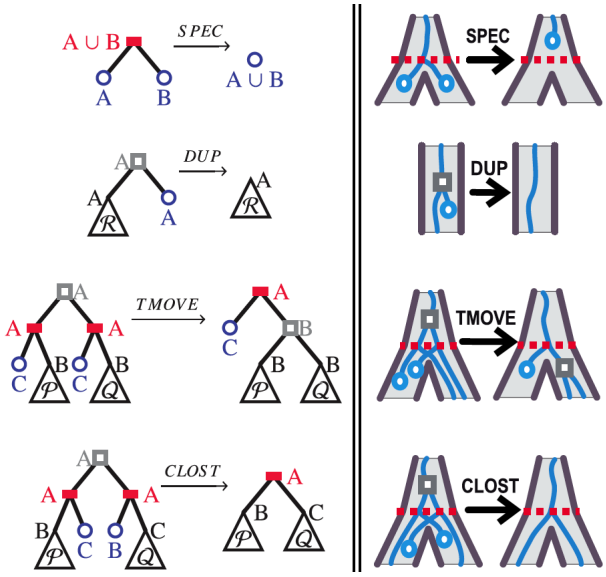

Figure 2.4: Les quatres règles DLS définies par [40]. À titre d'exemple, la règle SPEC transforme l'arbre $(A_\circ, B_\circ)_\blacksquare$ (une cospéciation suivie de deux pertes) en l'arbre $(A \cup B)_\circ$ (une seule perte du clade $A \cup B$). (Figure importée de [40]).

aussi les duplications, mais l'inverse est faux. Le critère de duplication est donc moins contraignant que celui de perte.

2.2.4 Superposition de G sur S

La réconciliation parcimonieuse (section 2.2.1) est définie de façon non ambiguë selon le couplage LCA et le modèle combinatoire d'arbre de réconciliation (section 2.2.2) et celui d'arbre DLS (section 2.2.3) sont basés sur une version étendue de l'arbre de gènes original. Une façon différente de visualiser une réconciliation a été proposée dans un contexte des histoires d'évolution conjointes d'un gène et de son espèce (systématique moléculaire), d'un parasite et de l'espèce hôte (parasitologie), et d'un organisme et de son milieu naturel (biogéographie) [71, 72]. Ce modèle combinatoire de réconciliation d'un arbre de gènes G (dit le parasite) et d'un arbre d'espèces S (dit l'hôte) superpose les deux arbres de façon à représenter une histoire d'évolution du parasite qui soit cohérente avec celle de son hôte : *soit une histoire de coévolution*. La première définition

formelle basée sur cette approche est donnée par Arvestad *et al.* [4, 3, 5, 6], elle n'induit pas que la réconciliation parcimonieuse mais un nombre fini de réconciliations qui sont toutes cohérentes en regard à la coévolution de G et de S. Cette définition est donnée ci-dessous et est suivie d'une interprétation de coévolution pour chaque contrainte.

Définition 2.12. *(Définition 3.3 de [6]) Une réconciliation entre un arbre d'espèces S et un arbre de gènes G est une fonction $\gamma : V(S) \rightarrow 2^{V(G)}$ qui respecte les contraintes données ci-dessous.*

1. $\bigcup_{x \in V(S)} \gamma(x) = V(G)$, $l \in \gamma(\sigma(l))$ *pour toute feuille l de G et* $r(G) \in \gamma(r(S))$.
2. *Pour tout nœud u de G, l'ensemble $\{x \in V(S) : u \in \gamma(x)\}$ induit un chemin orienté dans S où la source est notée $\tau(u)$ et le puits $\beta(u)$.*
3. *Pour tout nœud non-feuille u de G, soit $\beta(u) = \tau(u_1) = \tau(u_2)$ (duplication), soit $\beta(u) = p(\tau(u_1))$ et $\tau(u_2) = s(\tau(u_1))$ (cospéciation).*

La première contrainte oblige chaque nœud de G à être couvert par au moins un nœud de S, chaque gène feuille de G à être couplé à l'espèce contemporaine de S dont il provient et la racine de G à être couverte par celle de S. La deuxième force tous les sommets de S qui couvrent un même nœud de G à former un chemin connecté et orienté selon les arcs de S. La dernière vérifie que le couplage d'un gène non-feuille et celui de ses deux fils respectent soit les contraintes de duplication, soit celles de cospéciation. Par exemple, un nœud de cospéciation u couplé à une espèce ancestrale x doit être telle que l'une des lignées descendantes de u évolue à l'intérieur du sous-arbre S_{x_1} pendant que l'autre lignée évolue dans l'autre sous-arbre (i.e. S_{x_2}). La figure 2.5 donne un schéma d'une réconciliation $\gamma : V(S) \rightarrow 2^{V(G)}$, où il est important de noter que le nœud a de G est un gène dupliqué sur l'arc (x, y) de S.

L'observation ci-dessous explique pourquoi ce modèle combinatoire de réconciliation permet les duplications non-forcées (définition 2.6) et considère d'autres réconciliations que celle dite de parcimonie.

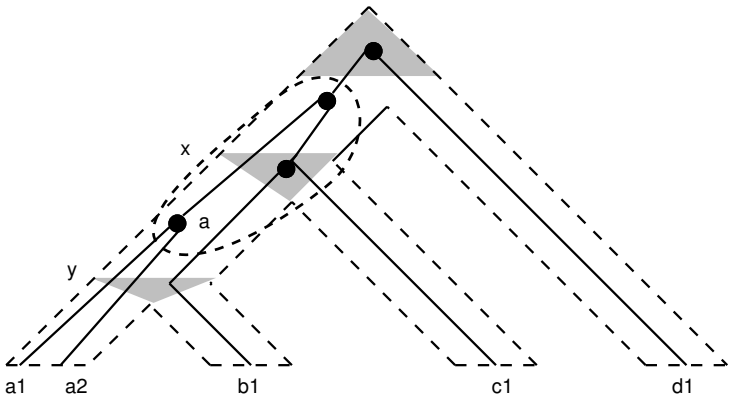

Figure 2.5: Selon la définition 2.12, une réconciliation notée $\gamma : V(S) \rightarrow 2^{V(G)}$ des arbres G et S de la figure 2.1. L'ensemble $\gamma(x)$ est indiqué par l'ellipse et correspond aux trois nœuds de G associés à l'espèce x. Notons que le gène a est à la fois élément de $\gamma(x)$ et de $\gamma(y)$ et $\tau(a) = x$ et $\beta(a) = y$ sont respectivement la source et le puits du chemin induit par a dans S.

Observation 1. *(Observation 4 de [6]) Soit $u \in V(G)$, $x \in V(S)$ et une réconciliation γ entre G et S. Si $u \in \gamma(x)$, alors $M_S(u) \leq_S x$.*

Ce concept de réconciliation de G avec S peut être difficile à interpréter pour un lecteur non familier avec la modélisation combinatoire de l'évolution des familles de gènes. Tout d'abord, un gène ancestral de G peut être couvert par plus d'une espèce de S, alors que l'événement de duplication ou de cospéciation auquel il correspond est survenu à un seul endroit dans l'arbre d'espèces. Aussi, l'histoire d'évolution de G selon les duplications et les cospéciations n'est pas facilement lisible. Autrement dit, déterminer à quel événement un gène ancestral correspond et où il est localisé dans l'arbre d'espèces ne sont pas des tâches triviales sans l'aide d'un schéma. Toutefois, il est important de noter que la raison pour laquelle Arvestad *et al.* ont utilisé une telle définition est probablement dûe au contexte probabiliste (voir le chapitre 3) dans laquelle elle est utilisée.

2.3 Conclusion

Le modèle combinatoire de réconciliation basé sur la superposition de l'arbre de gènes et de l'arbre d'espèces est utilisé dans les prochains chapitres. Le chapitre 3 décrit

l'algorithme développé en [4, 3, 5, 6] pour calculer la probabilité d'une réconciliation selon ce modèle et un modèle probabiliste d'évolution de gènes. Le chapitre 6 donne notre définition de réconciliation qui est équivalente, plus simple et aussi générale que ce modèle, et qui permet d'étudier l'espace combinatoire des réconciliations selon les coûts de duplication, de perte et de mutation (voir la section 2.2.1). Rappelons que la réconciliation basée sur le couplage LCA minimise chacun des trois coûts et qu'elle n'est pas l'unique réconciliation parcimonieuse pour les duplications. Dans ce contexte, le chapitre 6 montre comment notre algorithme d'exploration permet d'énumérer efficacement toutes les réconciliations parcimonieuses. Le chapitre 7 montre comment les outils combinatoires développés au chapitre 6 sont utilisés avec le modèle probabiliste de réconciliation définie par Arvestad *et al.* [4, 3, 5, 6].

CHAPITRE 3

MODÈLES PROBABILISTES DE RÉCONCILIATION

Ce chapitre décrit le modèle probabiliste d'Arvestad *et al.* [3, 5, 6] basé sur le modèle combinatoire de la section 2.2.4. La section 3.1 décrit le processus stochastique utilisé, la section 3.2 développe leur algorithme pour calculer la vraisemblance d'une réconciliation et la section 3.3 donne une brève description de leur architecture Bayésienne.

3.1 Modèle probabiliste d'évolution d'un gène

Un *Processus de Naissance-et-Mort* [52] (voir [70] pour une revue sur les applications biologiques), abrégé PNM par la suite, est un processus stochastique et continu où l'état au temps $t \geq 0$ correspond à un nombre d'individus $n_t \in \mathbb{N}$ représentant la taille d'une population. Les trois transitions possibles pour une période $\Delta t > 0$ sont la naissance d'un individu ($n_{t+\Delta t} = n_t + 1$), la mort d'un individu ($n_{t+\Delta t} = n_t - 1$) et aucun événement ($n_{t+\Delta t} = n_t$), et les probabilités sont respectivement $\lambda\Delta t$, $\mu\Delta t$ et $1-(\lambda+\mu)\Delta t$. Ici, λ et μ correspondent respectivement aux taux de naissance et de mort d'un individu. En assumant que l'état initial au temps $t = 0$ est $n_0 = 1$, la valeur de l'état n_t au temps $t > 0$ est modélisé par un processus de Markov dont les probabilités doivent respecter un système d'équations différentielles nommé "Kolmogorov forward equations" [7]. Formellement, la probabilité qu'un individu au temps $t = 0$ donne naissance à $n \geq 0$ individu(s) au temps $t > 0$ selon un PNM est notée $P_n(t)$ et est définie ci-dessous (où $\lambda \neq \mu$).

$$P_n(t) = \begin{cases} 1 - \frac{\lambda-\mu}{\lambda - u\, e^{-(\lambda-\mu)t}} & \text{si } n = 0 \\ (1 - P_0(t))\,(1-u_t)\,u_t^{n-1} & \text{sinon} \end{cases} \tag{3.1}$$

$$u_t = \frac{\lambda\,(1 - e^{-(\lambda-\mu)t})}{\lambda - u\, e^{-(\lambda-\mu)t}} \tag{3.2}$$

Le PNM est la base d'un modèle probabiliste d'évolution d'un gène à l'intérieur d'un arbre d'espèces S où chaque branche (x, x_1) de S est caractérisée par une longueur en temps notée t_{x_1} et des taux de duplication (naissance ; λ) et de pertes (morts ; μ). Pour un gène u d'une espèce ancestrale x de S, une telle histoire d'évolution débute par le PNM sur l'arc (x, x_1) (resp. (x, x_2)) et, pour chaque descendant de u à l'espèce x_1 (resp. x_2), elle se répète récursivement jusqu'aux espèces contemporaines localisées dans les feuilles de S_{x_1} (resp. S_{x_2}). Afin de modéliser une telle histoire d'évolution, nous devons considérer le cas où le gène u n'a aucun descendant dans les espèces feuilles de S_x, un tel gène est dit *fantôme* car sa lignée s'est éteinte avant d'atteindre n'importe laquelle de ces espèces contemporaines et il n'y a aucun signal de son existence (un gène non fantôme est dit réel). En considérant l'histoire d'évolution du gène u dans la branche (x, x_1), puis dans le sous-arbre S_{x_1}, nous avons les trois notations suivantes : $Q_{x_1}(l)$ correspond à la probabilité que l'espèce x_1 ait l gènes réels descendants du gène u ; $e_V(x)$ dénote la probabilité que u soit un gène fantôme et est donc égale à $Q_{x_1}(0)\ Q_{x_2}(0)$; et la probabilité que l'espèce x_1 ait l gènes réels et d gènes fantômes est notée $K_{x_1}(l, d)$ et est définie ci-dessous.

$$K_{x_1}(l,d) \quad = \quad P_{l+d}(t_{x_1}) \binom{l+d}{d} (1 - e_V(x_1))^l \ e_V(x_1)^d \tag{3.3}$$

Une des difficultés majeures pour calculer la probabilité $Q_{x_1}(l)$ est de considérer un nombre inconnu et illimité de gènes fantômes. Selon un modèle phylogénétique de gain, de perte et de duplication de gènes, Csűrös et Miklós [23, 25, 24] ont dû résoudre cette difficulté pour calculer efficacement la vraisemblance du profile phylogénétique d'une famille de gènes (c'est-à-dire le nombre de gènes par génome). Le modèle considéré ici est celui de perte et de duplication et nous décrivons ci-dessous les différents cas qui ont permis à Arvestad *et al.* [3, 5] de calculer la probabilité $Q_{x_1}(l)$. Si x_1 est une espèce contemporaine de S, $Q_{x_1}(l) = P_l(t_{x_1})$. Sinon, le cas où le nombre de gènes réels est $l = 0$ se décrit de la façon suivante : soit le gène u n'a pas survécu jusqu'à

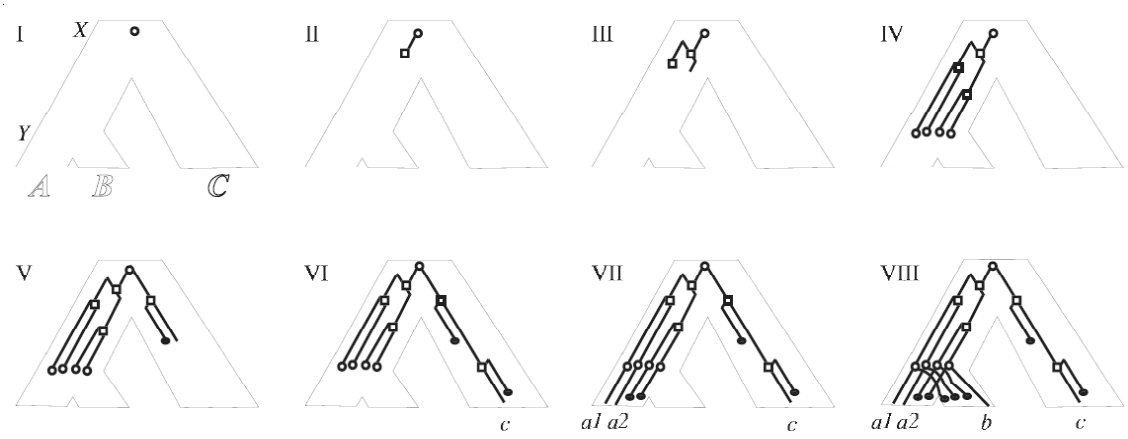

Figure 3.1: Représentation d'une histoire d'évolution d'un gène à l'intérieur d'un arbre d'espèces, où le PNM propre à chaque arc est représenté par les images I à IV pour (x, y), V et VI pour (x, c), VII pour (y, a) et VIII pour (y, b). Une duplication, une perte et une cospéciation sont respectivement représentées par un carré vide, un cercle rempli et un cercle vide. (VIII) En considérant les quatre gènes de l'espèce y, le troisième (en partant de la gauche) est le seul gène fantôme. (Figure importée de [3]).

l'espèce x_1, il n'y a donc aucun gène fantôme et $Q_{x_1}(0) = P_0(t_{x_1})$; soit il y a au moins un gène fantôme ($d \geq 1$) dont la lignée est morte dans le sous-arbre d'espèces S_{x_1} et $Q_{x_1}(0) = \sum_{d=1}^{+\infty} K_{x_1}(0, d)$. Enfin, le cas où le nombre de gènes réels est tel que $l > 0$ se décrit comme suit : le gène u a survécu jusqu'à l'espèce x_1 et a donné naissance à au moins un (resp. possiblement aucun) gène réel (resp. fantôme) et nous avons donc $Q_{x_1}(l) = \sum_{d=0}^{+\infty} K_{x_1}(l, d)$. En développant ces trois cas disjoints, nous obtenons l'équation suivante, où x_1 et t_{x_1} sont respectivement remplacés par y et t.

$$Q_y(l) = \begin{cases} P_l(t) & \text{si } y \in L(S) \\ P_0(t) \ + \ (1 - P_0(t)) \ \frac{(1 - u_t) \ e_V(y)}{1 - u_t e_V(y)}, & \text{si } y \notin L(S) \text{ et } l = 0 \\ (1 - P_0(t)) \ \frac{(1 - u_t) \ u_t^{l-1}}{(1 - u_t e_V(y))^{l+1}}, & \text{si } y \notin L(S) \text{ et } l \neq 0 \end{cases} \tag{3.4}$$

3.2 Algorithmes

3.2.1 Algorithme pour calculer la vraisemblance

Le modèle d'évolution d'un gène décrit dans la section 3.1 est utilisé par Arvestad *et al.* pour calculer la probabilité que l'histoire d'évolution d'un gène appartenant à l'espèce "racine" de S et évoluant récursivement dans S jusqu'aux espèces contemporaines ait pour résultat un arbre de gènes G et une réconciliation $\gamma : V(S) \rightarrow 2^{V(G)}$ (définition 2.12). Autrement dit, l'ensemble de gènes réels obtenu par le retrait des gènes fantômes induit une topologie isomorphe à G pour lequel l'histoire d'évolution (en termes de duplications et de cospéciations) est consistante avec la réconciliation γ. Pour des taux de duplication et de perte donnés, cette probabilité est notée $P(G, \gamma)$ et est décrite ci-dessous selon la notation introduite par [6].

Pour un sommet interne x de S et un nœud u de G telle que $u \in \gamma(x)$, $r_V(x, u)$ dénote la probabilité que l'évolution du gène u à l'intérieur de l'arbre d'espèces S_x donne la paire (G_u, γ_u), où γ_u est la sous-réconciliation entre G_u et S_x induite par γ, et $r_A(x_1, u)$ est la composante de $r_V(x, u)$ pour l'arc (x, x_1) et le sous-arbre S_{x_1} ([1]). Le calcul de la probabilité $P(G, \gamma)$ est basé sur une équation récursive (formellement décrite à la fin de cette section) dont les composantes sont la probabilité $Q_y(l)$ (équation 3.4) et deux nouvelles probabilités développées ci-dessous. Avant tout, considérons le sous-arbre de G dont la racine est u, les feuilles sont les descendants de u couverts par l'espèce x_1 et les nœuds internes correspondent à des duplications survenues sur l'arc (x, x_1). Cet arbre est noté $G_{u||x_1}$ et représente le sous-arbre de G_u qui a évolué sur l'arc (x, x_1) (voir la figure 3.2).

Si $T_u = G_{u||x_1}$ et $l(T_u) = |L(T_u)|$, la probabilité $Q_{x_1}(l(T_u))$ (section 3.1) ne prend pas en considération ni T_u, ni son histoire d'évolution dans (x, x_1). Cet aspect probabiliste est modélisé par des étiquettes sur les arcs et les nœuds de T_u, par le nombre d'arbres

[1] Pour $r_V(x, u)$ (resp. $r_A(x_1, u)$), "r" signifie reconciliation et l'indice "V" (resp. "A") indique qu'elle débute au sommet (resp. arc) considéré de S.

étiquetés isomorphes à T_u et assume que tous les arbres étiquetés sont équiprobables. Pour un nœud non-feuille v de T_u, les étiquettes de ses deux arcs fils déterminent lequel des deux sous-arbres T_{v_1} et T_{v_2} est à droite et l'autre à gauche selon un PNM sur l'arc (x, x_1) et s'ils sont isomorphes, les deux étiquetages possibles sont équivalents et v est dit automorphique. Si $w(T_u)$ est le nombre de nœuds automorphiques de T_u, le nombre de façons d'étiqueter ses arcs est $\alpha(T_u) = 2^{l(T_u)-1-w(T_u)}$. Les étiquettes des nœuds de T_u définissent leur ordre de naissance et doivent respecter les contraintes suivantes : la racine de T_u est le premier sommet né, toutes les feuilles de T_u sont nées au même moment, tous les autres sommets sont nés à des moments différents et un sommet non-feuille de T_u doit naître avant ses deux fils. Le nombre de façons d'étiqueter les nœuds de T_u est noté $\beta(T_u)$, où $\beta(T_u) = 1$ si $u \in L(T_u)$, sinon $\beta(T_u) = \binom{l(T_{u_1})+l(T_{u_2})-2}{l(T_{u_1})-1} \beta(T_{u_1})\, \beta(T_{u_2})$. Finalement, la probabilité d'obtenir un arbre étiqueté isomorphe à T_u est $h(T_u) = \frac{\alpha(T_u)\ \beta(T_u)}{(l(T_u)-1)!}$, où le dénominateur est le nombre d'arbres étiquetés ayant $l(T_u)$ feuilles. La figure 3.2 montre un exemple de cette méthode de dénombrement.

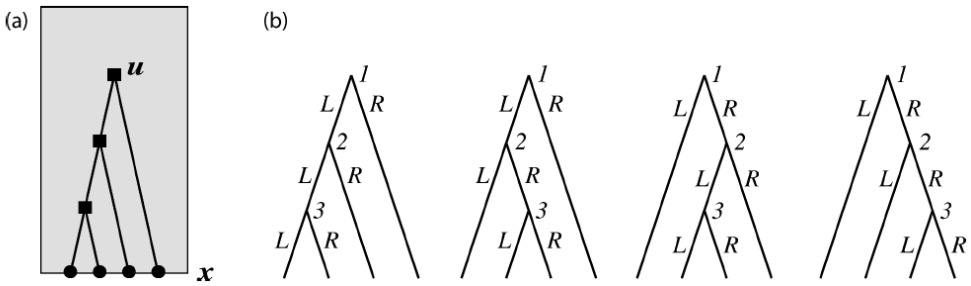

Figure 3.2: Évolution d'un gène sur un seul arc $(p(x), x)$ d'un arbre d'espèces. a) Réconciliation du sous-arbre $G_{u||x}$ sur cet arc, où la racine (i.e. u) de $G_{u||x}$ est couplée à l'espèce $p(x)$ et ses feuilles à l'espèce x. b) Les quatre arbres étiquetés (sur un total de 6) qui sont isomorphes à l'arbre $G_{u||x}$. L'étiquette d'un nœud indique son ordre de naissance par rapport aux autres nœuds et les étiquettes de ses deux arcs fils indiquent de quel "côté" les deux lignées de gènes ont évolué. (Figure importée de [6]).

Pour que le calcul de la probabilité $r_V(x, u)$ soit complet, il est nécessaire de considérer toutes les paires (G'_u, γ'_u) qui sont isomorphes à la paire désirée (G_u, γ_u). Le dénombrement consiste à assigner à chaque feuille $v \in L(T_u)$ du sous-arbre $T_u = G_{u||x_1}$

une étiquette représentant la classe isomorphique à laquelle la paire (G_v, γ_v) appartient. Un nœud non-feuille v de T_u tel que $T_{v_1} \cong T_{v_2}$, où $f : V(T_{v_1}) \to V(T_{v_2})$ est la fonction d'isomorphisme, possède un étiquetage distinct de ses feuilles si et seulement si $\exists w \in L(T_{v_1})$ tel que $(G_w, \gamma_w) \ncong (G_{f(w)}, \gamma_{f(w)})$. Un tel nœud induit une réconciliation différente et isomorphe à (G_u, γ_u) seulement par l'échange des deux réconciliations enracinées aux nœuds w et $f(w)$. Soit $\pi(G_{u||x_1})$ le nombre de nœuds automorphiques avec étiquetage distinct de ses feuilles. Alors, en ne considérant que les réconciliations enracinées aux feuilles de $G_{u||x_1}$, $W(x_1, u) = 2^{\pi(G_{u||x_1})}$ correspond au nombre de réconciliations isomorphes à la réconciliation induite par (G_u, γ_u) sur l'arc (x, x_1) et dans le sous-arbre S_{x_1}.

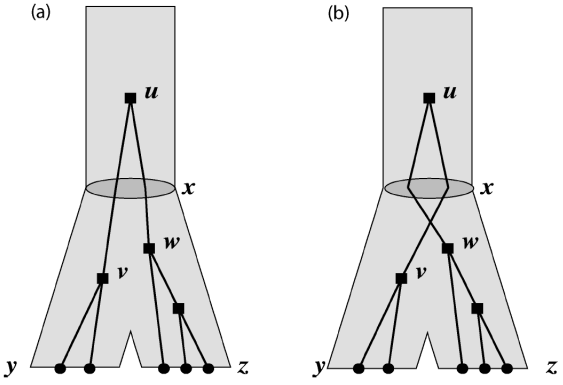

Figure 3.3: Évolution d'un arbre de gènes G dans un arbre d'espèces S enraciné à un arc $(p(x), x)$ et ayant deux feuilles y et z. (a) et (b) représentent deux évolutions de G dans S de telle sorte que les deux réconciliations induites sont isomorphes. (Figure importée de [6]).

Selon la probabilité $Q_x(l)$ décrite dans la section 3.1 et les différentes composantes développées ci-dessus, la probabilité $r_V(x, u)$ est calculée par la récursion donnée ci-dessous. Enfin, la complexité pour calculer $P(G, \gamma)$ est donnée au théorème 3.1.

Récursion 1. *La probabilité d'un scénario d'évolution* (G, γ)*. Soit* $x \in V(S)$*,*

$$r_V(x,u) = \begin{cases} 1 & \textit{si } x \in L(S), u \in L(G), u \in \gamma(x) \\ r_A(x_1,u)\, r_A(x_2,u) & \textit{sinon} \end{cases}$$

$$r_A(x,u) = \begin{cases} Q_x(0) & \textit{si } L(G_{u||x}) = \emptyset \\ Q_x(|L(G_{u||x})|)\, W(x,u)\, h(G_{u||x}) \prod\limits_{v \in L(G_{u||x})} r_V(x,v) & \textit{sinon.} \end{cases}$$

Théorème 3.1. *(Théorème 5.21 de [6]). La probabilité* $P(G, \gamma)$ *égale* $r_V(r(S), r(G))$ *et peut être calculée en temps* $O(n_S\, n_G)$*.*

3.2.2 Autres algorithmes

En s'inspirant de la récursion de $P(G, \gamma)$, Arvestad *et al.* ont développé un algorithme polynomial pour calculer la probabilité qu'un arbre de gènes G soit le résultat d'une histoire d'évolution dans S, selon des taux λ et μ donnés. Un tel algorithme est pleinement justifié car cette probabilité est telle que $P(G) = \sum_{\gamma \in \Psi(G,S)} P(G, \gamma)$, où $\Psi(G, S)$ est l'ensemble de toutes les réconciliations entre G et S et est probablement de taille exponentielle. Brièvement, l'idée repose sur un algorithme de type programmation dynamique qui calcule de façon implicite la somme donnée ci-haut en utilisant un opérateur sur deux ensembles de réconciliations entre sous-arbres afin de les combiner pour obtenir une réconciliation plus large. La complexité de leur algorithme est donnée ci-dessous.

Théorème 3.2. *(Théorème 6.9 de [6]) La probabilité* $P(G)$ *peut être calculée en temps* $O({n_G}^2\, n_S)$*.*

Selon les théorèmes 3.1 et 3.2 et celui de Bayes, où $P(\gamma|G) = \frac{P(G,\gamma)}{P(G)}$, la probabilité postérieure d'une réconciliation γ sachant G peut être calculée avec la même complexité que la probabilité $P(G)$. Le corollaire ci-dessous est immédiatement obtenu.

Corollaire 3.3. *La probabilité* $P(\gamma|G)$ *peut être calculée en temps* $O({n_G}^2\, n_S)$*.*

Finalement, la récursion utilisée pour calculer $P(G)$ induit une représentation implicite de la distribution de l'ensemble des réconciliations et permet d'échantillonner une réconciliation $\gamma \in \Psi(G, S)$ selon $P(\gamma|G)$. Elle permet aussi de calculer la réconciliation qui maximise la vraisemblance.

Théorème 3.4. *(Théorème 7.12 de [6]) Parmi l'ensemble des réconciliations pour G et S, calculer la réconciliation γ qui maximise $P(G, \gamma)$ se fait en temps $O(n_S\, n_G\, \log^3 n_G)$.*

3.3 Architecture Bayésienne

Pour une hypothèse notée H et des données notées D, la probabilité postérieure de H sachant D, notée $P(H|D)$, est définie selon le Théorème de Bayes décrit ci-dessous.

$$P(H|D) = \frac{P(D|H)P(H)}{P(D)} = \frac{P(D|H)P(H)}{\int P(D|H)\ P(H)\ dH}. \tag{3.5}$$

Dans Arvestad *et al.* [3], une architecture Bayésienne est introduite afin d'estimer la probabilité a posteriori d'une réconciliation $\gamma : V(S) \to 2^{V(G)}$ (définition 2.12) et des taux λ et μ sachant l'arbre de gènes G. Leur approche est basée sur un "Markov Chain Monte-Carlo" [37] (MCMC) dans lequel un triplet (γ, λ, μ) représente un état de la chaîne de Markov. La distribution stationnaire recherchée est la distribution a posteriori décrite ci-dessous (où une distribution uniforme sur les "priors" $P(\gamma, \mu)$ est utilisée).

$$\begin{aligned} P(\gamma, \lambda, \mu|G) &= \frac{P(G, \gamma|\lambda, \mu)\ P(\lambda, \mu)}{P(G)} \\ &= \frac{P(G, \gamma|\lambda, \mu)\ P(\lambda, \mu)}{\int P(G, \gamma|\lambda, \mu)\ P(\lambda, \mu)\ d\gamma\ d\lambda\ d\mu} \\ &= \frac{P(G, \gamma|\lambda, \mu)}{\int P(G, \gamma|\lambda, \mu)\ d\gamma\ d\lambda\ d\mu} \end{aligned}$$

Dans Arvestad *et al.* [5], le modèle d'évolution avec duplications et pertes (introduit dans [3] et décrit à la section 3.1) est étendu à un modèle hiérarchique qui considère aussi

les mutations. Selon un modèle d'évolution de séquences [33] choisi[2] et l'hypothèse de l'horloge moléculaire [54], ils ont développé un MCMC pour estimer la probabilité a posteriori d'un arbre de gènes selon un ensemble de séquences moléculaires. Ce modèle est amélioré en [1], où l'horloge moléculaire est relaxée grâce à un modèle d'évolution de taux de substitution entre les branches de l'arbre de gènes.

Pour des taux λ et μ donnés, la probabilité d'une cospéciation à un nœud interne de G est calculée en temps $O({n_G}^2\ n_S)$ par l'algorithme pour $P(G)$ (théorème 3.2), où seulement les réconciliations qui contiennent cette cospéciation sont considérées. En utilisant de façon conjointe cet algorithme avec un MCMC qui échantillonne les taux, [81] présente une architecture bayésienne pour estimer les probabilités postérieures des relations d'orthologie pour la famille de gènes.

3.4 Conclusion

Les méthodes d'Arvestad *et al.* pour calculer les probabilités d'un arbre de gènes G, d'une réconciliation et d'une relation d'orthologie sont basées sur un algorithme complexe qui calcule la probabilité exacte de G. Toutefois, un petit nombre de réconciliations suffit pour calculer une bonne approximation de cette probabilité. Nous présentons au chapitre 7 une méthode simple et efficace pour explorer ce sous-espace de réconciliations dans une architecture probabiliste.

[2] Leur MCMC peut facilement être adapté à différents modèles (Jukes-Cantor [51], GTR [55], ou "evolving rates" [86]).

CHAPITRE 4

INFÉRENCE D'UN ARBRE D'ESPÈCES SELON UN ENSEMBLE D'ARBRES DE GÈNES

La *phylogénomique* [26] infère une phylogénie d'espèces par l'analyse des différents signaux phylogénétiques de plusieurs familles de gènes. Nous présentons dans ce chapitre une méthode phylogénomique de type super-arbre [15], où les arbres des familles sont combinés afin d'inférer celui des espèces. Le critère considéré est la parcimonie des réconciliations des arbres de gènes G avec un arbre d'espèces S selon le coût soit des duplications, soit des pertes, soit des mutations (section 2.2.1), et est noté $c(G, S)$. Ce problème d'optimisation combinatoire est nommé "Gene Tree Parsimony" (GTP) [82] et est formellement défini ci-dessous.

Problème 1. *Le problème GTP.*
Entrée : une forêt d'arbres de gènes $\mathcal{F} = \{G_1, G_2, \ldots, G_r\}$.
Sortie : un arbre d'espèces S *tel que son coût de réconciliation avec la forêt* $\mathcal{F}$*, noté* $C(\mathcal{F}, S) = \sum_{i=1}^{r} c(G_i, S)$*, est minimal parmi l'ensemble des arbres d'espèces possibles pour* $\mathcal{F}$*. Formellement,* S *est tel que* $\Lambda(S) = \bigcup_{i=1}^{r} \Lambda(G_i)$.

4.1 Complexité du problème

Le GTP est NP-complet pour le coût de duplication et celui de mutation [63] et sa complexité est inconnue pour le critère de perte. En [19], le problème pour le coût de duplication est montré équivalent à une version restreinte du problème de super-arbre [15] définie comme suit : pour un ensemble $\mathcal{G}$ d'espèces et un ensemble $\mathcal{B}$ de bipartitions dites "désirées" pour $\mathcal{G}$, trouver un arbre d'espèces S qui respecte un maximum de bipartitions de $\mathcal{B}$. Un nœud interne u de G est dit *duplication apparente* lorsque $\Lambda(G_{u_1}) \cap \Lambda(G_{u_2}) \neq \emptyset$: u est une duplication pour tout arbre d'espèces S. L'équivalence

est prouvée en considérant l'ensemble $\mathcal{B}$ des bipartitions induites par les nœuds des arbres G de $\mathcal{F}$ qui ne sont pas des duplications apparentes. Enfin, ces auteurs montrent aussi qu'il est possible de décider en temps polynomial s'il existe un arbre S tel que $D(\mathcal{F}, S)$ est le nombre de duplications apparentes dans $\mathcal{F}$.

4.2 Méthodes basées sur le voisinage

Lorsqu'une énumération de tous les arbres d'espèces n'est pas possible, les heuristiques les plus utilisées appliquent une recherche locale basée sur des opérations d'édition d'arbres (ici l'arbre d'espèces) tel que le "Nearest Neighbor Interchange" (abrégée par NNI) et le "rooted Subtree Pruning and Regrafting" [85] (abrégée par rSPR). Pour un arbre d'espèces S, le mouvement NNI change la position de deux sous-arbres de S dont les racines sont jointes par un chemin formé de trois branches et le mouvement rSPR arrache un sous-arbre de S et le greffe sur une nouvelle branche (voir la figure 4.1).

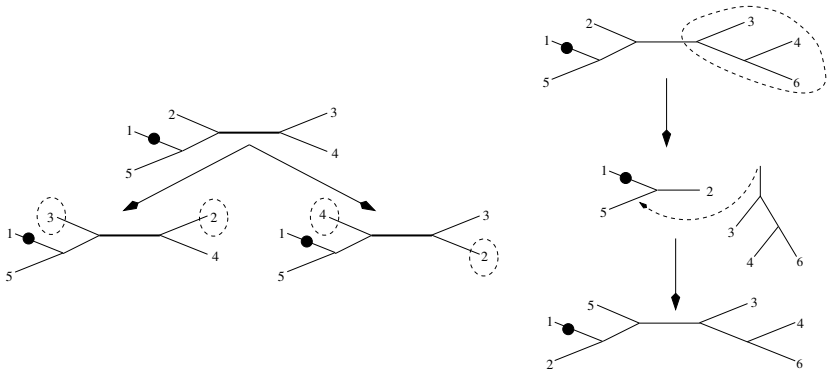

Figure 4.1: (Gauche) Mouvement NNI. Les sous-arbres adjacents de la branche représentée en gras sont permutés ; les sous-arbres permutés sont entourés par des pointillés. (Droite) Mouvement rSPR. Le sous-arbre entouré par des pointillés est arraché, puis greffé sur une nouvelle branche de l'arbre résultant. Le rond noir indique la position de la racine des arbres.

Ci-dessous, deux algorithmes efficaces sont décrits pour explorer les arbres d'espèces localisés dans le voisinage de S (induit soit par le NNI, soit par le rSPR) et calculer leur coût de réconciliation selon le critère de duplication. Assumons que la taille de tous les arbres de gènes G de $\mathcal{F}$ est $O(n_G)$. Rappelons que la complexité pour calculer le coût de

la réconciliation parcimonieuse entre G et S est $O(n_S + n_G)$ [97] (voir la section 2.2.1).

Une heuristique de recherche locale basée sur une exploration efficace du voisinage de S induit par au plus k NNI (dit voisinage k-NNI, pour $k = 1, 2, 3$) est décrite en [11]. Notons que l'algorithme naïf pour explorer un tel voisinage se fait en temps $O(r\ {n_S}^k\ (n_S + n_G))$, où $\Theta({n_S}^k)$ est la taille du voisinage de S. L'idée principale de leur algorithme est la suivante : après avoir calculé le coût des $\Theta(n_S)$ voisins immédiats (noté S') de S, celui de la majorité des voisins de S' peut être calculé en temps $\Theta(1)$ alors qu'il y a un nombre constant de voisins pour lesquels leur coût doit être recalculé. Enfin, leur algorithme a un temps de $O(r\ n_S\ (n_S + n_G))$ pour $k = 2, 3$. Leurs expériences sur des données simulées montrent que i) leurs temps de calcul sont grandement inférieurs à ceux de GeneTree [74] ; ii) l'arbre d'espèces obtenu par l'heuristique basée sur le voisinage 2-NNI a grandement amélioré le coût de réconciliation face au voisinage 1-NNI et iii) les temps de calcul de ces deux heuristiques sont linéairement comparables.

Dans [8], l'opérateur d'édition rSPR est utilisé pour l'exploration du voisinage immédiat de S, dont la taille est $\Theta({n_S}^2)$. L'exploration naïve de ce voisinage a une complexité de $\Theta(r\ {n_S}^2\ (n_S + n_G))$ et il est possible de le partitionner en $\Theta(n_S)$ sous-voisinages. Un tel sous-voisinage est formé des voisins de S obtenus par un rSPR appliqué à un sous-arbre noté P de S, est de taille $\Theta(n_S)$ et est efficacement exploré de la façon suivante : greffer P à la racine de S afin d'obtenir un premier voisin S', appliquer un parcours en profondeur de S' et greffer P au nœud courant (ceci équivaut à un NNI). Selon des propriétés combinatoires du couplage LCA et du NNI, chaque nouveau coût se calcule en temps constant après avoir calculé celui de S'. La complexité totale de leur algorithme est $O(r\ n_S\ (n_S + n_G))$ et leurs expériences sur des données simulées montrent que leur programme est beaucoup plus rapide que GeneTree. Enfin, cet algorithme est légèrement modifié dans [10], où le voisinage considéré est celui obtenu par le "Tree Bisection and Reconnection" [85].

Le programme *GeneTree* [74] calcule le couplage LCA en temps linéaire [31] et utilise le NNI, le rSPR ou une alternance entre les deux lors de la recherche locale.

Ce programme considère les coûts de duplication, de mutation et celui nommé “deep coalescence” [66] et il permet de restreindre l’espace de recherche avec des contraintes sur l’arbre d’espèces. Un second programme nommé *DupTree* [94] ne considère que le coût de duplication, il accepte des arbres de gènes non-enracinés et la recherche locale est basée sur l’opérateur rSPR et sur un algorithme efficace pour l’exploration du voisinage (décrit ci-dessus). Enfin, il permet d’assigner un poids sur le coût associé à chaque arbre de gènes et de définir des contraintes sur l’arbre d’espèces. Selon leurs expériences sur de larges jeux de données, DupTree est plus rapide que GeneTree.

4.3 Autres méthodes

En [79], seulement sept espèces sont considérées et une énumération exhaustive des 945 arbres d’espèces est effectuée afin de résoudre le problème GTP. En [20], une heuristique vorace est présentée pour le coût de perte selon un parcours de bas-en-haut de G et le choix glouton suivant : pour chaque ensemble maximal de nœuds de G dont les clades ne sont pas disjoints, une heuristique choisit un sous-arbre d’espèces et les pertes induites de cet arbre et cet ensemble sont réinsérées dans G.

Pour le coût de duplication et celui de mutation, [46] présente un algorithme FPT (“Fixed Parameter Tractable”), où le paramètre considéré est le nombre maximal de copies du gène pouvant simultanément exister dans une branche de S. [84] donne aussi un algorithme FPT, où le paramètre est le nombre de duplications nécessaires pour réconcilier G avec S.

4.4 Conclusion

Jusqu’à ce jour, aucune méthode exacte pour le problème GTP avec chacun des trois coûts n’est proposée et la majorité des heuristiques existantes considèrent les coûts de duplication et de mutation et une seule celui de perte. Le chapitre 8 présente une première méthode exacte pour résoudre ce problème et est intéressante pour évaluer l’efficacité

des heuristiques, les comparer et améliorer les phylogénies d'espèces proposées.

CHAPITRE 5

CONTRIBUTIONS DE MA RECHERCHE

Ce chapitre décrit les principales contributions de cette thèse : un nouveau modèle combinatoire de réconciliation (section 5.2) ; des contributions théoriques et expérimentales au modèle probabiliste d'Arvestad *et al.* (section 5.3) ; un algorithme exact pour le problème GTP (section 5.4). La section 5.1 introduit deux schémas d'algorithme pour énumérer les éléments d'un modèle combinatoire [76], utilisés dans cette thèse pour les réconciliations et les phylogénies d'espèces.

5.1 Exploration d'un espace combinatoire

Considérons l'objet combinatoire suivant : soit un entier $k \in \mathbb{N}$, un alphabet $\mathcal{A}$ de cardinalité n et un prédicat noté P, l'ensemble d'objets considérés est formé de tous les mots de k lettres respectant le prédicat P et défini par $S = \{w \in \mathcal{A}^k : P(w) = vrai\}$. Un algorithme d'énumération des mots de S est dit en *Temps Amorti Constant* (TAC[1]) si le temps pour générer tous ses éléments est proportionel à sa cardinalité : entre deux objets générés successivement, le délai est constant en moyenne sur tout le processus de génération.

Un algorithme nommé *Retour Arrière* est basé sur un arbre d'exploration noté $\mathcal{T}$ et décrit comme suit : la racine est le mot vide noté ε, ses fils sont les n mots obtenus par la concaténation d'une lettre de $\mathcal{A}$ à ε et cette arborescence est répétée récursivement jusqu'aux mots de longueur k localisés dans les feuilles de $\mathcal{T}$. L'énumération de S basée sur $\mathcal{T}$ génère plusieurs mots $w \in \mathcal{A}^k$ tel que $P(w)$ est faux. Une énumération sans échec repose sur un sous-arbre $\mathcal{T}'$ de $\mathcal{T}$ pour lequel tous et seulement tous les mots de S forment l'ensemble de ses feuilles. Enfin, si la taille de $\mathcal{T}'$ est linéaire selon la cardinalité de S et si la génération de tous les mots fils d'un mot donné est en temps linéaire, alors

[1] "Constant Amortized Time per element".

l'exploration de $\mathcal{T}'$ permet une énumération de S avec une complexité TAC.

Un *algorithme lexicographique* génère tous les objets de S grâce à une fonction nommée *Prochain(x)* qui retourne l'élément qui suit un objet $x \in S$ selon un *ordre total strict*. Pour plusieurs objets combinatoires, la complexité de cette fonction est $O(k)$ dans le pire des cas et si son temps moyen est constant, l'algorithme est TAC.

5.2 Nouveau modèle combinatoire de réconciliation

Le chapitre 6 donne une nouvelle définition de réconciliation entre un arbre de gènes et un arbre d'espèces qui respecte les contraintes nécessaires pour définir une histoire de coévolution cohérente entre G et S (propriété 2.5). Ce nouveau modèle combinatoire permet une lecture facile des événements de duplication et de spéciation et de leur localisation dans S ; il induit un espace combinatoire de taille finie où la réconciliation LCA a le rôle de minima selon les couplages permis ; et il est équivalent au modèle d'Arvestad *et al.* (section 2.2.4) tout en étant plus intuitif.

Ce modèle combinatoire nous a permis de développer des algorithmes simples et efficaces pour générer aléatoirement une réconciliation, calculer la taille de l'espace des réconciliations et explorer cet espace en temps optimal. Cet exploration est basée sur des opérateurs appliqués à une réconciliation initiale afin d'en définir une nouvelle, où le nombre de duplications et de pertes est mis à jour en temps constant, et sur une arborescence enracinée à la réconciliation LCA et de profondeur quadratique selon la taille de G et de S. L'algorithme suit un ordre lexicographique des réconciliations et est TAC selon la taille de l'espace.

Ces différents outils combinatoires apportent une contribution non négligeable au MCMC développé par Arvestad *et al.* (voir la section 3.3) pour estimer les probabilités postérieures des réconciliations. En particulier, ils ne définissent pas leurs opérateurs et ne prouvent pas qu'ils sont suffisants pour une exploration complète de l'espace des réconciliations. Aussi, une des méthodes généralement utilisée pour étudier la conver-

gence d'un MCMC est de comparer les probabilités calculées par différentes chaînes de Markov dont les points de départ sont aléatoirement choisis selon une distribution uniforme. Notre algorithme de génération aléatoire peut donc être utilisé pour tester la convergence de leur MCMC. Connaître la taille de l'espace permet aussi de justifier une telle approche.

Afin d'évaluer l'efficacité de la parcimonie à retrouver le vrai scénario d'évolution, nous avons considéré des arbres de gènes synthétiques et analysé l'arborescence des réconciliations selon la distribution des coûts de duplication, de perte et de mutation (voir les définitions 2.6, 2.7 et 2.8).

5.3 Contributions au modèle probabiliste d'Arvestad *et al.*

En utilisant l'algorithme d'exploration et l'arborescence des réconciliations (chapitre 6) avec le modèle probabiliste d'Arvestad *et al.*, le chapitre 7 décrit une nouvelle méthode pour calculer efficacement les probabilités postérieures des réconciliations. Dans le contexte où la vraisemblance d'une réconciliation se calcule en temps quadratique selon les tailles de G et de S (voir le théorème 3.1), notre contribution théorique est une mise à jour de cette vraisemblance en temps linéaire après l'application d'un opérateur. Cette diminution de la complexité nous a permis d'étudier la forme de l'arborescence selon les probabilités des réconciliations. Pour des taux de duplication et de perte réalistes, nos résultats montrent que la réconciliation LCA maximise la probabilité postérieure et que la masse probabiliste de l'arborescence est localisée dans une sous-arborescence de petite taille. Ainsi, les probabilités de l'arbre de gènes et des réconciliations les plus probables peuvent être estimées rapidement et précisément. Enfin, les résultats obtenus avec des taux plus élevés sont similaires.

Ces analyses de l'espace des réconciliations sur un grand jeux de données sont les premiers réalisés suite aux avancés théoriques d'Arvestad *et al.* pour calculer la vraisemblance et la probabilité d'une réconciliation. Alors que la parcimonie propose des

topologies souvent érronées lors d'études phylogénétiques, ces résultats préliminaires indiquent qu'elle pourrait être plus appropriée pour la réconciliation d'un arbre de gènes et d'un arbre d'espèces.

5.4 Méthode exacte pour le problème GTP

Comme nous l'avons indiqué au chapitre 4, aucune méthode exacte n'a été proposée pour résoudre le problème GTP selon chacun des trois coûts (duplication, perte et mutation). Aussi, pour tout problème d'optimisation combinatoire, la valuation des solutions proposées par différentes heuristiques en rapport à leur coût requiert le coût d'une solution optimale.

Le chapitre 8 développe la première méthode exacte pour résoudre ce problème selon une approche de "Branch-and-Bound" basée sur une exploration de type Retour Arrière sans échec (section 5.1) et d'une arborescence d'arbres pour n espèces. Celle-ci est différente de l'architecture classique utilisée en inférence phylogénétique, elle permet de mettre à jour efficacement chacun des trois coûts et de prendre en considération des connaissances a priori sur la phylogénie des espèces considérées, réduisant ainsi les temps de calcul. En réduisant l'espace des phylogénies pour 29 eucaryotes, nous avons appliqué notre approche sur 1111 familles de gènes et les solutions calculées pour les trois coûts sont similaires à la phylogénie proposée par TreeFam [57].

CHAPITRE 6

PRÉSENTATION DU PREMIER ARTICLE

6.1 Détails de l'article

Space of Gene/Species Trees Reconciliations and Parsimonious Models

Jean-Philippe Doyon, Cedric Chauve et Sylvie Hamel

Publié dans *Journal of Computational Biology*

6.2 Partage du travail

M. Chauve, M. Doyon et Mme. Hamel ont défini le plan. M. Chauve et M. Doyon ont rédigé l'article. M. Doyon a implémenté les algorithmes, s'est occupé des expériences et a défini le plan des expériences avec M. Chauve. M. Doyon a développé les algorithmes, les preuves et la majorité des composantes théoriques. M. Chauve a aidé M. Doyon à rédiger certaines preuves.

Space of Gene/Species Trees Reconciliations and Parsimonious Models *

Jean-Philippe Doyon[1], Cedric Chauve[2], and Sylvie Hamel[1]

[1] DIRO, Université de Montréal, CP6128, succ. Centre-Ville, H3C 3J7, Montréal (QC),Canada,

[2] Department of Mathematics, Simon Fraser University, 8888 University Drive, V5A 1S6, Burnaby (BC), Canada,

Abstract. We describe algorithms to study the space of all possible reconciliations between a gene tree and a species tree, that is counting the size of this space, uniformly generate a random reconciliation, and exploring this space in optimal time using combinatorial operators. We also extend these algorithms for optimal and sub-optimal reconciliations according to the three usual combinatorial costs (duplication, loss, and mutation). Applying these algorithms to simulated and real gene family evolutionary scenarios, we observe that the LCA (Last Common Ancestor) based reconciliation is almost always identical to the real one.

1 Introduction

Genomes of contemporary species, especially eukaryotes, are the result of an evolutionary history that started with a common ancestor from which new species evolved through evolutionary events called speciations. One of the main objectives of molecular biology is the reconstruction of this evolutionary history, that can be depicted with a rooted binary tree, called a *species tree*, where the root represents the common ancestor, the internal nodes the ancestral species and speciation events, and the leaves the extant species. Other events than speciation can happen, that do not result immediately in the creation of new species but are essential in eukaryotic genes evolution, such as gene duplication and loss (Graur and Li, 1999). Duplication is the genomic process where one or more genes of a single genome are copied, resulting in two copies of each duplicated gene. Gene duplication allows one copy to possibly develop a new biological function through point mutation, while the other copy often preserves its original role. A gene is considered to be lost when the corresponding sequence has been deleted by a genomic rearrangement or has completely lost any functional

* Previously published as a proceeding (Doyon *et al.*, 2008)

role (i.e. has become a pseudogene). (See Graur and Li (1999) for example). Other genomic events such as lateral gene transfer, that occurs mostly in bacterial genomes, will not be considered here.

Genes of contemporary species that evolved from a common ancestor, through speciations and duplications, are said to be homologs (Fitch, 2000) and are grouped into a gene family. Such gene families are in general inferred using protein sequence comparison. The evolution of a gene family can be depicted with a rooted binary tree, called a *gene tree*, where the leaves represent the homologous contemporary genes, the root their common ancestral gene and the internal nodes represent ancestral genes that have evolved through speciations and duplications.

Given a gene tree G and the species tree S of the corresponding genomes, an important question is to locate in S the evolutionary events of speciations and duplications. A *reconciliation* between G and S is a mapping of the genes (extant and ancestral) of G onto the nodes of S that induces an evolutionary scenario, in terms of speciations, duplications and losses, for the gene family described by G. In this perspective, the notion of reconciliation was first introduced in the pioneering work of Goodman *et al.* (1979) and a first formal definition was given by Page (1994) to explain the discrepancies between genes and species trees. The LCA-mapping, that maps a gene u of G onto the most recent species of S that is ancestor of all genomes that contain a gene descendant of u, is the most widely used mapping, as it depicts a parsimonious evolutionary process according to the number of duplications or duplications and losses it induces. It is generally accepted that parsimony is a pertinent criterion in evolutionary biology, but that it does not always reflect the true evolutionary history. This leads to the definition of more general notions of reconciliations between G and S (Bonizzoni *et al.*, 2005; Górecki and Tiuryn, 2006; Arvestad *et al.*, 2004) and the natural problem of exploring all evolutionary scenarios of a given gene family. Arvestad *et al.* (2004) developed a Markov Chain Monte Carlo method that explores the possible reconciliations and approximates their posterior probabilities, but the efficient exploration of all reconciliations or only the most parsimonious ones has not been addressed until now.

Our theoretical contributions are the development of algorithms to study combinatorial aspects of the *space of the reconciliations* between G and S, and more specifically its explo-

ration. These results allow us to give a first insight in the following question: is parsimony relevant to infer the true evolutionary scenario of a gene family? In Section 2, we introduce basic notations and a very general notion of reconciliation. In Section 3, we describe an algorithm that counts the total number of reconciliations or of sub-optimal ones (for the duplication cost) and an algorithm that generates a random reconciliation under the uniform distribution. In Section 4, we first define combinatorial operators that are sufficient to explore the complete space of reconciliations, and then develop an algorithm that exhaustively explores this space in optimal time. This allows us to compute the distribution of reconciliation scores in the duplication, loss, and mutation (duplication + loss) cost models. We also describe a variant of this algorithm that explores all and only all the sub-optimal reconciliations (according to an upper bound) for any of these models. There are several applications of our algorithms in functional and evolutionary genomics, such as inferring orthologs and paralogs (Fitch, 1970; Jensen, 2001), the gene content of an ancestral genome (Ma *et al.*, 2007), or in the context of Markov Chain Monte Carlo analysis of gene families (Arvestad *et al.*, 2004). In Section 5, we simulate several gene family evolutionary scenarios along two known species trees (Hahn *et al.* (2007b) and Hahn *et al.* (2007a)), with length (in time) and gene duplication and loss rates along each branch. We then study the shape of the reconciliation spaces according to the three usual cost models. Our main conclusion is that the less the cost of a reconciliation is, the more the reconciliation is similar to the real one.

2 Preliminaries

Let T be a binary tree with vertices $V(T)$ and edges $E(T)$, and such that only its leaves are labeled. Let $r(T)$, $L(T)$, and $\Lambda(T)$ respectively denote its root, the set of its leaves, and the set of the labels of its leaves. We will adopt the convention that the root is at the top of the tree and the leaves at the bottom. A *species tree* S is a binary tree such that each element of $\Lambda(S)$ represents an extant species and labels exactly one leaf of S (there is a bijection between $L(S)$ and $\Lambda(S)$). A *gene tree* G is a binary tree. From now on, we consider a species tree S, with $|V(S)| = n$ and a gene tree G such that $\Lambda(G) \subseteq \Lambda(S)$ and $|V(G)| = m$. Let $\sigma : L(G) \to L(S)$ be the function that maps each leaf of G to the unique leaf of S with the same label.

For a vertex u of T, we denote by u_1 and u_2 its children and by T_u the subtree of T rooted at u. For a vertex $u \in V(T) \setminus \{r(T)\}$, we denote by $p(u)$ its parent. A *cell* of a tree T is either a vertex of T or an edge of T. Given two cells c and c' of T, $c' \leq_T c$ (resp. $c' <_T c$) if and only if c is on the unique path from c' to $r(T)$ (resp. and $c \neq c'$); in such a case, c' is said to be a *descendant* of c. The *LCA-mapping* $M : V(G) \to V(S)$ maps each vertex u of G to the unique vertex $M(u)$ of S such that $\Lambda(S_{M(u)})$ is the smallest cluster of S containing $\Lambda(G_u)$.

Definition 1. A reconciliation between a gene tree G and a species tree S is a mapping $\alpha : V(G) \to V(S) \cup E(S)$ such that

1. (Base constraint) $\forall u \in L(G), \alpha(u) = M(u) = \sigma(u)$.
2. (Tree Mapping Constraint) For any vertex $u \in V(G) \setminus L(G)$,
   (a) if $\alpha(u) \in V(S)$, then $\alpha(u) = M(u)$.
   (b) If $\alpha(u) \in E(S)$, then $M(u) <_S \alpha(u)$.
3. (Ancestor Consistency Constraint) For any two vertices $u, v \in V(G)$, such that $v <_G u$,
   (a) if $\alpha(u), \alpha(v) \in E(S)$, then $\alpha(v) \leq_S \alpha(u)$,
   (b) otherwise, $\alpha(v) <_S \alpha(u)$.

Remark 1. This definition of reconciliation differs slightly from the classical ones as vertices of G can be mapped onto edges of S, in order to represent duplication events (see explanations below). However, it is equivalent to the definitions given by Arvestad *et al.* (2004) and Górecki and Tiuryn (2006), that are the most complete ones known so far, and it is more general than the Inclusion-Preserving mapping of Bonizzoni *et al.* (2005).

The whole set of reconciliations between a gene tree G and a species tree S is denoted $\Psi(G, S)$. A reconciliation α of $\Psi(G, S)$ implies an evolutionary scenario for the genes of G in terms of gene duplications, gene losses, and speciations. A vertex u of G that is mapped onto an edge (x, y) of S (where $x = p(y)$) represents a gene of the ancestral species $p(y)$ that has been duplicated in y. If u is mapped onto an internal vertex x of S, then this represents a gene that will be present in a single copy in the two genomes x_1 and x_2 following a speciation event that happened to x. It is important to point out that the number of reconciliations is finite. Briefly, a reconciliation α between G and S represents any birth-and-death scenario

along S such that the resulting gene tree is consistent with G and each duplication event that implies an internal vertex u of G is consistent with the mapping $\alpha(u)$. (See Figure 1).

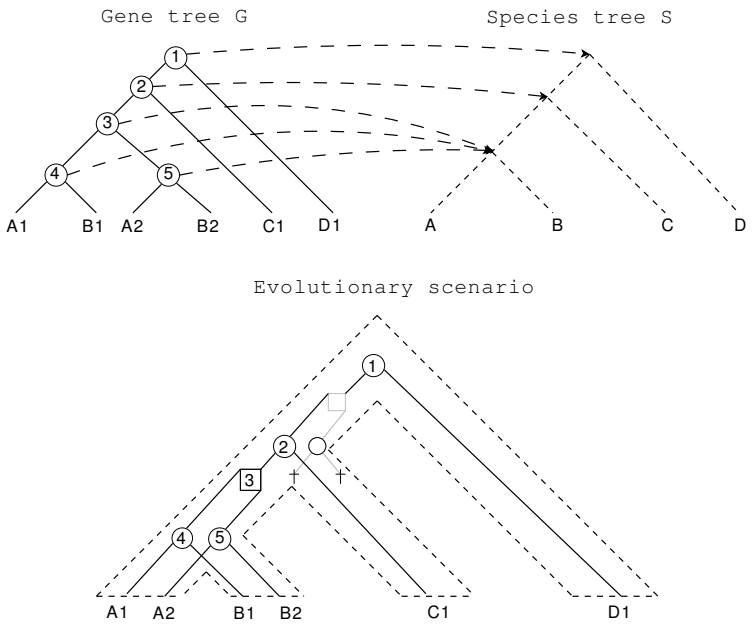

Fig. 1. Above: the species tree S has four (extant) species (A, B, C, and D). The gene tree G has six (extant) genes, where each gene belongs to one of the four species (i.e. gene A1 belongs to species A). The arrows represent the LCA-mapping between G and S. **Below:** a reconciliation between G and S. A circle (square) represents an internal vertex of G that is mapped on an internal vertex (resp. edge) of S, that is a speciation (resp. duplication) event. A cross represents a gene loss. The right lineage of the first duplication has no extant gene that descents from it, as opposite to its left lineage. We then say that this duplication is hypothetical, because it is not a useful information for the evolutionary scenario of the extant genes of G along S. Hence, such duplication is not depicted by the reconciliation.

We denote by $dup(\alpha)$ and $los(\alpha)$ respectively the number of duplications and losses induced by a reconciliation α. $dup(\alpha)$ is the number of vertices of G that are mapped onto an edge of S [3]. Given two cells $c, c' \in V(S) \cup E(S)$, where $c' <_S c$, $D(c, c')$ is the number of vertices $x \in V(S)$ such that $c' <_S x <_S c$. Also, if $c = c'$, then $D(c, c') = 0$. The number of losses associated to a vertex $u \in V(G) \setminus L(G)$ is noted l_u and equal to $D\left(\alpha(u), \alpha(u_1)\right) + D\left(\alpha(u), \alpha(u_2)\right)$ (see Ma *et al.* (2001) for example). $los(\alpha)$ is then the sum of l_u over all internal vertices u. The third constraint of Definition 1 leads to the notion of

[3] To consider duplication that precedes the first speciation event represented by $r(S)$, we can insert in S an "artificial" cell c such that $r(S) <_S c$. For the sake of clarity, and as handling such early duplications follows easily from our work, we assume here that no duplication occurs in the most ancestral species.

forced duplication, that corresponds to vertices of G that can only be mapped onto an edge of S: an internal vertex $u \in V(G) \setminus L(G)$ is said to be a forced duplication if and only if $M(u) = M(u_1)$ or $M(u) = M(u_2)$.

For a vertex $u \in V(G)$, a cell of S *covers* it if u can be mapped onto this cell according to Definition 1. The set of cells that can cover it is denoted by $A(u)$ and is defined below.

$$A(u) = \begin{cases} \{M(u)\} & \text{if } u \in L(G) \text{ or } u = r(G) \\ \{c \in E(S) : M(u) <_S c\} & \text{if } u \text{ is a forced duplication} \\ \{c \in E(S) : M(u) <_S c\} \cup \{M(u)\} & \text{otherwise} \end{cases}$$

It is important to point out that there is three mappings that are considered here: $M(u)$, $\alpha(u)$, and $A(u)$. From now on, except when indicated, the term mapping will refer to the reconciliation mapping $\alpha(u)$ of Definition 1.

Finally, combinatorial and probabilistic criteria can be used to compare the different possible reconciliations and pick one that is supposed to reflect the most the true evolution of G according to S. Three parsimonious cost models, that aim to minimize the number of genomic events, have been proposed so far: duplication (Ma *et al.*, 2001), loss (Chauve *et al.*, 2008), and mutation (duplication+loss; Ma *et al.* (2001)). Arvestad *et al.* (2004) also introduced a notion of likelihood of a reconciliation in the framework of birth-and-death processes (Kendall, 1948).

3 Counting and uniform random generation

In this section, we describe an efficient algorithm that computes a random reconciliation between G and S following the uniform distribution. This problem is important in the context of MCMC analysis for gene families, as a major issue is to analyze if the Markov chain converges to the true posterior probabilities. One of the most popular and simple tests of convergence is to run several Markov chains, each starting at a different state in the space, which motivates our random generation algorithm.

As usual in uniform random generation, it is based on a preprocessing that computes the cardinality of $\Psi(G, S)$ (Denise and Zimmermann, 1997). We first address this problem, then describe the random generation algorithm.

Counting reconciliations. For every vertex $u \in V(G)$ and cell $c \in A(u)$, we denote by $Nb(u,c)$ the number of reconciliations of G_u and S_c for which u is mapped on c. It follows immediately that $|\Psi(G,S)| = Nb\left(r\left(G\right), r\left(S\right)\right)$.

Lemma 1. *Let $u \in V(G)$ and $c \in A(u)$ be a cell that covers u. Then $Nb(u,c) = 1$ if $u \in L(G)$, and otherwise*

$$Nb(u,c) = \sum_{c_1 \in A(u_1),\ c_1 \leq_S c} Nb(u_1, c_1) \sum_{c_2 \in A(u_2),\ c_2 \leq_S c} Nb(u_2, c_2). \tag{1}$$

Proof. If $u \in L(G)$, it is obvious that $Nb(u,c) = 1$ is the number of reconciliations of G_u and S_c for which u is mapped on c. We prove equation (1). The case of pairs $c \in A(u)$ and $u \in L(G)$ are the base cases. Consider an internal vertex $u \in V(G) \setminus L(G)$, its children u_1 and u_2, and suppose that u is covered by a cell $c \in A(u)$. There are two cases: c is either a vertex or an edge of S, and Definition 1 respectively implies that $c_1 <_S c$ and $c_1 \leq_S c$. These constraints are considered by the left term in the summation of equation (1), where $c \notin A(u_1)$ if $c \in V(S)$. Hence, this left term is the total number of reconciliations for G_{u_1} and S_c when u is mapped on c. By applying the same reasoning for u_2, we obtain the right term of the summation in equation (1). The mapping of u_1 and u_2 are independent of each other, we can then conclude that equation (1) is the number of reconciliations of G_u and S_c for which u is mapped on c. □

Proposition 1. *$|\Psi(G,S)|$ can be computed in $O(mn)$ time and space.*

Proof. It follows from Lemma 1 and the obvious facts that A and M can be computed in $O(mn)$ worst-case time. □

It was shown by Chauve and El-Mabrouk (Accepted) that there is a single optimal reconciliation for the loss and mutation costs, but that there can be several ones for the duplication cost. An important question is then to count the number of these optimal reconciliations, and a more general problem is to count the number of sub-optimal reconciliations. We consider here the case of the duplication cost, and $\Psi_{dup}(G,S,\delta) = \{\alpha \in \Psi(G,S) : dup(\alpha) \leq \delta\}$ is the set of sub-optimal reconciliations, for a given bound δ.

Let $K(G)$ be the set of vertices of G that are not forced duplications, that is $K(G) = \{u \in V(G) : M(u) \in A(u)\}$. For a vertex $u \in V(G)$ and a cell $c \in A(u)$, let $f(c)$ $(d(c))$ be

the ancestor (resp. descendant) cell of c in $A(u)$, that is the cell of $A(u)$ that is the closest one to c and above (resp. below) it. The lowest (highest) cell of $A(u)$ is the one that has no descendant (resp. ancestor) cell in $A(u)$.

Definition 2. α_{min} (resp. α_{max}) is the unique reconciliation of $\Psi(G, S)$ where, for each vertex u of G, $\alpha_{min}(u)$ (resp. $\alpha_{max}(u)$) is the lowest (resp. highest) cell of $A(u)$.

Note that α_{min} corresponds to the classical LCA-mapping reconciliation as every vertex u that is not (resp. is) a forced duplication is mapped onto the LCA (resp. the edge preceding the LCA) of $\Lambda(G_u)$. Together with classical results on this LCA-mapping (see Chauve and El-Mabrouk (Accepted) and references there) and the definition of $dup(\alpha)$ and $K(G)$, we have the following properties.

Property 1. 1. α_{min} minimizes the duplication, loss, and mutation cost models.
2. For every reconciliation $\alpha \in \Psi(G, S)$, $dup(\alpha_{min}) \leq dup(\alpha) \leq dup(\alpha_{min}) + |K(G)|$.
3. For every $dup(\alpha_{min}) \leq \delta \leq dup(\alpha_{min}) + |K(G)|$, a reconciliation α is in $\Psi_{dup}(G, S, \delta)$ if and only if the number of vertex u of $K(G)$ such that $\alpha(u) \neq M(u)$ is at most $\delta - dup(\alpha_{min})$.

From these properties, we can then generalize Proposition 1 and describe a counting algorithm whose complexity is exponential in the worst-case time, but parameterized by the quantity $\delta - dup(\alpha_{min})$. In particular, Proposition 2 used with $\delta = dup(\alpha_{min})$ allows to know in polynomial time the exact number of optimal (for the duplication cost) reconciliations.

Proposition 2. *For a given $dup(\alpha_{min}) \leq \delta \leq dup(\alpha_{min}) + |K(G)|$, $|\Psi_{dup}(G, S, \delta)|$ can be computed in $O(qmn)$ time and $O(mn)$ space, where $q = \sum_{\ell=0}^{\gamma} \binom{|K(G)|}{\ell}$ and $\gamma = \delta - dup(\alpha_{min})$.*

Proof. For each ℓ in $\{0, 1, \ldots, \gamma\}$, we have to count the number of reconciliations $\alpha \in \Psi(G, S)$ such that $dup(\alpha) - dup(\alpha_{min}) = \ell$. According to Property 1.(3), we have to consider each of the $\binom{|K(G)|}{\ell}$ combinations of vertex u of $K(G)$ such that $\alpha(u) \neq M(u)$. For each of these combinations, we can adapt the equation 1 of Lemma 1 to compute its number of reconciliations. □

We will describe in Section 4.5 how to exhaustively explore the set of sub-optimal reconciliations for any of the three cost models.

Generating a random reconciliation. Algorithm 1.1 below computes a random reconciliation between G and S.

Algorithm 1.1 Uniform random generation in $\Psi(G,S)$.

1: Let α be an empty reconciliation.
2: Perform a prefix traversal of G, and let $u \in V(G)$ be the current vertex.
3: **if** $u = r(G)$ or $u \in L(G)$ **then** $\alpha(u) \leftarrow M(u)$
4: **else**
5: Let $\hat{c} \leftarrow \alpha(p(u))$.
6: {Choose randomly a cell $c \in A(u)$ such that $c \leq_S \hat{c}$}
7: Let $k \leftarrow \sum_{c \in A(u), c \leq_S \hat{c}} Nb(u,c)$
8: Generate randomly and uniformly an integer $n \in \{1,\ldots,k\}$.
9: $c \leftarrow$ lowest cell in $A(u)$ {If u is a forced duplication, then $M(u) \notin A(u)$}
10: $l \leftarrow Nb(u,c)$
11: **while** $l < n$ **do** $c \leftarrow f(c)$, $l \leftarrow l + Nb(u,c)$
12: $\alpha(u) \leftarrow c$
13: **return** α

Theorem 1. *Given a reconciliation $\alpha \in \Psi(G,S)$, Algorithm 1.1 returns α with probability $\frac{1}{|\Psi(G,S)|}$. Given the table Nb and the sets $A(u)$ for every vertex u of G, it can be implemented to run in $O(mn)$ space and $\Theta(mn)$ time in the worst case and $\Theta(m)$ time in the best case.*

Proof. Let $Pr(\alpha)$ be the probability that Algorithm 1.1 returns α. It follows immediately from lines 5-12 of the algorithm that

$$Pr(\alpha) = \prod_{u \in V(G) \setminus \{r(G) \cup L(G)\}} \frac{Nb(u, \alpha(u))}{\sum_{c \in A(u),\ c \leq_S \alpha(p(u))} Nb(u,c)} \tag{2}$$

By expanding the term $Nb(u,\alpha(u))$ in (2) according to Lemma 1, we obtain

$$Pr(\alpha) = \prod_{u \in V(G) \setminus \{r(G) \cup L(G)\}} \frac{\sum_{c_1 \in A(u_1),\ c_1 \leq_S \alpha(u)} Nb(u_1,c_1) \sum_{c_2 \in A(u_2),\ c_2 \leq_S \alpha(u)} Nb(u_2,c_2)}{\sum_{c \in A(u),\ c \leq_S \alpha(p(u))} Nb(u,c)}. \tag{3}$$

Cancellations leads to the following formula, where r_1 and r_2 are the two children of the root $r(G)$:

$$Pr(\alpha) = \frac{\prod_{u \in L(G)} \sum_{c \in A(u)} Nb(u,c)}{\sum_{c_1 \in A(r_1)} Nb(r_1,c_1) \sum_{c_2 \in A(r_2)} Nb(r_2,c_2)}, \tag{4}$$

that, together with Lemma 1, implies that $Pr(\alpha) = \frac{1}{|\Psi(G,S)|}$.

For the time complexity, suppose that for each vertex $u \in V(G)$, the size of $A(u)$ is in $\Theta(n)$. In the worst case, each vertex is mapped on the closest edge to $r\,(S)$ and the algorithm is in $\Theta(mn)$. In the best case, where each vertex is mapped onto the closest cell to $M(u)$ (lowest cell of $A(u)$), the time complexity is in $\Theta(m)$. The space complexity follows from the fact that there are $O(nm)$ pairs (u, c). □

Hence, the preprocessing time of our algorithm (computing the table Nb and the sets $A(u)$) requires $O(mn)$ time and space. However, it needs to be done once and can be used for generating several random reconciliations.

4 Exploring the space $\Psi(G, S)$

We present in this section an algorithm that visits the set of all possible reconciliations between a gene tree G (with $|V(G)| = m$) and a species tree S (with $|V(S)| = n$) in time $\Theta(|\Psi(G, S)|)$ (see Theorem 3), which gives a CAT (Constant Amortized Time) algorithm to generate $\Psi(G, S)$.

We first define combinatorial operators used to explore the whole space of reconciliations $\Psi(G, S)$ (Section 4.1). Second, we define a tree that covers $\Psi(G, S)$ and that is used by our algorithm to explore this space (Section 4.2). Third, we give some theoretical preliminaries that are required for the algorithm (Section 4.3) and then formally describe the algorithm (Section 4.4). We conclude this section by a variant of this algorithm that explores all and only all sub-optimal reconciliations for any of the three cost models (duplication, loss, or mutation) (Section 4.5).

4.1 Space exploration operators

We present in this section a combinatorial operator, called *Nearest Mapping Change* (NMC), acting on a reconciliation between a gene tree G and a species tree S. A similar operator was described by Górecki and Tiuryn (2006), in the framework of DLS-trees, to show that every DLS-tree can be obtained from the most parsimonious one. We develop it here with our definition of reconciliation and studies its properties. We first show that this operator is sufficient to explore the space of all possible reconciliations.

Definition 3. Let $\alpha : V(G) \to V(S) \cup E(S)$ be a given reconciliation between G and S, and u a vertex of $V(G) \setminus L(G)$ such that $u \neq r(G)$. Let $\hat{c}, c, c_1$, and c_2 respectively denote $\alpha(p(u))$, $\alpha(u)$, $\alpha(u_1)$, and $\alpha(u_2)$.

1. An *upward* NMC (uNMC) can be applied to u if $c <_S \hat{c}$, and if $\hat{c} \in V(S)$ and $c \in E(S)$, then $D(\hat{c}, c) > 0$. It changes $\alpha(u)$ into its ancestor cell $f(\alpha(u))$ of $A(u)$.
2. A *downward* NMC (dNMC) can be applied to u if $c_1 <_S c$, $M(u) <_S c$, and if $c_1 \in V(S)$ and $c \in E(S)$, then $D(c, c_1) > 0$ (idem for c_2). It changes $\alpha(u)$ into its descendant cell $d(\alpha(u))$ of $A(u)$.

It follows immediately from the definition of NMC operators that, given $\alpha \in \Psi(G, S)$, applying a NMC operator to a vertex u of G results in a reconciliation α' between G and S. More precisely, it can induce the following changes in the evolutionary scenario for the gene family (see Figure 2).

- Changing a speciation by a duplication (uNMC, $\alpha(u) = M(u)$).
- Changing a duplication by a speciation (dNMC, $\alpha'(u) = M(u)$).
- Moving a duplication upward (uNMC, $\alpha(u) \neq M(u)$).
- Moving a duplication downward (dNMC, $\alpha'(u) \neq M(u)$).

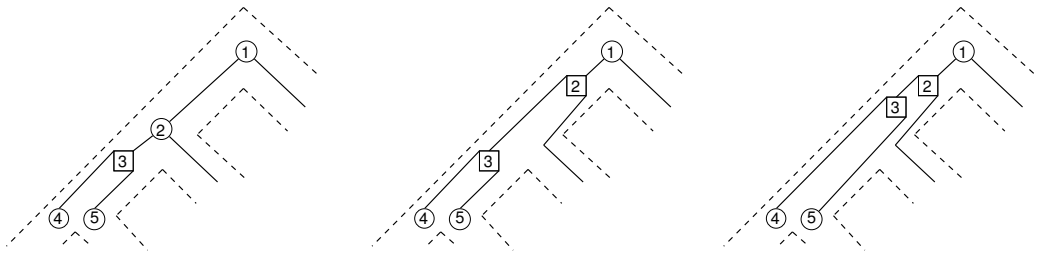

Fig. 2. Left: a section of the reconciliation depicted in Figure 1. Here, the mapping of vertex 2 forbids to move up vertex 3. Center: the vertex 2 changes from a speciation to a duplication by moving it up. Right: then, vertex 3 can be moved up and still is a duplication.

For $u \in V(G)$, and $c, c' \in A(u)$, $d_u(c, c')$ is the number of cells of $A(u)$ between c and c', where $d_u(c, c') = 0$ if and only if $c = c'$. For two reconciliations α and α', $D_{NMC}(\alpha, \alpha') = \sum_{u \in V(G)} d_u\big(\alpha(u), \alpha'(u)\big)$. We call $D_{NMC}(\alpha, \alpha')$ the *NMC distance* between α and α'. A valid

(according to Definition 3) NMC application to α can be encoded by a vertex $u \in V(G)$, that is the vertex being moved, and by a direction that is either downward or upward. We denote by $uNMC(\alpha)$ the subset of $V(G)$ such that an upward operator can be applied on any $u \in uNMC(\alpha)$ and by $uNMC(\alpha, \alpha')$ the set of vertex $u \in uNMC(\alpha)$ such that applying an upward operator on u results in a new reconciliation where the mapping of u is closer to its mapping in α'. Formally,

$$uNMC(\alpha, \alpha') = \left\{ u \in uNMC(\alpha) : d_u\left(\alpha(u), \alpha'(u)\right) = 1 + d_u\left(f(\alpha(u)), \alpha'(u)\right) \right\}.$$

For the downward operator, let $dNMC(\alpha)$ and $dNMC(\alpha, \alpha')$ be the sets defined similarly as for the upward operator. Observe that $uNMC(\alpha) \cup dNMC(\alpha)$ is the set of all possible operators for α.

The lemma below is the first step toward the definition of a combinatorial structure with vertex set $\Psi(G, S)$ (Definition 4 below), where each two reconciliations α and α' are connected by a path of minimal length $D_{NMC}(\alpha, \alpha')$ (Theorem 2 below).

Lemma 2. *Let α and α' be two reconciliations of $\Psi(G, S)$. Then,*

1. $uNMC(\alpha, \alpha') \cap dNMC(\alpha, \alpha') = \emptyset$;
2. $uNMC(\alpha, \alpha') = dNMC(\alpha', \alpha)$ *and* $dNMC(\alpha, \alpha') = uNMC(\alpha', \alpha)$;
3. *for any two nodes $u, v \in V(G)$, where $u <_G v$ and $u \in uNMC(\alpha, \alpha')$, if $\alpha(u) \leq_S \alpha(v) \leq_S \alpha'(u)$, then $v \in uNMC(\alpha, \alpha')$.*

Proof. The first two statements are obvious consequences of Definitions 1 (reconciliation) and 3 (operators) and of $uNMC(\alpha, \alpha')$ and $dNMC(\alpha, \alpha')$. For the third statement, Definition 1 implies that $\alpha(u) \leq_S \alpha(v) \leq_S \alpha'(u) \leq_S \alpha'(v)$, and because $u \in uNMC(\alpha, \alpha')$, $\alpha(u) \neq \alpha'(u)$, $\alpha(v) <_S \alpha'(v)$, and then $v \in uNMC(\alpha, \alpha')$ by definition. □

Theorem 2. *Let α and α' be two reconciliations of $\Psi(G, S)$. There exists a sequence of $D_{NMC}(\alpha, \alpha')$ operators that transforms α into α'. No shorter sequence of operators can transform α into α'.*

Proof. Assume that $\alpha \neq \alpha'$ (otherwise the statement obviously holds). In order to prove that there is a sequence of $D_{NMC}(\alpha, \alpha')$ operators that transforms α into α', we proceed in

two steps. First, we prove that $uNMC(\alpha, \alpha') \cup dNMC(\alpha, \alpha') \neq \emptyset$, and Lemma 2.(2) implies that it is sufficient to prove, without lost of generality, that $uNMC(\alpha, \alpha') \neq \emptyset$. Second, we prove that there is an "intermediate" reconciliation α'' such that i) there is sequence of upward operators of length $\sum_{u \in uNMC(\alpha,\alpha')} d_u\left(\alpha(u), \alpha'(u)\right)$ that transforms α into α'', ii) $uNMC(\alpha'', \alpha') = \emptyset$ and $dNMC(\alpha'', \alpha') = dNMC(\alpha, \alpha')$, and iii) for any $u \in uNMC(\alpha, \alpha')$ $(dNMC(\alpha, \alpha'))$, $\alpha''(u) = \alpha'(u)$ (resp. $= \alpha(u)$).

Let u be a vertex of G such that $\alpha(u) \neq \alpha'(u)$. Let $c = \alpha(u)$ and $c' = \alpha'(u)$. Without loss of generality, we can assume that $c <_S c'$ and that $\alpha(p(u)) \neq c$. If $c = M(u)$, let $y = M(u)$ and (x, y) the edge of S such that x is the parent of y. Then, by definition of the upward operator, it is allowed to map u onto (x, y), and then $u \in uNMC(\alpha, \alpha')$. Now assume that $c \neq M(u)$, which implies that $c, c' \in E(S)$. Let $c = (x, y)$ and $c' = (s, t)$. As $c <_S c'$, we have that $x \leq_S t$. If $\alpha(p(u)) \neq x$, then $u \in uNMC(\alpha, \alpha')$. Otherwise, if $\alpha(p(u)) = x$, as $\alpha'(u) \leq_S \alpha'(p(u))$, then $p(u) \in uNMC(\alpha, \alpha')$. Hence, $uNMC(\alpha, \alpha') \neq \emptyset$. Now, let u be the vertex of $uNMC(\alpha, \alpha')$ with the smallest index $id(u)$. The Lemma 2.(3) and the definition of this index imply that there is a reconciliation α'' obtained by applying $d_u\left(\alpha(u), \alpha'(u)\right)$ times the upward operator on u, and only this one, such that $uNMC(\alpha'', \alpha') = uNMC(\alpha, \alpha') \setminus \{u\}$ and $dNMC(\alpha'', \alpha') = dNMC(\alpha, \alpha')$. Because $uNMC(\alpha, \alpha')$ and $d_u\left(\alpha(u), \alpha'(u)\right)$ are both finite, repeating this process recursively with $\alpha \leftarrow \alpha''$ results in the "intermediate" reconciliation α'' described above.

We can use the same two steps described earlier with the two reconciliations α' and α'' instead of α and α', respectively. Because Lemma 2.(2) implies that $uNMC(\alpha', \alpha'') = dNMC(\alpha'', \alpha') = dNMC(\alpha, \alpha')$, the resulting sequence that transforms α' into the "intermediate" reconciliation α'' has a length equals to $\sum_{u \in dNMC(\alpha,\alpha')} d_u\left(\alpha(u), \alpha'(u)\right)$.

Finally, the concatenation of the two sequences results in a sequence of $D_{NMC}(\alpha, \alpha')$ operators that transforms α into α'. The fact that no shorter sequence exists follows immediately from the definitions of $D_{NMC}(\alpha, \alpha')$ and $uNMC(\alpha, \alpha')$, and the fact that no operator can modify $D_{NMC}(\alpha, \alpha')$ by more than 1. □

Definition 4. $\mathcal{G}(G, S)$ is the graph with vertex set $\Psi(G, S)$ and where two reconciliations are linked by an edge if and only if they differ by a single NMC.

The following results shows that although $\Psi(G,S)$ can have an exponential size, NMC operators are sufficient to define a structure on this space of polynomial diameter.

Corollary 1. *The diameter of $\mathcal{G}(G,S)$ is equal to $D_{NMC}(\alpha_{min},\alpha_{max})$ and is in $O(nm)$.*

Proof. By definition of α_{min} and α_{max} (Definition 2), for every vertex u of $V(G)$, the distance between the mapping of u in these two reconciliations is $|A(u)|-1$, and is maximal for u as $A(u)$ is the set of all possible cells that can cover u. This implies immediately that the diameter of $\mathcal{G}(G,S)$ is $D_{NMC}(\alpha_{min},\alpha_{max})$. The fact that this diameter is in $O(nm)$ follows immediately from the fact that for every m vertices u of G, $|A(u)| \in O(n)$. □

Finally, as our NMC operators are intended to explore the space of reconciliations between a gene tree and a species tree, we address now the issue of updating the classical combinatorial criteria used to evaluate a reconciliation: the following observation implies that they can be easily updated in constant time.

Property 2. Let α and α' be two reconciliations of $\Psi(G,S)$ such that α' is obtained from α by a single upward operator on a vertex $u \in uNMC(\alpha)$. If $\alpha(u) = M(u)$, then $dup(\alpha') = dup(\alpha)+1$ and $los(\alpha') = los(\alpha)+2$. Otherwise, $dup(\alpha') = dup(\alpha)$ and $los(\alpha') = los(\alpha)+1$.

4.2 A spanning tree of $\mathcal{G}(G,S)$

The exploration algorithm described in the next sections uses only upward operators and not downward operators, and from now on the term operator refers to the former.

For a vertex $u \in V(G)$, let $id(u)$ be the number of vertices that precede u according to the prefix traversal of G, where the left child u_1 of a vertex $u \in V(G) \setminus L(G)$ is visited before the right child u_2.

Definition 5. For a gene tree G and a species tree S, let $\mathcal{T}(G,S)$ be the ordered tree, with vertex set $\Psi(G,S)$, defined as follows (see Figure 3).

1. The root is the reconciliation α_{min} and its children are the reconciliations that can be obtained from α_{min} by applying a single operator on a vertex from $uNMC(\alpha_{min})$.

2. Given a reconciliation α, that differs from its parent by an operator on a vertex $u_i \in V(G)$, its children are the reconciliations that can be obtained from α by applying a single operator on a vertex $u_j \in uNMC(\alpha)$ such that $id(u_i) \leq id(u_j)$.
3. Consider a reconciliation α and two of its children α_i and α_j respectively obtained by an operator on the vertices u_i and u_j from $uNMC(\alpha)$. α_i precedes α_j in the ordered children of α if and only if $id(u_i) < id(u_j)$.

We now introduce properties that will be used to prove that $\mathcal{T}(G, S)$ is a spanning tree of $\mathcal{G}(G, S)$ (see Proposition 3 below). The first property (Property 3 below) follows immediately from Definitions 3 (operators) and 5 ($\mathcal{T}(G, S)$). The second one (Property 4) follows from Definition 5 and Property 3.

Property 3. Let α and α' be two reconciliations of $\mathcal{T}(G, S)$, where the latter is a child of the former obtained by an operator on a vertex $u \in uNMC(\alpha)$. Any reconciliation α'' in the subtree of $\mathcal{T}(G, S)$ rooted at α' is such that $\alpha'(u) \leq_S \alpha''(u)$.

Property 4. Let a vertex of $\mathcal{T}(G, S)$ labeled by the reconciliation α. Consider two children α_i and α_j of α respectively obtained by the operator on the vertices u_i and $u_j \in uNMC(\alpha)$, where $id(u_i) < id(u_j)$. Then,

1. for any reconciliation α' in the subtree of $\mathcal{T}(G, S)$ rooted at α_j, $\alpha(u_i) = \alpha'(u_i)$; and
2. the two subtrees of $\mathcal{T}(G, S)$, respectively rooted at α_i and α_j, are disjoints.

Proposition 3. *$\mathcal{T}(G, S)$ is a spanning tree of $\mathcal{G}(G, S)$.*

Proof. $\mathcal{T}(G, S)$ is a tree by definition. In order to prove it is a spanning tree of $\mathcal{G}(G, S)$, we only need to prove that every reconciliation $\alpha \in \Psi(G, S)$ appears once and exactly once as a vertex of $\mathcal{T}(G, S)$.

First, if $uNMC(\alpha_{min}, \alpha) = \emptyset$, $\alpha = \alpha_{min}$ and then α is in $\mathcal{T}(G, S)$ by definition. Otherwise, let $u \in uNMC(\alpha_{min}, \alpha)$ with the smallest index $id(u)$. By definition, there is a reconciliation α', obtained by applying $d_u\left(\alpha_{min}(u), \alpha(u)\right)$ times this operator, that is in the subtree of $\mathcal{T}(G, S)$ rooted at α_{min}, and such that $\alpha'(u) = \alpha(u)$. Then, $u \notin uNMC(\alpha', \alpha)$, and for any $u' \in uNMC(\alpha', \alpha)$, $id(u') > id(u)$. Because both $d_u\left(\alpha_{min}(u), \alpha(u)\right)$ and $uNMC(\alpha_{min}, \alpha)$

are finite, repeating this recursion with $\alpha_{min} \leftarrow \alpha'$ ends at a reconciliation α' as a vertex of this subtree and then of $\mathcal{T}(G,S)$ such that $uNMC(\alpha',\alpha) = \emptyset$ and then $\alpha' = \alpha$.

Now assume that the reconciliation α labels two vertices of $\mathcal{T}(G,S)$. By definition, these vertices are not comparable in $\mathcal{T}(G,S)$, and this is in contradiction with Property 4.(2). $\square$

4.3 Preliminaries to the algorithm

For a reconciliation α of $\mathcal{T}(G,S)$, we denote by $P(\alpha) \subseteq uNMC(\alpha)$ the list of allowed operators that can be applied to obtain the children of α, where the vertices are ordered according to the increasing value of their indexes *id*. Assuming that the children of α are visited in the order described in the Definition 5, it follows immediately that the efficient traversal of $\mathcal{T}(G,S)$ reduces to the following problem: for a reconciliation α' that is a child of α, how the list $P(\alpha')$ can be computed in constant time given the list $P(\alpha)$?

The solution to this problem is based on the two properties below (Properties 5 and 6), which both are obvious consequences of Definitions 3 (operators) and 5 ($\mathcal{T}(G,S)$).

Property 5. Let α and α' be two reconciliations of $\mathcal{T}(G,S)$ such that α' is the first child of α and is obtained by an operator on the first vertex of $P(\alpha)$, noted u. Then, the difference between $P(\alpha)$ and $P(\alpha')$ implies a constant number of vertices and is such that

1. $P(\alpha') \setminus P(\alpha) \subseteq \{u_1, u_2\}$;
2. and $P(\alpha) \setminus P(\alpha') \subseteq \{u\}$.

Property 6. Let α be a reconciliation of $\mathcal{T}(G,S)$, and consider two of its children α' and α'' respectively obtained by an operator on the vertices u' and u'' from $P(\alpha)$. Suppose that α'' (u'') is the child (resp. vertex) of α (resp. $P(\alpha)$) immediately before α' (resp. u'). Then, the difference between $P(\alpha')$ and $P(\alpha'')$ implies a constant number of vertices and is such that

1. $P(\alpha') \setminus P(\alpha'') \subseteq \{u'_1, u'_2\}$;
2. $P(\alpha'') \setminus P(\alpha') \subseteq \{u', u'', u''_1, u''_2\}$ and $u'' \notin P(\alpha')$;
3. and u' may or may not be in $P(\alpha')$.

Let α be a reconciliation of $\mathcal{T}(G,S)$, and consider any vertex u of $P(\alpha)$. For the sake of clarity, we suppose in the following proposition that the insertion (or removal) of any of the three vertices u, u_1, or u_2 into (resp. from) the list can be done in constant time.

Proposition 4. *Let α be a reconciliation of $\mathcal{T}(G,S)$. Given the list $P(\alpha)$, if the children of α are visited in the order described in the Definition 5, the list $P(\alpha')$ can be computed in constant time for any reconciliation α' that is a child of α.*

Proof. The proof is by recurrence on the i-th child of α. If α' is the first child of α, the desired result follows directly from Property 5. Otherwise, assume that α'' is the child of α immediately before α' and that $P(\alpha'')$ is known. According to Property 6, $P(\alpha')$ can be computed in constant time given $P(\alpha'')$. □

Let α be a reconciliation of $\mathcal{T}(G,S)$ and α' be one of its children that is obtained by an operator on a vertex $u \in P(\alpha)$. In the context of Proposition 4, the possible removal (insertion) of u (resp. u_1) from (resp. into) the list can obviously be done in constant time (resp. because $id(u_1) = id(u) + 1$). We now explain how to compute in constant time the position of u_2 in the list $P(\alpha')$ when u_2 is not in $P(\alpha)$. The main problem is that the number of vertices in $P(\alpha')$ that are between $id(u)$ and $id(u_2)$ is not constant. In this perspective, we have to implement the list in a particular way, which is formally described in the next section. Below, we formulate two lemmas that give a first intuition for the implementation and that are used to prove the completeness and the complexity of the algorithm that explores $\mathcal{T}(G,S)$ (see next section).

Lemma 3. *Let α be a reconciliation of $\mathcal{T}(G,S)$, u be the first vertex of $P(\alpha)$, and suppose that $u_2 \notin P(\alpha)$. Then, for any $w \in P(\alpha)$ such that $w \notin P(\alpha_{min})$, $w \neq u$, and w is the second child of $f(w)$, $id(u_2) < id(w)$.*

Proof. Suppose that $id(u_2) > id(w)$. There is two possibilities. First, if $id(w) < id(u)$, this is in contradiction with the definition of $P(\alpha)$. Otherwise, $id(u) < id(u_1) < id(w) < id(u_2)$, and then w is in the subtree G_{u_1}. However, because we assume that $w \notin P(\alpha_{min})$, there is a reconciliation α' that is along the path of $\mathcal{T}(G,S)$ that connects α_{min} and α such that $w \in P(\alpha')$. Then, $f(w)$ is used (at least one time) by an operator along this path, this

operator precedes the one(s) on the vertex u, and this is in contradiction with the definition of $\mathcal{T}(G,S)$. □

Lemma 4. *For any reconciliation α of $\mathcal{T}(G,S)$, there is a sequence of reconciliations that starts with α, ends with a reconciliation α_n that is in the subtree of $\mathcal{T}(G,S)$ rooted at α, and that respects the next three constraints. First, $P(\alpha) \cap P(\alpha_n) = \emptyset$. Second, each reconciliation of the sequence, except the first one (α), is the first child of the previous one. Third, each vertex of $P(\alpha)$ is the first vertex of $P(\alpha')$, for at least one reconciliation α' of this sequence.*

Proof. Let $u \in P(\alpha)$ be the first vertex of $P(\alpha)$ and α'_1 be the first child of α obtained by the operator on u. Because this operator can be repeated a limited number of times, each time defining the first child of the previous reconciliation, we obtain a finite sequence of reconciliations where the last one, noted α_1, is such that $P(\alpha) \setminus P(\alpha_1) = \{u\}$ (see Property 5). According to the definition of $\mathcal{T}(G,S)$, for any reconciliation α'' that is in the subtree of $\mathcal{T}(G,S)$ rooted at α_1, u is not in $P(\alpha'')$. Hence, because $V(G)$ is finite, by repeating recursively the two steps described above induces a sequence of reconciliations that ends at α_n and respects the desired constraints. □

4.4 Algorithm for the prefix traversal of $\mathcal{T}(G,S)$

We now give a complete description of an algorithm that exhaustively explores $\Psi(G,S)$ in time $\Theta(|\Psi(G,S)|)$ by a prefix traversal of the spanning tree $\mathcal{T}(G,S)$.

In the context of the prefix traversal of $\mathcal{T}(G,S)$, where a child α' of a reconciliation α is visited and is obtained by an operator on $u \in P(\alpha)$, recall that the difficulty to compute $P(\alpha')$ given $P(\alpha)$ comes from the (possible) insertion of u_2 into the list. We describe below how the list $P(\alpha)$ can be implemented to efficiently perform this insertion.

Definition 6. Let α be a reconciliation of $\mathcal{T}(G,S)$. The list $P(\alpha)$ is implemented on two sublists of $V(G)$ noted P and S and such that i) $P \cap S = \emptyset$, ii) any vertex w of S is the right child of $f(w)$ and is not in $P(\alpha_{min})$, iii) $P(\alpha) = \{w \in P : id(w) \geq id(v)\} \cup S$, where v is the first vertex of $P(\alpha)$, and iv) v is in P.

The recursion starts at the root of $\mathcal{T}(G,S)$ with the reconciliation $\alpha = \alpha_{min}$ and the ordered lists $P = P(\alpha_{min})$, that are computed during a preprocessing phase, and v as the first vertex of P and $S = \emptyset$. For a vertex $u \in V(G)$ and a cell $c \in A(u)$, recall that $f(c)$ is its ancestor cell in $A(u)$.

Algorithm 1.2 Exhaustive exploration algorithm of the space $\Psi(G,S)$.

```
1:  RecurExplore (v)
2:      u ← v
3:      while u ≠ end(P) do
4:          α(u) ← f(α(u))
5:          If u1 ∉ P and u1 ∉ L(G), then insert u1 immediately after u in P
6:          If u2 ∉ P ∪ S and u2 ∉ L(G), then insert u2 at the front of S.
7:          if u is a valid uNMC for α then
8:              RecurExplore (u)
9:          else
10:             Let x be the vertex immediately after u in P and y be the first one of S
11:             if id(x) > id(y) then
12:                 Insert y immediately before x in P and remove y from S
13:             Let v' be the vertex immediately after u in P
14:             RecurExplore (v')
15:         Undo lines 4, 5, and 6. If line 6 is undone, the removal is done on P instead of S.
16:         Let u be the vertex immediately after u in P
```

Theorem 3. *Algorithm 1.2 visits all reconciliations of $\Psi(G,S)$. Given α_{min} and $P = P(\alpha_{min})$, it can be implemented to run in time $\Theta(|\Psi(G,S)|)$ and space $O(nm)$.*

Proof.

We first address the correctness of the algorithm. We prove that the algorithm visits all reconciliations of $\Psi(G,S)$ using Proposition 3, and then that it performs a prefix traversal of $\mathcal{T}(G,S)$. In this perspective, we need to prove that the children of a given reconciliation are visited according to the order described in the definition of $\mathcal{T}(G,S)$ (Definition 5). Formally, for each recursion of the algorithm, where α is the current reconciliation, we have to prove that $P(\alpha)$ is consistent (that is with the implementation of Definition 6).

By construction, $P(\alpha_{min})$ is consistent, where $P = P(\alpha_{min})$ and $S = \emptyset$. Suppose that this is true for a recursion with $\alpha \in \mathcal{T}(G,S)$ as the current reconciliation and a vertex $v \in V(G)$

as the parameter. We then have to prove that the children of α are visited according to the usual order and that for any child α' of α, $P(\alpha')$ is consistent. Let u'' (u') be the vertex u considered at line 3 during the first (resp. second) pass of the loop and α'' (resp. α') be the new reconciliation defined at line 4.

Because of the hypothesis, $u'' = v$ is the first vertex of $P(\alpha)$ and then α'' is the first child of α according to the definition of $\mathcal{T}(G, S)$. According to Property 5, $P(\alpha)$ and $P(\alpha'')$ differ by at most three vertices, that are u'', u''_1 and u''_2. The required insertions of u''_1 and u''_2 are respectively done in lines 5 and 6. According to Lemma 3, any vertex w of S is such that $id(u''_2) < id(w)$, the usual order of the vertices of S is then conserved after the insertion of u''_2 at the front of this list. If u'' is a valid operator for the new reconciliation α'', u'' stays in $P(\alpha'')$. Otherwise, the removal of u'' from $P(\alpha'')$ is the consequence of line 13. In both case, $P(\alpha'')$ is consistent.

Moreover, by recursively applying the case of the first child described above, with α'' as the considered reconciliation instead of α, the Lemma 4 implies that each vertex $w \in P(\alpha'')$ is used at least one time as the parameter during this recursion. Then, if w was in S before the recursion on α'', w is moved from S into P at line 12 and is not moved back into S afterward. Hence, when all the children of α'' are visited, S is empty, all its vertices are in P, and the fact that $P(\alpha'')$ is consistent results from each time line 15 is performed.

Recall that α'' is the first child of α, and that $P(\alpha'')$ is consistent and S is empty immediately after the recursive call on α'' is completed (line 14). We now prove that the second child of α is α' and that $P(\alpha')$ is consistent. Remember the difference between the lists $P(\alpha'')$ and $P(\alpha')$ formally described in Property 6. First, the possible removals of the vertices u''_1 and u''_2 from the list are done in line 15. Afterward, $P(\alpha)$ is consistent and then, because u'' is the first vertex of $P(\alpha)$, the consequences of lines 16 and 4 respectively are that u' is the second vertex of $P(\alpha)$ and that α' is the second child of α. The removal of u'' from the list follows immediately (see Property 6 and Definition 6). Second, the possible insertions of u'_1 and u'_2 are respectively done in lines 5 and 6, as with the previous child α'' of α (see above), where S is in the usual order because of its emptiness before the insertion of u'_2. The possible removal of u' is similar to the one of u'' described above. Hence, α' is the second child of α and $P(\alpha')$ is consistent.

By recurrence on the i-th child of α, we can conclude that the children of α in $\mathcal{T}(G,S)$ are visited according to the usual order, that the algorithm performs a prefix traversal of $\mathcal{T}(G,S)$ and that all the reconciliations of $\Psi(G,S)$ are visited.

We now address the complexity of the algorithm. Because it performs a prefix traversal of $\mathcal{T}(G,S)$, its time complexity is in $\Theta(|\mathcal{T}(G,S)|q)$, where q is the time needed for any loop of RecurExplore. With a constant number of local variables for line 15, any loop can be done in constant time, that is without considering the recursive call. We can then conclude that the time complexity of the algorithm is in $\Theta(\left|\Psi(G,S)\right|)$.

The space complexity is first defined by the total size of both lists P and S, which is in $O(m)$, where $m = |V(G)|$. Second, for each recursive call, there is a constant number of local variables, and the maximal depth of $\mathcal{T}(G,S)$ is the diameter of $\mathcal{G}(G,S)$, which is in $O(nm)$ (see Corollary 1). We can then conclude that the space complexity of the algorithm is in $O(nm)$. □

Together with Property 2, that implies that updating the number of duplications and/or losses after a single operator can be done in constant time, this algorithm allows to compute efficiently the exact distribution of the duplication, loss and mutation costs in optimal time $\Theta(|\Psi(G,S)|)$ (see Section 5).

4.5 Exhaustive exploration of sub-optimal reconciliations

It is often interesting to consider not all reconciliations between a gene tree and a species tree, but only a subset, whose cost is close to the optimal cost, or more generally bounded. In Proposition 2, we described how to count the number of such reconciliations, for the duplication cost, that is $|\Psi_{dup}(G,S,\delta)|$. Here, we rely on a monotony property (Lemma 5 below) of path in $\mathcal{T}(G,S)$ to adapt Algorithm 1.2 in order to explore the space of sub-optimal reconciliations for any of the three cost models. Let $\Psi_{cost}(G,S,\delta) = \{\alpha \in \Psi(G,S) : cost(\alpha) \leq \delta\}$ be the considered set of reconciliations, where δ is the given bound and *cost* is one of the three usual cost models. Observe that the set $\Psi_{dup}(G,S,\delta)$ is a special case of this problem.

Lemma 5. *Let α and α' be two reconciliations of $\mathcal{T}(G,S)$, where the latter is a child of the former obtained by an operator on a vertex $u \in uNMC(\alpha)$. Any reconciliation α'' in the*

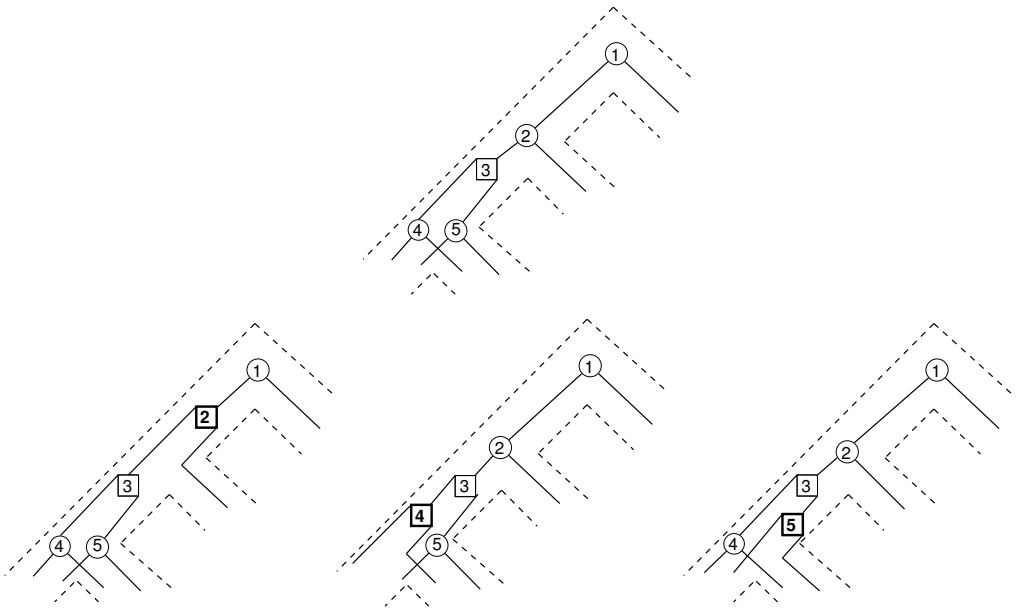

Fig. 3. The subtree of $\mathcal{T}(G,S)$ rooted at α_{min} for the trees G and S depicted in Figure 1. α_{min} and its children respectively are at the top and bottom of the figure. For each child, the vertex that has been moved upward is in boldface.

subtree of $\mathcal{T}(G,S)$ rooted at α' is such that $cost(\alpha'') \geq cost(\alpha)$, for any of the three cost models.

Proof. Obvious consequence of the definition of $\mathcal{T}(G,S)$ (Definition 5), Property 2, and Property 3.

□

Because Property 1.(1) says that α_{min} minimizes the three cost models, the considered bound is such that $\delta \geq cost(\alpha_{min})$. Let $\mathcal{G}_{cost}(G,S,\delta)$ be the subgraph of $\mathcal{G}(G,S)$ where each of its reconciliation is from $\Psi_{cost}(G,S,\delta)$. Because Algorithm 1.2 is based on the spanning tree $\mathcal{T}(G,S)$ to perform the exploration of $\Psi(G,S)$, we define below a similar combinatorial structure to explore $\Psi_{cost}(G,S,\delta)$.

Definition 7. *For a gene tree G, a species tree S, a cost model, and a bound $\delta \geq cost(\alpha_{min})$, let $\mathcal{T}_{cost}(G,S,\delta)$ be the subtree of $\mathcal{T}(G,S)$ rooted at α_{min} and defined as follows: any child α' of α_{min} in $\mathcal{T}(G,S)$ is also a child of α_{min} in $\mathcal{T}_{cost}(G,S,\delta)$ if and only if $cost(\alpha') \leq \delta$. The same rule is recursively applied to define the children of these vertices and then the whole structure of the tree.*

Proposition 5. *$\mathcal{T}_{cost}(G, S, \delta)$ is a spanning tree of $\mathcal{G}_{cost}(G, S, \delta)$.*

Proof. Obvious consequence of Property 1.(1), Definition 7, and Lemma 5. □

For a reconciliation $\alpha \in \mathcal{T}_{cost}(G, S, \delta)$, we denote by $P_{cost}(\alpha)$ the sublist of $P(\alpha)$ (see Section 4.3) that contains the allowed operators that can be applied to obtain the children of α in $\mathcal{T}_{cost}(G, S, \delta)$. We then use this list, instead of $P(\alpha)$, in the Algorithm 1.2 so that it explores the whole space $\Psi_{cost}(G, S, \delta)$.

Theorem 4. *Algorithm 1.2 can be adapted to visit all reconciliations of $\Psi_{cost}(G, S, \delta)$. Given α_{min} and $P = P_{cost}(\alpha_{min})$, it can be implemented to run in time $\Theta(|\Psi_{cost}(G, S, \delta)|)$ and space $O(nm)$.*

Proof. From the definition of $\mathcal{G}_{cost}(G, S, \delta)$ and the Proposition 5, the set of reconciliations of $\Psi_{cost}(G, S, \delta)$ is equal to the one of $\mathcal{T}_{cost}(G, S, \delta)$. Because $\mathcal{T}_{cost}(G, S, \delta)$ is a prefix tree of $\mathcal{T}(G, S)$, the Algorithm 1.2 can be adapted to explore $\mathcal{T}_{cost}(G, S, \delta)$ with the modifications described below.

In the context of the Algorithm 1.2, let α, α', and α'' respectively be the current reconciliation, the one defined by the operator on the vertex $u \in P_{cost}(\alpha)$ on line 4, and the one obtained by a second operator on the vertex u_1. Recall that because of Property 2, $cost(\alpha')$ and $cost(\alpha'')$ can be computed in constant time given $cost(\alpha)$. In line 5, if u_1 is inserted into P, then $cost(\alpha'') \leq \delta$. Moreover, if u_1 is already in P and $cost(\alpha'') > \delta$, then u_1 is removed from P. For the case of the vertex u_2 and the lists P and S, the conditions on its insertion or removal are similar as for u_1.

The correctness and the time and space complexities come from the Theorem 3 and the fact that $\mathcal{T}_{cost}(G, S, \delta)$ is a prefix tree of $\mathcal{T}(G, S)$. □

5 Experimental results

Based on two known species trees, we simulated gene family evolutionary scenarios [4], which resulted in realistic gene trees, and studied the reconciliation spaces and the effectiveness of

[4] This is done using the birth-and-death process (Kendall, 1948) along the considered species tree, where each branch has its own length (in time) and gene duplication and loss rates.

parsimonious models to retrieve the true reconciliation (the one that corresponds to the real evolutionary scenario). The first phylogeny (Figure 4) has 12 *Drosophila* species (Hahn *et al.*, 2007b, Figure 1), and the second one (Figure 5) includes only 6 mammalian species (Hahn *et al.*, 2007a, Figure 1). In the smallest phylogeny, the clade of the 3 primates has high duplication and loss rates according to the rest of the tree, and we suspected that this may generate evolutionary scenarios relatively far from the parsimonious ones. However, the results of the two species trees are similar according to both the "average shape" of the reconciliation spaces and the performance of the LCA reconciliation to retrieve the real evolutionary scenarios. Section 5.1 presents the results of the Drosophila gene families and Section 5.2 briefly summarizes the ones of the mammals.

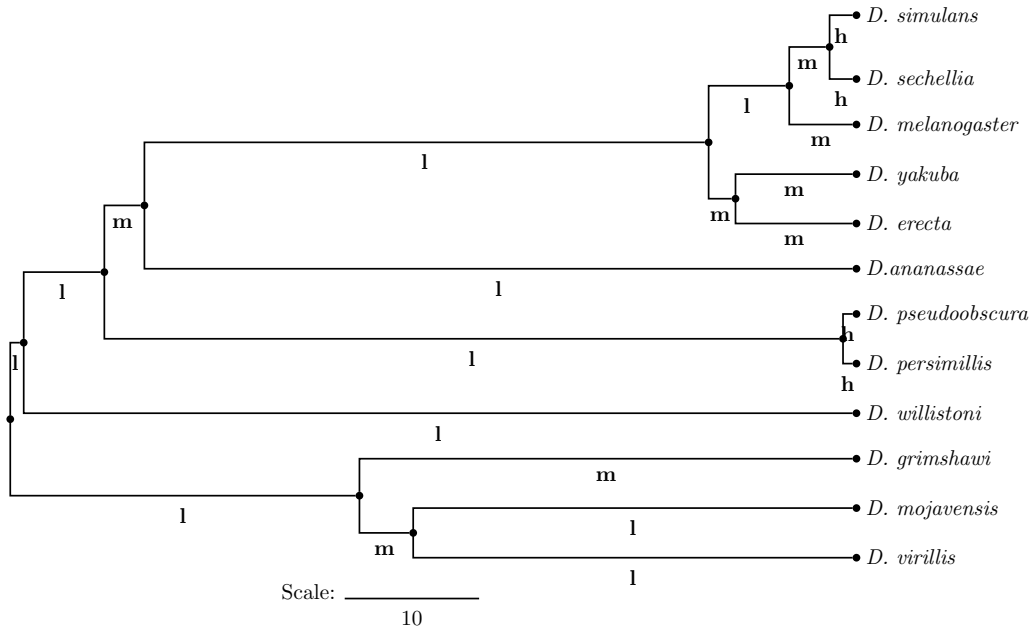

Fig. 4. Species tree for the Drosophila group (Hahn et al., 2007b, Figure 1), where divergence time is in Million Years and the gene duplication/loss rate for each branch is as follows: (h) high rate (0.0193 gene/MY), (m) medium rate (0.0022 gene/MY), and (l) low rate (0.0006 gene/MY).

5.1 Drosophila group

Along the species tree S of the 12 Drosophila (Figure 4), we generated 1000 synthetic gene trees and obtained 249 unique ones after removing multiple copies. Figure 6 describes the

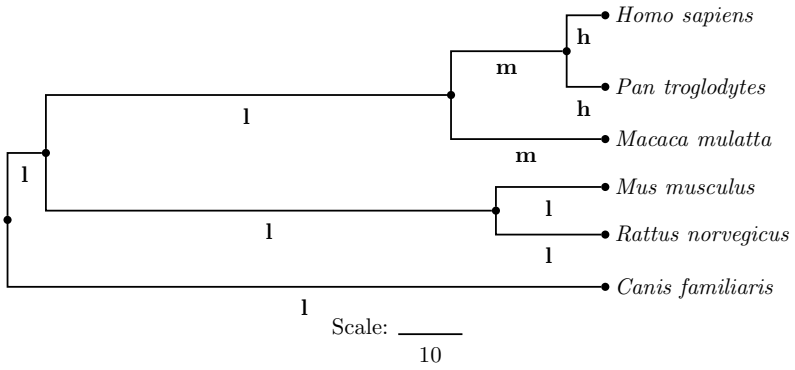

Fig. 5. Species tree for the mammalian group (Hahn et al., 2007a, Figure 1), where divergence time is in Million Years and the gene duplication/loss rate for each branch is as follows: (h) high rate (0.0039 gene/MY), (m) medium rate (0.0024 gene/MY), and (l) low rate (0.0014 gene/MY).

distribution of the size of each gene tree G, the cardinality and the diameter of the reconciliation space $\mathcal{G}(G,S)$. We also computed the average diameter over all the 249 reconciliation spaces, which is $\mu_{diam} = 34.142$.

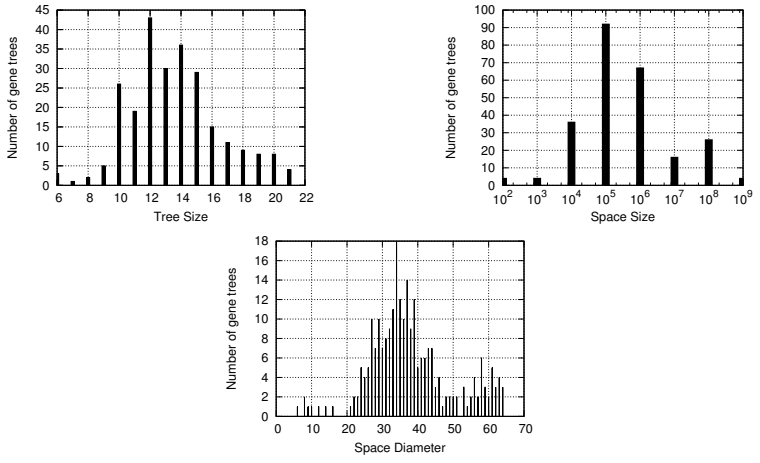

Fig. 6. Distribution of the 249 gene trees according to their number of leaves (above; left); the reconciliation space size (above; right), where a gene tree G is counted in the bar 10^i iff $10^{i-1} \leq \left|\Psi(G,S)\right| < 10^i$; and the diameter of $\mathcal{G}(G,S)$ (below), with an average diameter of 34.142.

The first experimental concern that we address here is as follows: what is the average shape of the 249 reconciliation spaces according solely to the NMC operators? Recall that α_{min} and α_{max} are located on the border of $\mathcal{G}(G,S)$ and their NMC distance is the diameter

of this space (Corollary 1). Figure 7 plots the average number $N(d)$ of reconciliations that are at a given NMC distance d to α_{min}, and we can easily observe (left plot) that $N(d)$ is inversely proportional to $|d - \mu_{diam}/2|$. This suggests that the two regions of $\mathcal{G}(G, S)$ with the lowest concentration of reconciliations are located around α_{min} and α_{max} and the highest concentrated region is equidistant to these two reconciliations. Although this implies that there is relatively (according to the size of the space) few (sub-)optimal reconciliations different from α_{min}, we can see (right plot) that their number is non-negligible. This justifies the use of our exploration algorithm to visit other parsimonious reconciliations and to consider them as an alternative for the real evolutionary scenario.

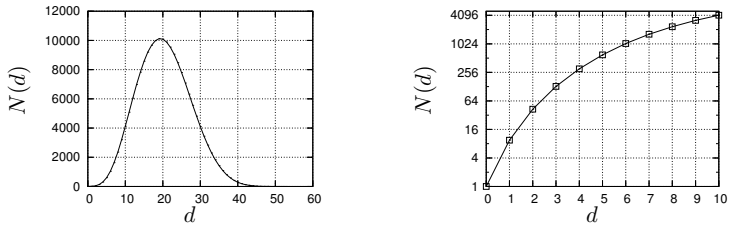

Fig. 7. Over all 249 gene trees, average distribution of the number $N(d)$ of reconciliations α such that $D_{NMC}(\alpha_{min}, \alpha) = d$, where $0 \leq d \leq D_{NMC}(\alpha_{min}, \alpha_{max})$ (i.e. the diameter).

For each of the 249 unique gene trees, we used the algorithm 1.2 to explore the whole space $\Psi(G, S)$ focusing on the duplication cost (for the loss and mutation criteria, the results are similar). For the duplication criterion, 237 gene trees have a unique global minimum, and 12 have two. In each of these 12 cases, the NMC distance between the two global minimums is one. Over all the 249 gene trees, the LCA reconciliation α_{min}, that is a global minimum, is either identical or, in the worst case, at a distance of a single NMC to the true evolutionary scenario induced by the birth-and-death and noted α_{real}.

For a reconciliation $\alpha \in \Psi(G, S)$, let $d_{\text{cost}}(\alpha) = dup(\alpha) - dup(\alpha^*)$, where α^* is a global minimum (for the duplication cost) that minimizes $D_{NMC}(\alpha, \alpha^*)$. We denote by $N(k)$ the number of reconciliations $\alpha \in \Psi(G, S)$ such that $d_{\text{cost}}(\alpha) = k$, for a given $k \in \mathbb{N}$. Figure 8 (left) shows that, on average over all gene trees, $N(k)$ is proportional to k from $k = 0$ to $k = 13$ and inversely proportional from $k = 13$ to $k = 18$. This can be explained by the following facts: the maximum value of d_{cost} is equal to the number of internal nodes u of G

that can be mapped on $M(u)$, and the average number of such nodes is 13. All this suggests that, for a given gene tree, $N(k)$ is maximized at this maximum value of $d_{\text{cost}} = k$.

We analyzed the relationship between the NMC and cost distances using the average value of $D_{NMC}(\alpha, \alpha^*)$ over all gene trees G and all reconciliations $\alpha \in \Psi(G, S)$ such that $d_{\text{cost}}(\alpha) = k$, for a given $k \in \mathbb{N}$. We also computed the number of nodes $u \in V(G)$ such that $\alpha(u) \neq \alpha_{\text{real}}(u)$. We observe that the cost distance of a reconciliation α is proportional both to the NMC distance with the closest optimal reconciliation α^* (Figure 8, center) and to how much α differs from the real reconciliation α_{real} (Figure 8, right).

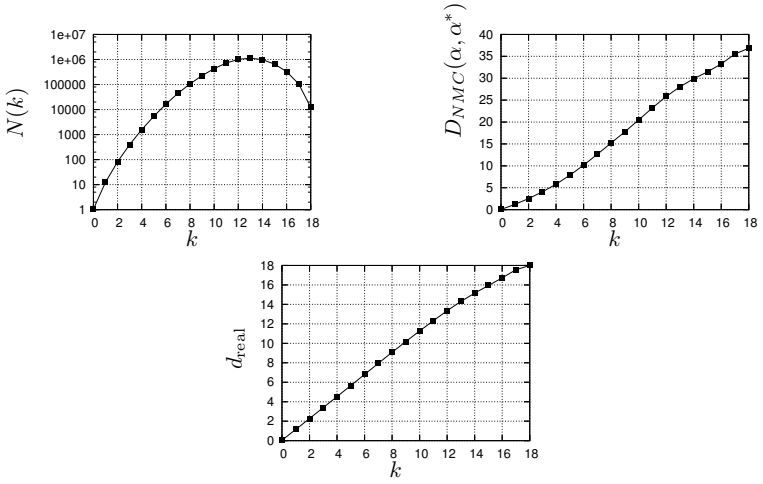

Fig. 8. Above/left: over all 249 gene trees, average distribution of the number $N(k)$ of reconciliations α such that $d_{\text{cost}}(\alpha) = dup(\alpha) - dup(\alpha^*) = k$, for $k \in \mathbb{N}$. α^* is a global minimum that minimizes the NMC distance $D_{NMC}(\alpha, \alpha^*)$. Above/right (below): the same distribution with the average value of $D_{NMC}(\alpha, \alpha^*)$ (resp. d_{real}) for all reconciliations $\alpha \in \Psi(G, S)$ such that $d_{\text{cost}}(\alpha) = k$, for a given $k \in \mathbb{N}$. The real distance d_{real} is the number of nodes u of G such that $\alpha(u) \neq \alpha_{\text{real}}(u)$.

5.2 Mammalian group

We generated 10000 gene trees along the species tree of the mammals (Figure 5), and obtained 9823 non-empty ones and 446 gene trees after removing multiple copies. As with the Drosophila gene families, over all the 446 gene trees, α_{min} is either identical or at a distance of a single NMC to α_{real}. Moreover, the number of global minimums for the duplication cost

ranges from 1 to 5, where 374 gene trees have a single minimum and 64 have 2. Figures 9 and 10 (below) present the same experimental results as the ones for the Drosophila.

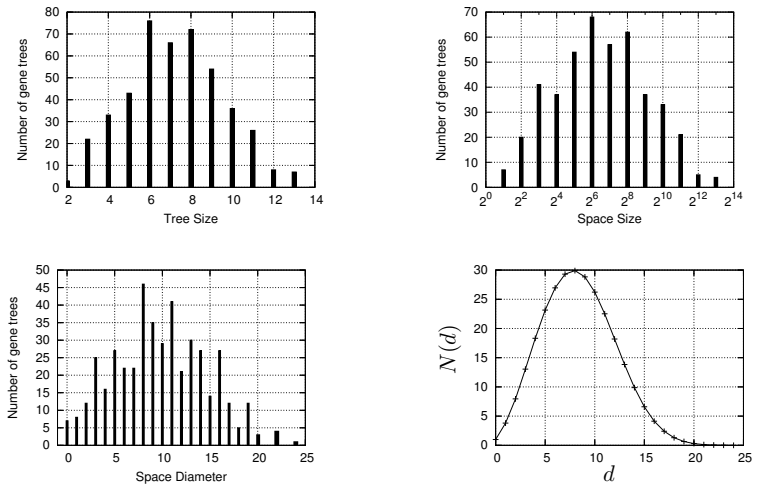

Fig. 9. Distribution of the 446 gene trees according to (above:left) their number of leaves; (above:right) the reconciliation space size, where a gene tree G is counted in the bar 2^i if $2^{i-1} \leq \left|\Psi(G,S)\right| < 2^i$; (below:left) and the diameter of $\mathcal{G}(G,S)$. Below (right), over all 446 gene trees, average distribution of the number $N(d)$ of reconciliations α such that $D_{NMC}(\alpha_{min},\alpha) = d$, where $0 \leq d \leq D_{NMC}(\alpha_{min},\alpha_{max})$ (i.e. the diameter). Over all 446 gene trees, the average standard deviation of $D_{NMC}(\alpha_{min},\alpha)$ to the mean $D_{NMC}(\alpha_{min},\alpha_{max})/2$ is 1.7263.

6 Conclusion

We described in this work several algorithms for exploring the space of all reconciliations between a gene tree and a species tree. From an algorithmic point of view, our exhaustive exploration algorithm is optimal as it requires an (amortized) constant time between successive reconciliations. Our experiments on two simulated and real datasets with low duplication/loss rates show that even in this case the number of reconciliations can be very large, but that for all three combinatorial criteria considered there are relatively few optimal or near-optimal reconciliations, always located close (in terms of NMC distance) to the LCA reconciliation. Recall that this parsimonious reconciliation is known to be a minimum for all the three cost models, and the only one for the loss and mutation criteria. This is opposite to the duplication cost, where there can be more than one optimal reconciliation, which can

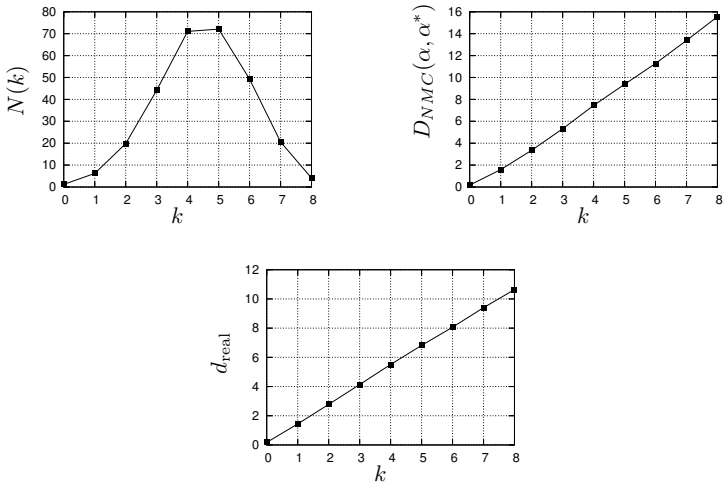

Fig. 10. Above/left: over all 446 gene trees, average distribution of the number $N(k)$ of reconciliations α such that $d_{\text{cost}}(\alpha) = dup(\alpha) - dup(\alpha^*) = k$, for $k \in \mathbb{N}$. α^* is a global minimum that minimizes the NMC distance $D_{NMC}(\alpha, \alpha^*)$. Above/right (below): the same distribution with the average value of $D_{NMC}(\alpha, \alpha^*)$ (resp. d_{real}) for all reconciliations $\alpha \in \Psi(G, S)$ such that $d_{\text{cost}}(\alpha) = k$, for a given $k \in \mathbb{N}$. The real distance d_{real} is the number of nodes u of G such that $\alpha(u) \neq \alpha_{\text{real}}(u)$.

be counted in polynomial time and explored in optimal time. We are also able to perform such controlled exploration of sub-optimal reconciliations for the three cost models, but we know to count the number of such sub-optimal reconciliations only for the duplication cost. It would be interesting to explore further the combinatorial structure of $\mathcal{T}(G, S)$ to see if it is possible to have as much control on gene losses than we currently have on duplication events.

According to our experimental results on fly and mammalian genomes, parsimonious models (based on combinatorial costs) are fully justified to infer the real evolutionary scenario for all gene families considered here. However, more in-depth studies are required to evaluate the efficiency of parsimony as well as of probabilistic methods in evaluating possible reconciliation scenarios. In this perspective, we are currently studying the impact of lower/higher duplication and loss rates on the shape of the reconciliation space when using both, parsimonious and probabilistic criteria. Other natural generalizations of the algorithms we described in the present work include handling either non-binary gene or species trees (Chang and Eulenstein, 2006; Vernot *et al.*, 2008) (or both) and attacking the more difficult problem of multiple gene duplications (Fellows *et al.*, 1998).

Acknowledgements. Cedric Chauve acknowledges the support of NSERC through an Individual Discovery Grant and of Simon Fraser University through a Startup Grant. Sylvie Hamel acknowledges the support of NSERC through an Individual Discovery Grant.

Bibliography

Arvestad, L., Berglund, A.-C., Lagergren, J., and Sennblad, B., 2004. Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. *RECOMB 2004* 326–335.

Bonizzoni, P., Vedova, G. D., and Dondi, R., 2005. Reconciling a gene tree to a species tree under the duplication cost model. *Theor. Comput. Sci.* 347, 36–53.

Chang, W.-C. and Eulenstein, O., 2006. Reconciling gene trees with apparent polytomies. *COCOON 2006* 235–244.

Chauve, C., Doyon, J.-P., and El-Mabrouk, N., 2008. Gene family evolution by duplication, speciation, and loss. *J. Comput. Biol.* 15, 1043–1062.

Chauve, C. and El-Mabrouk, N. New perspectives on gene family evolution: Losses in reconciliation and a link with supertrees. *RECOMB 2009* (Accepted) .

Denise, A. and Zimmermann, P., 1997. Uniform random generation of decomposable structures using floating-point arithmetic. *Theor. Comput. Sci.* 218, 233–248.

Doyon, J.-P., Chauve, C., and Hamel, S., 2008. Algorithms for exploring the space of gene tree/species tree reconciliations. *RECOMB-CG 2008* 1–13.

Fellows, M. R., Hallet, M. T., and Stege, U., 1998. On the multiple gene duplication problem. *ISAAC 1998* 347–356.

Fitch, W. M., 1970. Distinguishing homologous from analogous proteins. *Syst. Zool.* 19, 99–113.

Fitch, W. M., 2000. Homology - a personal view on some of the problems. *Trends Genet.* 16, 227–231.

Goodman, M., Czelusniak, J., Moore, G. W., Herrera, R. A., and Matsuda, G., 1979. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Syst. Zool.* 28, 132–163.

Górecki, P. and Tiuryn, J., 2006. Dls-trees: a model of evolutionary scenarios. *Theor. Comput. Sci.* 359, 378–399.

Graur, D. and Li, W.-H., 1999. *Fundamentals of Molecular Evolution second edition*. Sinauer Associates, Sunderland, MA.

Hahn, M. W., Demuth, J., and Han, S.-G., 2007a. Accelerated rate of gene gain and loss in primates. *Genetics* 177, 1941–1949.

Hahn, M. W., Han, M. V., and Han, S.-G., 2007b. Gene family evolution across 12 *drosophila* genomes. *PLoS Genet.* 3, e197.

Jensen, R., 2001. Orthologs and paralogs - we need to get it right. *Genome Biol.* 2, interactions 1002.1–interactions 1002.3.

Kendall, D. G., 1948. On the generalized "birth-and-death" process. *Ann. Math. Statistics* 19, 1–15.

Ma, B., Li, M., and Zhang, L., 2001. From gene trees to species trees. *SIAM J. Comput.* 30, 729–752.

Ma, J., Ratan, A., Zhang, L., Miller, W., and Haussler, D., 2007. A heuristic algorithm for reconstructing ancestral gene orders with duplications. *RECOMB-CG* 122–135.

Page, R. D., 1994. Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. *Syst. Biol.* 43, 58–77.

Vernot, B., Stolzer, M., Goldman, A., and Durand, D., 2008. Reconciliation with non-binary species trees. *J. Comput. Biol.* 15, 981–1006.

CHAPITRE 7

PRÉSENTATION DU DEUXIÈME ARTICLE

7.1 Détails de l'article

An efficient method for exploring the space of gene tree/species tree reconciliations in a probabilistic framework

Jean-Philippe Doyon, Sylvie Hamel et Cedric Chauve

Soumis à *IEEE/ACM Transactions on Computational Biology and Bioinformatics*

7.2 Partage du travail

M. Chauve, M. Doyon et Mme. Hamel ont défini le plan du papier. M. Chauve et M. Doyon ont rédigé l'article. M. Doyon a implémenté les algorithmes, s'est occupé des expériences et a défini le plan des expériences avec M. Chauve.

An efficient method for exploring the space of gene tree/species tree reconciliations in a probabilistic framework

Jean-Philippe Doyon, Sylvie Hamel, and Cedric Chauve

S. Hamel and JP.Doyon are with DIRO, Université de Montréal.

C. Chauve is with Department of Mathematics, Simon Fraser University

Abstract

Background. Inferring an evolutionary scenario for a gene family is a fundamental problem both in functional and evolutionary genomics. The gene tree/species tree reconciliation approach has been widely used to address this problem, but mostly in a parsimony framework, that considers only the reconciliation that minimizes the number of duplication and/or loss events. Recently a probabilistic approach has been developed, based on the classical birth-death process, including efficient algorithms for computing posterior probabilities of reconciliations and orthology prediction.

Results. We recently proposed an efficient algorithm for exploring the whole space of gene tree/species tree reconciliations, that we adapt here to compute efficiently, either exactly or approximately depending on the space size, the posterior probability of the visited reconciliations. We use this algorithm to analyze the probabilistic landscape of the space of reconciliations for, both, a real dataset of fungal gene families and simulated data.

Conclusion. Our results suggest that with realistic gene duplication and loss rates, a very small subset of all reconciliations needs to be explored in order to approximate very closely the posterior probability of the most likely reconciliations. For cases where the posterior probability mass is more evenly dispersed, our method allows to explore efficiently the required subspace of reconciliations.

Index Terms

Comparative genomics, species tree, gene tree, probability, reconciliation, parsimony.

I. Introduction

Genomes of contemporary species, especially eukaryotes, are the result of an evolutionary history that started with a common ancestor from which new species evolved through evolutionary events called speciations. One of the main objectives in molecular biology is the reconstruction of this evolutionary history, that can be depicted with a rooted binary tree, called a *species tree*, where the root represents the common ancestor, the internal nodes the ancestral species and speciation events, and the leaves the extant species. Other events than speciation can happen, that do not result immediately in the creation of new species but are essential in eukaryotic genes evolution, such as gene duplication and loss [13]. Duplication is the genomic process where one or more genes of a single genome are copied, resulting in two copies of each duplicated

gene. Gene duplication allows one copy to possibly develop a new biological function through point mutation, while the other copy often preserves its original role; another outcome can be subfonctionalization, where each copy develops a specific subfunction of the function of the ancestral gene. A gene is considered to be lost when the corresponding sequence has been deleted by a genomic rearrangement or has completely lost any functional role (i.e. has become a pseudogene). (See [13] for example). Other genomic events such as lateral gene transfer, that occurs mostly in bacterial genomes, will not be considered here.

Genes of contemporary species that evolved from a common ancestor, through speciations and duplications, are said to be homologs [9] and are grouped into a gene family. Such gene families are in general inferred using protein sequence comparison and clustering methods. The evolution of a gene family can be depicted with a rooted binary tree, called a *gene tree*, where the leaves represent the homologous contemporary genes, the root their common ancestral gene and the internal nodes represent ancestral genes. Given a gene tree G and the species tree S of the corresponding genomes, a fundamental question is to infer the evolutionary history that led to G, which amounts to locate in G the evolutionary events of speciations and duplications and the branch or nodes of S where they occurred. Inferring the evolutionary scenario for a gene family has applications in phylogenomics [20], functional genomics, especially for the identification of orthologous genes [21], or comparative genomics and paleogenomics [18].

A *reconciliation* between G and S is a mapping of the genes (extant and ancestral) of G onto the nodes of S that induces such an evolutionary scenario, in terms of speciations, duplications and losses, for the gene family described by G. The notion of reconciliation was first introduced in the pioneering work of Goodman *et al.* [10] and a first formal definition was given in [19] to explain the discrepancies between gene and species trees. The Minimum Parsimony Reconciliation (MPR from now) is defined by the mapping of each gene u of G onto the most recent species of S that is ancestor of all genomes that contain a gene descendant of u (called the LCA-mapping, see Figure 1). It has been shown to be the reconciliation that induces the unique evolutionary scenario that minimizes both the number of gene duplication events and gene loss events [8]. It is generally accepted that parsimony is a pertinent criterion in evolutionary biology, but that it does not always reflect the true evolutionary history. This leads to the definition of more general notions of reconciliations between G and S [6], [11], [2],

[8] and the natural problem of exploring all, or many, evolutionary scenarios for a given gene family [8].

In the context of probabilistic orthology analysis, an important breakthrough is due to Sennblad *et al.* [21], who developed a method for computing the posterior probability that a given pair of genes of a gene family tree G are orthologous, according to the definition of Fitch [9] (two extant genes are orthologs if their most recent ancestor (LCA) in G is a speciation vertex). For fixed gene duplication and gene loss rates along the branches of S, the method computes in polynomial time the exact posterior probability that a vertex of G is a speciation, and a Markov Chain Monte Carlo, based on a Bayesian framework, is used to integrate over the rates (that is given the trees G and S). With these two steps, their method estimates the posterior probability of orthology relationships between extant genes of G, with prior on the duplication and loss rates. They focus their study on orthology relationships between pairs of genes, or equivalently on the probability that a given vertex of G represents a speciation event, but they also provide algorithms to compute the posterior probability of a given reconciliation. With regard to this, their experimental results suggest that, for low gene duplication and loss rates, the posterior probability mass is dominated by the MPR, but using simulated data, they show that it is not rare that the MPR implies wrong orthology relationships and that their probabilistic framework improves on the traditional parsimony approach. They also argue against the explicit exploration of the whole space of all reconciliations, due to its possible huge size, and they develop in [2] efficient, but sophisticated, dynamic programming algorithms to both compute the posterior probability of a given reconciliation and sample reconciliations according to the posterior distribution. Both algorithms have a time complexity that is quadratic in the size of G and linear in the size of S.

The goal of the present study is to complement the recent work of Sennblad *et al.* by (1) providing more efficient algorithms to explore a subspace of the space of all reconciliations and (2) analyzing the shape of the space of all reconciliations between a gene tree and a species tree according to a probabilistic criterion, when gene duplication and loss rates are known. Our contribution is twofold. First, we extend the algorithm described in [8], to explore the whole space of the reconciliations between a gene tree and a species tree, in order to compute or estimate efficiently the posterior probability of the set of all visited reconciliations, whether this set represents the whole space or only a subspace if the latter is too large. This algorithm improves

on the algorithms described in [2], as computing the probability for the visited reconciliations can be done in average linear time, both in the size of G and in the size of S, for each visited reconciliations. This makes possible to explore larger sets of reconciliations, a property we use in our experiments to study the probabilistic landscape of real and simulated datasets. We study a real dataset of fungal gene families and several simulated datasets, obtained with moderate but realistic gene duplication and loss rates, and we show that, in general, a small subset of reconciliations located close to the MPR cover the whole probability mass. Hence, the posterior probability of the plausible evolutionary scenarios of a gene tree G according to S can be estimated efficiently with very good precision.

II. Material and Methods

A. Trees and reconciliations

Let T be a binary tree with vertices $V(T)$ and edges $E(T)$, with labeled leaves. Let $r(T)$, $L(T)$, and $\Lambda(T)$ respectively denote its root, the set of its leaves, and the set of the labels of its leaves. For a vertex u of T, we denote by u_1 and u_2 its children and by T_u the subtree of T rooted at u. For a vertex $u \in V(T) \setminus \{r(T)\}$, we denote by $p(u)$ its parent and by $(p(u), u)$ the edge of T with $p(u)$ as the departure vertex.

Let G be a gene tree and S be a species tree, with $\Lambda(G) \subseteq \Lambda(S)$. Following [8], a reconciliation between G and S maps each vertex u of G onto either a vertex or an edge of S and is denoted $\alpha : V(G) \rightarrow V(S) \cup E(S)$. (see Appendix I for a complete formal definition and Figure 1 for an illustration). If $\alpha(u) = x \in V(S)$, u represents a gene that will be present in single copy in the two genomes x_1 and x_2 following a speciation event that happened to x. Otherwise, $\alpha(u) = (p(x), x) \in E(S)$ and u represents a gene of the ancestral species $p(x)$ that has been duplicated in the descendant species x. For simplicity, we assume that $r(G)$ has a singe copy in the ancestral genome $r(S)$, as the case where $r(G)$ is duplicated in the ancestor species $r(S)$ can be handled easily by adding a branch prior to $r(S)$.

B. Exploring the space of reconciliations

Following [8], we explore reconciliations between G and S according to an exploration tree denoted $\mathcal{T}(G, S)$ (see Figure 15 in Appendix I), whose root is the MPR, also denoted α_{min}

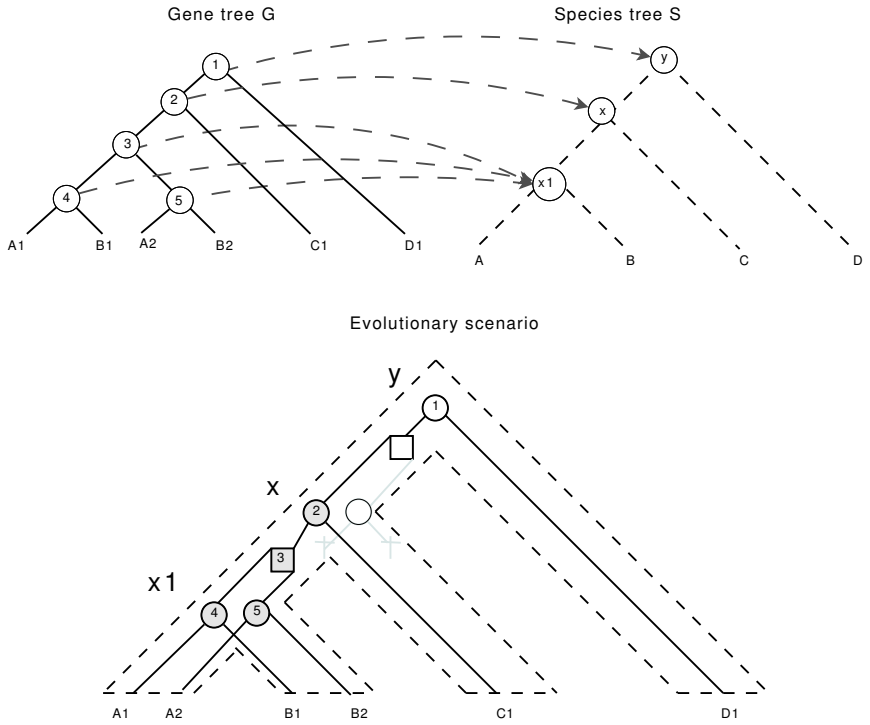

Fig. 1. **Top:** the species tree S has four (extant) species (A, B, C, and D). The gene tree G has six (extant) genes, where each gene belongs to one of the four species (i.e. gene A1 belongs to species A). The arrows represent the LCA-mapping between G and S. **Bottom:** Minimum Parsimony Reconciliation between G and S induced by the LCA mapping. A circle (square) represents an internal vertex of G that is mapped on an internal vertex (resp. edge) of S, that is a speciation (resp. duplication) event. A cross represents a gene loss. The right lineage of the first duplication has no extant gene that descents from it, as opposite to its left lineage. We then say that this duplication is hypothetical, because it is not a useful information for the evolutionary scenario of the extant genes of G along S. Hence, such duplication is not depicted by the reconciliation.

from now, and such that two reconciliations that are incident in $\mathcal{T}(G,S)$ differ only by a single Nearest Mapping Change (NMC). An NMC is an operator introduced in [8] which transforms a first reconciliation into a second one by moving the mapping of an internal vertex of G by one vertex/edge of S either downward or upward. (See Figure 2 for an illustration).

$D_{NMC}(\alpha_{min},\alpha)$ corresponds to the depth in $\mathcal{T}(G,S)$ of a given reconciliation α, i.e. the minimum number of NMC needed to transform α into α_{min}. $D(\mathcal{T}(G,S))$ denotes the maximum depth of a reconciliation of $\mathcal{T}(G,S)$; for a given depth d, $\mathcal{T}_d(G,S)$ is the subtree of $\mathcal{T}(G,S)$ rooted at α_{min} and formed of each reconciliation $\alpha \in \mathcal{T}(G,S)$ such that $D_{NMC}(\alpha_{min},\alpha) \leq d$.

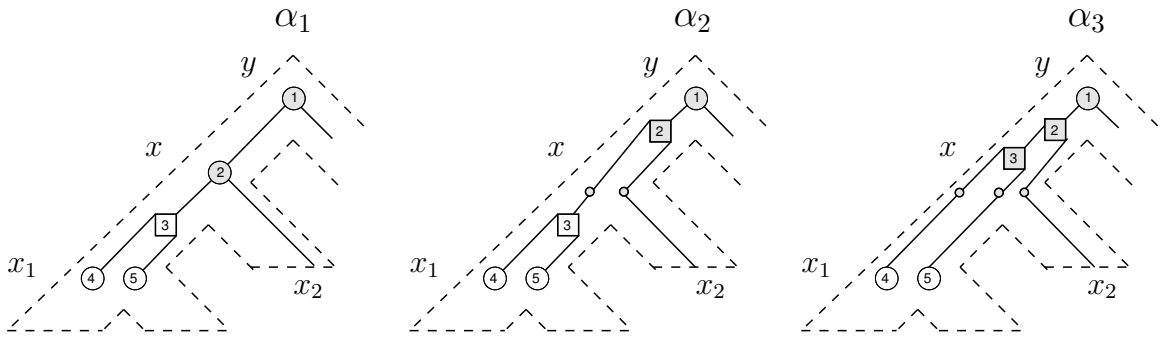

Fig. 2. **Left:** a section of the reconciliation depicted in Figure 1. Here, the mapping of vertex 2 forbids to move up vertex 3. **Center:** the vertex 2 changes from a speciation to a duplication by moving it up. **Right:** then, vertex 3 can be moved up and still is a duplication.

C. The probabilistic framework

We now assume that for each branch of S, the length of the branch, as well as a gene duplication rate and a gene loss rate are known.

For a reconciliation α between G and S, $P(G, \alpha)$ denotes the probability that a single gene of the ancestral species $r(S)$ evolves along S and generates a gene tree that is isomorphic to G following the evolutionary scenario induced by α. Hence, the probability $P(G)$ of generating G along S is the sum of $P(G, \alpha)$ over all reconciliations α between G and S, and the posterior probability of a reconciliation α, given G, is $P(\alpha|G) = \frac{P(G,\alpha)}{P(G)}$.

Given a set $\mathcal{T}$ of K reconciliations $\{\alpha_1, \ldots, \alpha_K\}$, the probability of G based on these reconciliations is defined as follows: $P_{\mathcal{T}}(G) = \sum_{i=1,\ldots,K} P(G, \alpha_i)$. If $\mathcal{T} = V(\mathcal{T}(G, S))$, then $P_{\mathcal{T}}(G)$ is the exact probability of G. Otherwise, it is a $\mathcal{T}$-approximation.

Given a subtree $\mathcal{T}_d(G, S)$ of $\mathcal{T}(G, S)$, the sum of the posterior probability of each reconciliation α located in $\mathcal{T}_d(G, S)$ is called its probability mass and is defined as follows $P(\mathcal{T}_d(G, S)|G) = \sum_{\alpha \in \mathcal{T}_d(G,S)} P(\alpha|G)$.

We now present our main algorithmic results.

Theorem 1: Let G be a gene tree and S a species tree, with $|V(S)| = m$ and $|V(G)| = n$, and $\mathcal{T}$ a connected subtree of $\mathcal{T}(G, S)$ containing K reconciliations between G and S. Then,

1) computing the exact posterior probability $P(\alpha|G)$ for all K reconciliations α of $\mathcal{T}$ can be done in time and space $O(mn^2 + K(m + n))$,

2) computing the $\mathcal{T}$-approximation $P_{\mathcal{T}}(\alpha|G)$ of the posterior probability for all K reconciliations α of $\mathcal{T}$ can be done in time and space $O(mn + K(m+n))$.

Proof: The computation scheme is based on

- an $O(mn^2)$ algorithm to compute $P(G)$ described in [2, Theorem 6.9], if one is interested in computing the exact posterior probabilities of the visited reconciliations,
- the exploration of $\mathcal{T}$ starting at α_{min}, using the general scheme described in [8], that requires time $O(K)$,
- the computation of the probability $P(G, \alpha_{min})$ in $O(mn)$ time [2, Theorem 5.21],
- Lemma 1 below that states that the probability of a newly visited reconciliation can be obtained from the previous one in $O(m+n)$ time .

All together, this gives the stated complexities ■

If one is interested in the posterior probability distribution of a subset of reconciliations, this result improves, from an efficiency point of view, on the algorithm presented in [2], that has an $O(mn)$ time complexity to compute $P(G, \alpha)$ for a given reconciliation α, while Lemma 1 below shows it can be updated in $O(m+n)$ time after a single NMC. Moreover, provided the $\mathcal{T}$-approximation of the posterior probability $P(\alpha|G)$ is a good approximation, the computational cost gain of our method makes it valuable in the context where duplication and loss rates are not known but are computed through an MCMC approach, where a large number of posterior probability computations is required (see [21]). The precise experimental results we present in the Results section, especially for very large sets of reconciliations, were obtained thanks to this computational complexity improvement.

Lemma 1: Given two reconciliations α and α' of $\mathcal{T}(G, S)$ that are separated by a single NMC, the time complexity to compute $P(G, \alpha')$ given $P(G, \alpha)$ is $O(m+n)$.

Proof: The proof relies on the recursion described in [2] to compute the probability of generating (G, α) (called a reconciled tree from now) along S, and we outline here its main properties.

Following the notation of [2] and given an internal vertex x of S and a vertex u of G such that $\alpha(u) = x$, $r_V(x, u)$ denotes the probability that the evolution of the gene u along the species tree S_x results in the reconciled tree (G_u, α_u), where α_u is the reconciliation between G_u and S_x that is induced by α. The computation of this probability is based on Recursion 1 below and

on the four components:

- $r_A(x_1, u)$ is the component of $r_V(x, u)$ for the edge (x, x_1) and the subtree S_{x_1} ([1]).
- $Q_{x_1}(l)$ is the probability that the evolution of u along this edge generates l non-ghost genes[2] that belong to x_1.
- If the subtree of G induced by this evolution and rooted at u is denoted $G_{u||x_1}$, which is called a sliced subtree (see Figure 1: the colored vertices of G correspond to $G_{u||x_1}$, where u denotes the speciation vertex 2), $h(G_{u||x_1})$ is the probability that the sliced subtree of G generated by this evolution is isomorphic to $G_{u||x_1}$, assuming that all such subtrees are equiprobable.
- Finally, $W(x_1, u)$ corresponds to the number of ways the reconciled trees rooted at the leaves of $G_{u||x_1}$ can be exchanged to produce a reconciled tree that is isomorphic to (G_u, α_u).

Recursion 1: ([2, Recursion 5.20]) The probability of a reconciled tree (G, α). Let $x \in V(S)$,

$$r_V(x,u) = \begin{cases} 1 & \text{if } x \in L(S), u \in L(G), u \in \alpha^{-1}(x) \\ r_A(x_1,u)\ r_A(x_2,u) & \text{otherwise} \end{cases}$$

$$r_A(x,u) = \begin{cases} Q_x(0) & \text{if } L(G_{u||x}) = \emptyset \\ Q_x(|L(G_{u||x})|)\ W(x,u)\ h(G_{u||x}) \prod\limits_{v \in L(G_{u||x})} r_V(x,v) & \text{otherwise.} \end{cases}$$

According to [2, Theorem 5.21], if the root of G is mapped on the root of S (that is $\alpha(r(G)) = r(S)$), the probability that an evolutionary scenario produces the reconciled tree (G, α) is $P(G, \alpha) = r_V(r(S), r(G))$ and can be computed in time $O(mn)$, where $|V(S)| = m$ and $|V(G)| = n$ ([3]).

We now prove the lemma. Consider the three reconciliations of Figure 2, and let v and u, and a_1 and a_2 respectively denotes the vertices labeled 1 and 2, and the left and right artificial

[1] In $r_V(x, u)$ (resp. $r_A(x_1, u)$), the r stands for reconciliation and the subscript V (resp. A) indicates that it starts at the considered vertex (resp. arc) of S.

[2] Given an evolutionary scenario for a gene family G, an ancestral gene that belongs to an ancestral species x of S that goes extinct before reaching the leaves of S_x is called a ghost gene. Hence, a non-ghost gene in such an evolutionary scenario is a gene that has at least one descendant among the extinct genes of G.

[3] The case where $r(G)$ is not mapped on $r(S)$, but is mapped on an edge or another vertex of S, is similar to the one described above and we omit it for simplicity.

vertices that are mapped on species x in α_2. Given that the probability $P(G, \alpha_1)$ is computed and the NMC that moves u upward is applied on α_1 and results in α_2, the new probability $P'(G, \alpha_2)$ is updated as follows: $r'_A(x_1, a_1) = r_A(x_1, u)$, $r'_A(x_2, a_1) = Q_{x_2}(0)$, $r'_A(x_1, a_2) = Q_{x_1}(0)$, and $r'_A(x_2, a_2) = r_A(x_2, u)$. As the sliced subtree $G_{v||x}$ changes following the NMC applied on u, both $h(G_{v||x})$ and $W(x, v)$ have to be recomputed. According to [2], the time complexities to compute $Q_{x_1}(0)$ and $Q_{x_2}(0)$ and to compute $h(G_{v||x})$ and $W(x, v)$ respectively are $O(m)$ and $O(n)$. It is then immediate that the time complexity to compute $P'(G, \alpha_2)$ given $P(G, \alpha_1)$ is $O(n + m)$. For the three others NMCs, the proof is similar. ■

D. A dataset of fungal gene families

We considered 12 fungal genomes, whose species tree is given in Figure 16 of Appendix II. Althought there is debate (see [16]) over the true phylogeny, we argue that the signals of our experiments are sufficiently clear so that a phylogenetic error would have a small impact. For these 12 species, we considered 1278 gene trees from [24], which originally contains a total of 20598 gene families from 12 fungal genomes. After keeping only a single copy for sets of isomorphic gene trees, 1543 gene trees remain. From these 1543 gene trees, we conserved the 1278 ones whose reconciliation space contains between 10 and 10^7 reconciliations, called from now the A-trees. The distribution of all the A-trees G according to the number of genes and species present in G and to the size and depth of $\mathcal{T}(G, S)$ are depicted in Figure 17 of Appendix II.

The branch length (in Million of Years) of the species tree S were computed by a Bayesian framework that takes as input homologous DNA sequences and assumes that the rates of nucleotide substitutions is constant over any branch, but it can differ among the branches (relaxed molecular clock) [22]. The duplication and loss rates along the branches of S were estimated by CAFE [5], which takes as input the branch lengths and the profiles (i.e. the number of genes per species) of the 20598 fungal gene families, and performs an Expectation-Maximization algorithm to find the rates that maximizes the probability of the observed profiles. We performed several runs of the algorithm, without regrouping the branches of S into rate classes, and observed that it converges to the same rate for each branch of S.

E. Datasets of synthetic gene trees

We computed synthetic sets of gene trees obtained from the fungi species tree S, with three different rates of gene duplication and loss. We defined three classes of rates using 1, 1.4, and 1.8 as the Increasing Factor (I.F. for short) on the previous ones [4] and, for each I.F., we generated synthetic gene trees along S using a birth-and-death process starting from a single ancestral gene. Given length and duplication and loss rates along each branch of S, the generation of a gene tree starts with a single gene u at $r(S) = x$, simulates its evolution along the branch (x, x_1) (resp. (x, x_2)) using the birth-and-death [17] process, which computes the probability that u has n descendants that belong to x_1 (resp. x_2), and recursively repeats this process throw the extant species of S. Afterward, the synthetic gene tree G is obtained by the removal of each ghost gene, and the resulted evolutionary scenario corresponds to the real reconciliation α_{real} between G and S. The I.F. 1.8 is the highest one considered because it induces 14 gene trees that cover the 12 genomes, as opposite with an I.F. of 2 for which none such tree were generated. For each considered I.F., the considered synthetic gene trees were obtained by 2000 such simulations and the conservation of all the unique A-trees, after the removal of multiple copies. We then obtained 1051 A-trees with I.F. 1, 1025 with I.F. 1.4 and 924 with I.F. 1.8. The characteristics of these genes trees, in terms of number of genes, species, and reconciliation spaces are depicted in Figure 18 (see Appendix II).

It is important to point out that the birth-and-death process does not take into consideration some evolutionary properties: genes that belong to the same protein complexes tend to have similar evolutionary scenarios [12]; whole genome or large segmental duplications (such events happened in yeasts evolution [23], [25]); after duplication of a gene, the evolution of one copy tends to affect the other [7]; the duplication and loss rates may not be constant along the branches of S. These discrepancies may be the reason why the distributions of the number of genes and species present in a gene tree differ between the trees generated by the rates estimated by CAFE and the 20598 real ones mentioned before, which are used by CAFE to estimates these rates (see Figure 17 and 18). However, these synthetic gene trees provide valuable and reasonable datasets to study the probabilistic landscape of the space of reconciliations, which is our goal in the present work.

[4] That is each (duplication and loss) rate along any branch of S is multiplied by the considered factor.

III. RESULTS

The main concerns of our experiments is to sustain the two following observations.

1) The probability mass of the whole tree of reconciliations is technically covered (i.e. approximated with very high precision) by a small set $\mathcal{T}$ of reconciliations located close to α_{min}.
2) For a given reconciliation α that belongs to $\mathcal{T}$, the approximation $P_{\mathcal{T}}(\alpha|G)$ is a very precise approximation of the exact posterior probability $P(\alpha|G)$.

We first illustrate these facts on a real dataset of fungal gene families, then we show they remain true with synthetic gene trees obtained with higher duplication and loss rates.

A. Fungal gene trees

To begin, let us describe two observations regarding the MPR α_{min} over the 1278 A-trees.

1) In 1276 cases, the MPR is the most probable reconciliation, and in the two remaining cases, the most probable reconciliation α^* is one NMC away from α_{min}.
2) The average probability $P(\alpha_{min}|G)$ is 0.94672 with a standard deviation of 0.03906, and varies from 0.98 when the depth of $\mathcal{T}(G,S)$ is 5 or less to 0.88 when the depth of $\mathcal{T}(G,S)$ is between 55 and 60 (See Figure 3). Note however that it happens that α_{min} has a significantly lower posterior probability, pointing at gene families where the probability mass is more evenly distributed.

Next, we recorded the variation of the exact posterior probability $P(\alpha|G)$ with respect to the depth of α. Our results, described in Figure 4 below show that, on the average, this probability decreases very quickly with the depth.

These observations agree with previous ones for datasets with low gene duplication and loss rates [21] and give a clear insight on the fact that the probability mass of the whole space tree is concentrated around α_{min}, and the depth of this tree has a relative small impact (although not negligible) on these probabilities. Note also that, although the standard variation of the average probability of the MPR is relatively low, this does not prevent some cases where the MPR has a relatively low probability a posteriori (down to less than 0.3 in some cases). This suggests that, for some gene families, even with low rates, exploring a relatively large subspace of the space of reconciliations is worth it.

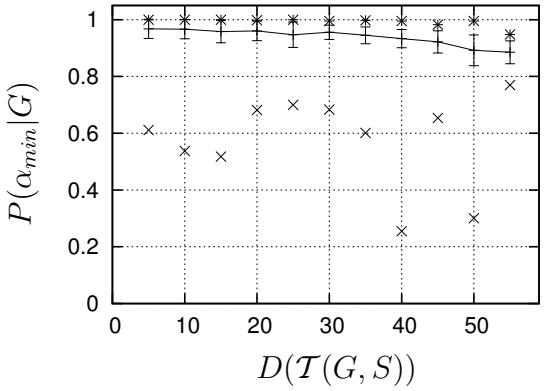

Fig. 3. Over all 1278 A-trees G, average posterior probability of α_{min} (y axis) according to the depth of $\mathcal{T}(G,S)$ (x axis; gene trees are grouped by classes for which the depth divided by 5 is equal), together with the standard deviation, the maximum and minimum probabilities.

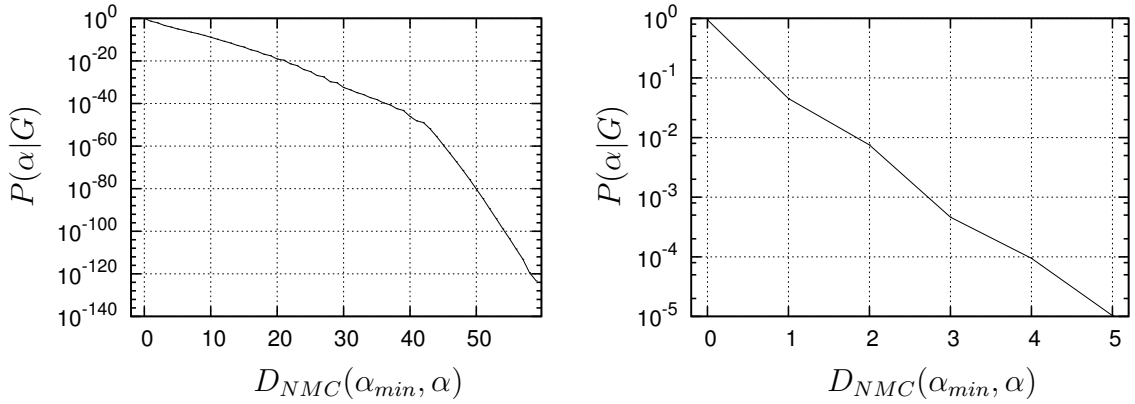

Fig. 4. Over all 1278 A-trees G, (**left**) average probability (y axis) of each reconciliation $\alpha \in \mathcal{T}(G,S)$ located at the same depth in $\mathcal{T}(G,S)$ (x axis), and (**right**) zoom on the reconciliations of depth at most 5.

Next, we analyzed the depth required to explore sufficiently many reconciliations to capture most of the probability mass. Let $d^{\circ}(\mathcal{T}(G,S))$ be the smallest depth for which the probability mass of $\mathcal{T}_{d^{\circ}(\mathcal{T}(G,S))}(G,S)$ is technically equal to one (that is according to the usual C++ floating point precision [15], the sum of the probabilities of the reconciliations in this subspace is 1). Figure 5(left) below plots $d^{\circ}(\mathcal{T}(G,S))$ against $D(\mathcal{T}(G,S))$. It is clear that, even for very deep reconciliation spaces, a small depth is enough to capture the probability mass: for $D(\mathcal{T}(G,S)) = 55$, the average value of $d^{\circ}(\mathcal{T}(G,S))$ is only 4.5, and the maximum required depth is only 9,

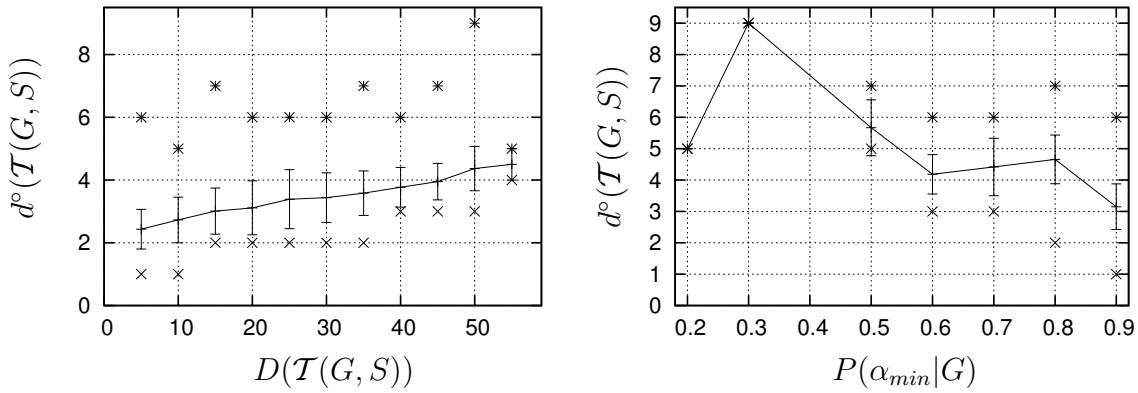

Fig. 5. Over all 1278 A-trees G, **(left)** average depth $d^\circ(\mathcal{T}(G,S))$ (y axis) for each space tree $\mathcal{T}(G,S)$ for which $D(\mathcal{T}(G,S))$ belongs to the same depth range (x axis), together with standard deviation, minimum and maximum depth, and **(right)** average depth $d^\circ(\mathcal{T}(G,S))$ (y axis) according to the probability of α_{min} (x axis; gene trees are grouped by classes for which $P(\alpha_{min}|G)$ divided by 0.1 is equal), together with the standard deviation, minimum and maximum depth.

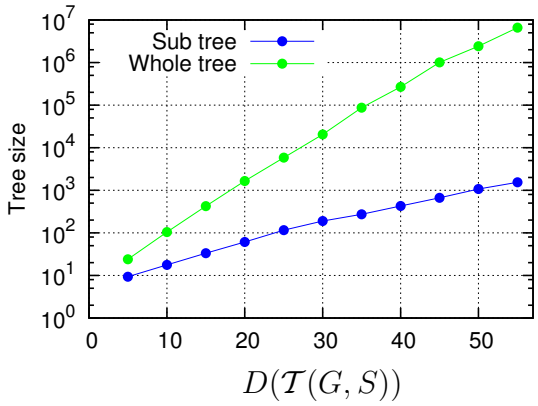

Fig. 6. Over all 1278 A-trees G, average size of the subtree $\mathcal{T}_{d^\circ}(G,S)$ and of the whole tree $\mathcal{T}(G,S)$ (y axis) for each space tree $\mathcal{T}(G,S)$ for which $D(\mathcal{T}(G,S))$ belongs to the same depth range (x axis).

even when the posterior probability of α_{min} is low, as shown by Figure 5(right).

To complement these observations, we show in Figure 6 the respective sizes, on average, of the whole reconciliation spaces and of the subspace that covers the probability mass. It shows that as few as a thousand reconciliations are necessary to cover the probability mass even with spaces of up to ten millions of reconciliations.

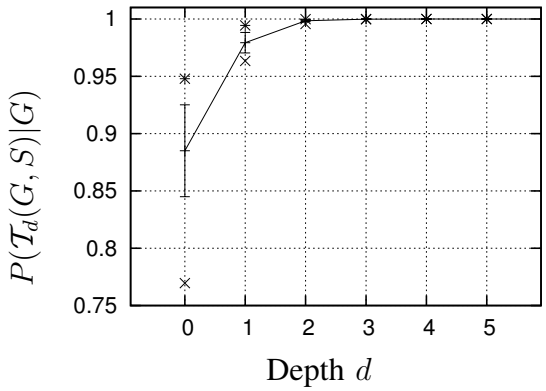

Fig. 7. Over all 24 A-trees such that $55 \leq D(\mathcal{T}(G,S)) \leq 59$, the average probability mass covered by $\mathcal{T}_d(G,S)$ (y axis) for the considered depth d (x axis).

Another argument for the concentration of the whole probability mass around the MPR is how fast it increases for a subtree $\mathcal{T}_d(G,S)$ according to a given depth d. We performed such analysis on the 24 A-trees G for which $D(\mathcal{T}(G,S))$ belongs to the highest depth range (that is between 55 and 59), which are the ones with the largest required depth and the smallest probability $P(\alpha_{min}|G)$ (see Figures 5(left) and 3), and as we can see in Figure 7 below, with a depth d as small as 2, the probability mass covered by $\mathcal{T}_d(G,S)$ is almost equal to one.

Although the immediate neighborhood $\mathcal{T}$ of α_{min} of depth $d^\circ(\mathcal{T}(G,S))$ technically covers the probability mass of the whole space of reconciliations between G and S, the probability of each reconciliation (that is reconciled tree) located beyond this minimal depth may have a non negligible contribution in the computation of the (exact) probability $P(G)$. This question is important due to the difference in terms of computational complexity between the exact posterior probability $P(\alpha|G)$ for each visited reconciliation α (Theorem 1, point 1) and its $\mathcal{T}$-approximation $P_{\mathcal{T}}(\alpha|G)$ (Theorem 1, point 2). To assess this point, we compared the exact probability $P(G)$ and its $\mathcal{T}$-approximation $P_{\mathcal{T}}(G)$, where the error ratio of the latter according to the former is $1 - P_{\mathcal{T}}(G)/P(G)$. The results are depicted in Figures 8, 9 and 10 below: (1) the error ratio is inversely proportional to the depth d of the considered subspace $\mathcal{T}$; (2) with a depth d as small as 1, the average ratio is 0.02 and the average number of visited reconciliations is 10, and (3) for each gene tree, the approximation $P_{\mathcal{T}}(G)$ computed with a subtree $\mathcal{T}$ of depth $d \geq 8$ is equal to $P(G)$.

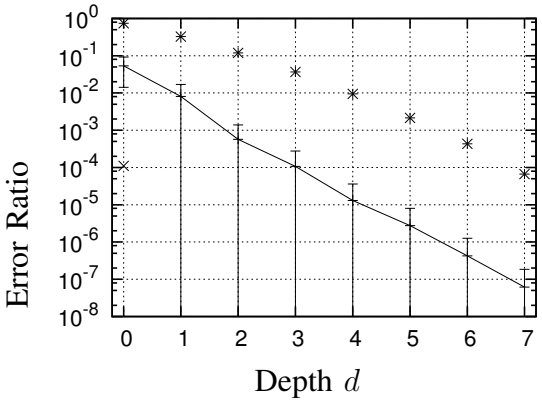

Fig. 8. Over all 1278 A-trees G, the error ratio of the approximated probability $P_{\mathcal{T}}(G)$ (y axis) for the subtree $\mathcal{T}$ (of $\mathcal{T}(G,S)$) of depth $d \in \{0,1,\ldots,7\}$ (x axis).

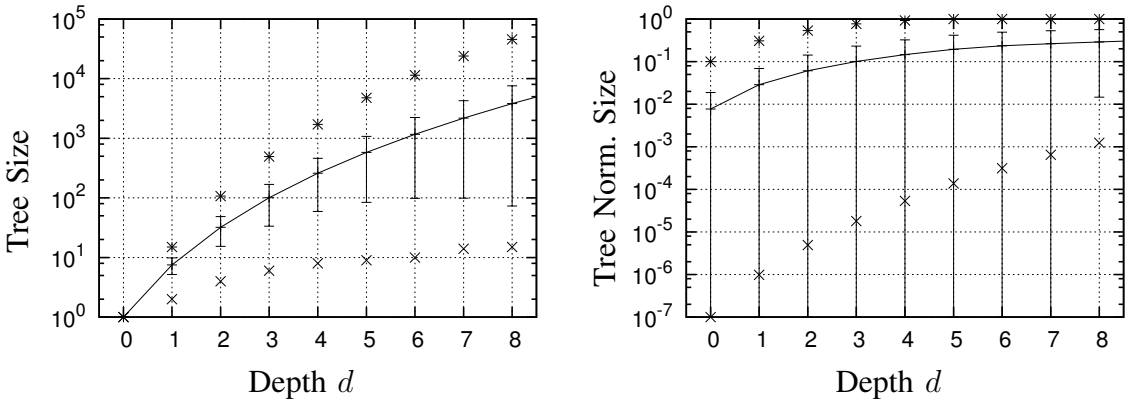

Fig. 9. Over all 1278 A-trees G and for each depth $d \in \{0,1,\ldots,8\}$ (x axis), the average size of the corresponding subtree of $\mathcal{T}(G,S)$ (y axis) both in absolute value (**left**) and normalized by the number of reconciliations (**right**).

B. Synthetic gene trees

With the duplication and loss rates used above for the 1278 real gene trees, the conclusion is that the immediate neighborhood of the MPR mostly covers the probability mass of the whole tree of reconciliations. The question that we address now is whether or not this is true for gene trees that would be generated with higher rates? As we can see in Table I below, increasing the duplication and loss rates suggests that the average probability of the MPR α_{min} decreases and the frequency where it is not the most likely one (denoted by α^*) increases. Figures 11

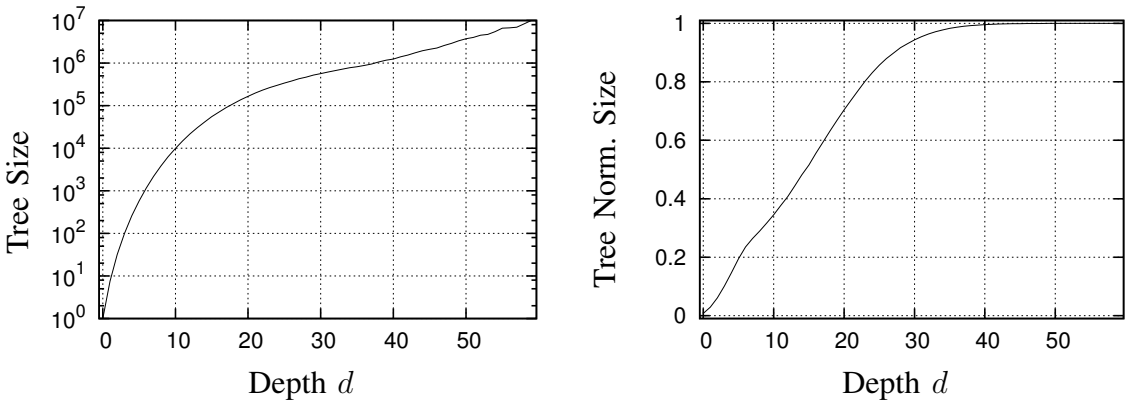

Fig. 10. Same than Figure 9, but for all possible depths.

I.F.	Nb. of gene trees	Average $P(\alpha_{min}\|G)$	% of the Nb. of G s.t. $\alpha_{min} \neq \alpha^*$
1	1051	0.97876	0.09% (1)
1.4	1025	0.95234	0.78% (8)
1.8	924	0.91781	1.51% (14)

TABLE I

FOR EACH I.F.: THE NUMBER OF GENE TREES (A-TREES) GENERATED, THE AVERAGE PROBABILITY $P(\alpha_{min}|G)$, AND THE NUMBER OF GENE TREES G SUCH THAT $\alpha_{min} \neq \alpha^*$. THE MAXIMAL NMC DISTANCE BETWEEN α_{min} AND α^* IS 2.

to 14 below show that with higher duplication and loss rates, the probability mass is more evenly dispersed among the reconciliations, but the highest concentration is still located among the most parsimonious evolutionary scenarios and a small subset of reconciliations need to be explored to cover the probability mass.

IV. DISCUSSION AND CONCLUSION

In this work, we presented an efficient algorithm to compute, either exactly or approximately, the posterior probabilities of a subspace of the space of all reconciliations between a given gene tree and a given species tree, provided the gene duplication and loss rates are known. Based on this algorithm, we were able to explore large reconciliations spaces both for real and simulated datasets and we showed that, for realistic species tree and duplication and loss rates,

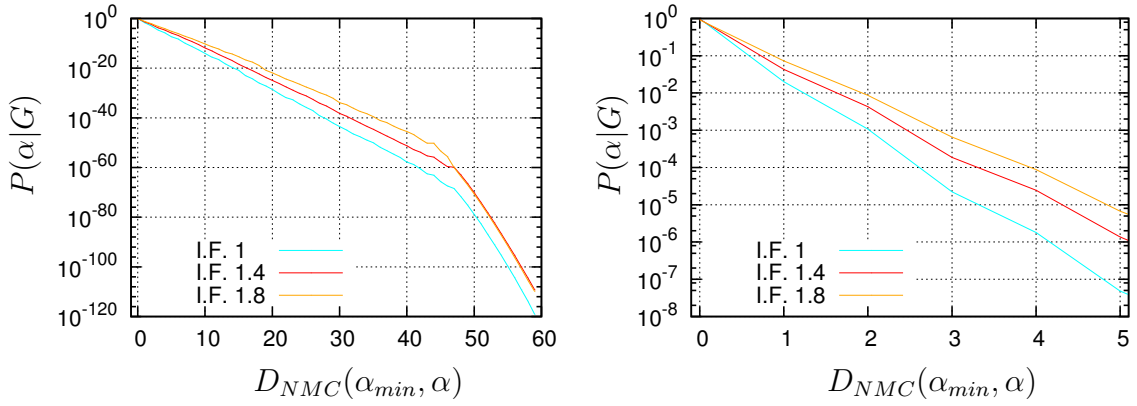

Fig. 11. For each I.F and the considered gene trees (A-trees), **(left)** average probability $P(\alpha|G)$ (y axis) of each reconciliation $\alpha \in \mathcal{T}(G,S)$ at the distance $D_{NMC}(\alpha_{min}, \alpha)$ to α_{min} (x axis), **(right)** zoom on depth less than or equal to 5.

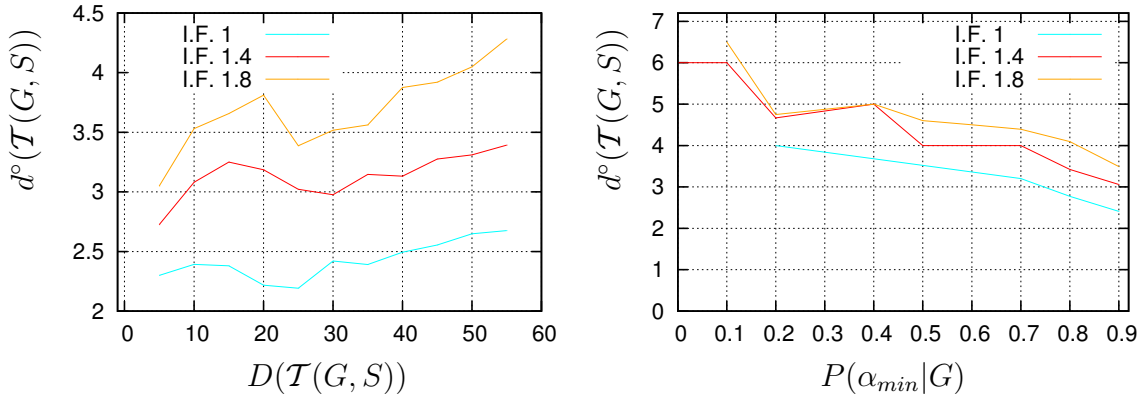

Fig. 12. For each I.F and the considered gene trees (A-trees), **(left)** average depth $d^{\circ}(\mathcal{T}(G,S))$ (y axis) for each space tree $\mathcal{T}(G,S)$ for which $D(\mathcal{T}(G,S))$ belongs to the same depth range (x axis), and **(right)** average depth $d^{\circ}(\mathcal{T}(G,S))$ (y axis) for each space tree $\mathcal{T}(G,S)$ for which $P(\alpha_{min}|G)$ belongs to the same probability range (x axis).

only a very small subset of reconciliations need to be explored to obtain in a very short time very precise approximations of the posterior probabilities of the most likely reconciliations. Such computational speed-up allows to analyze gene families with potentially a very large number of reconciliations, as we demonstrated in our experiments. It can also have applications in a Bayesian framework where duplications and losses rates are estimated using an MCMC approach [21], by reducing the time required in each state of the Markov Chain, for given duplication and loss rates. Our second contribution is experimental.

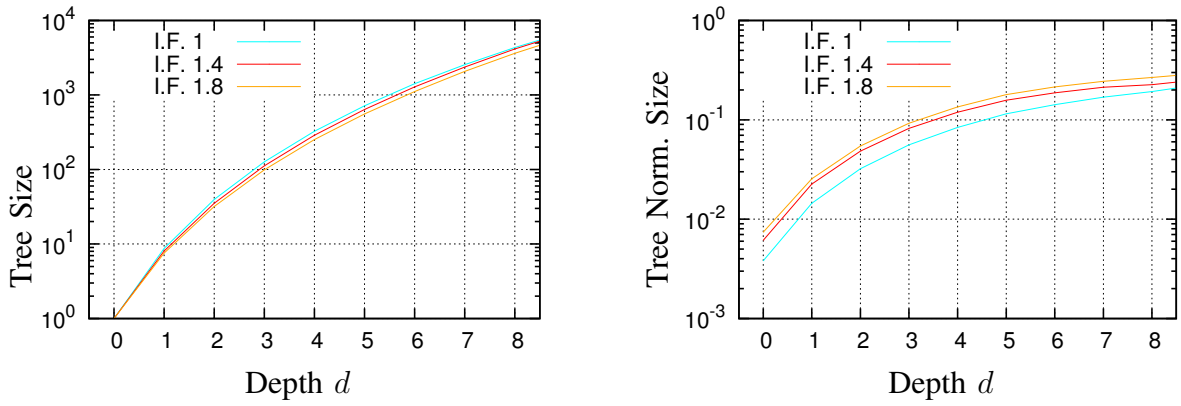

Fig. 13. For each I.F and the considered gene trees (A-trees), and for each depth $d \in \{0, 1, \ldots, 8\}$ (x axis), the average size of the corresponding subtree of $\mathcal{T}(G, S)$ (y axis) both in absolute value (**left**) and normalized by the number of reconciliations (**right**).

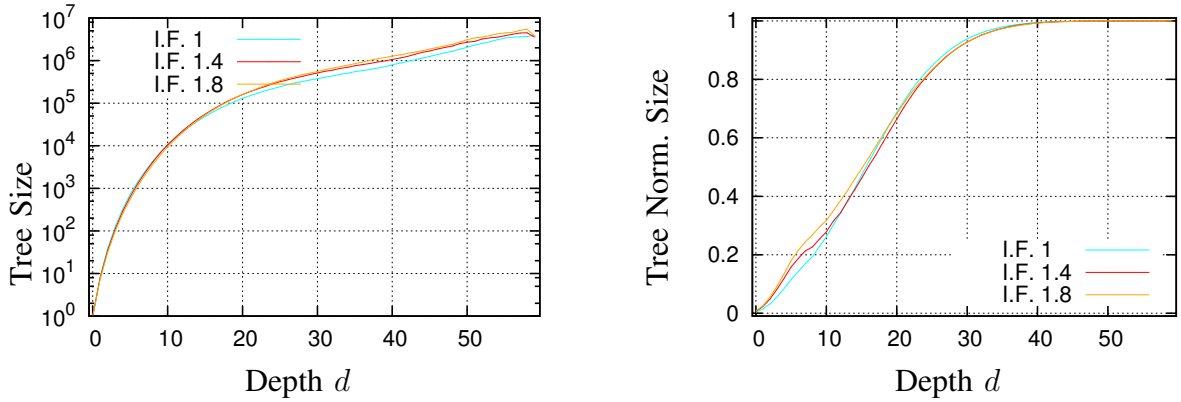

Fig. 14. Same than Figure 10, but for all possible depths.

With gene families from 12 fungal genomes and realistic duplication and loss rates along the corresponding species phylogeny, our analysis on the probabilistic landscape of the space of reconciliations show that the more probable is a reconciliation, the more it is located close (in term of NMC operators) to the MPR and the immediate neighborhood of the latter covers most of the whole space probability mass. For synthetic gene trees generated with higher rates, although its probability mass is more evenly dispersed, the same property holds. We believe these results offer the first detailed probabilistic analysis of the space of reconciliations and, together with the

recent works of Sennblad *et al.* [2], [21], [1], clearly indicate that the probabilistic analysis of gene family evolution is applicable to large datasets, even if more experiments have to be done with different data (higher rates along the considered phylogeny, different duplication and loss rates along a given branch, larger species and gene trees). Recent works on ancestral genome reconstruction for example [18] could benefit from such algorithms.

It would also be important to study different types of data (duplication and loss rates and species phylogeny) where the most probable reconciliation is located far from the MPR and the probability mass of the whole space is not concentrated around a single reconciliation. For such problematic data, it would also be of particular interest to take a look over the generated gene trees (gene family profiles and sizes) to point out gene tree characteristics for which our approach would not be relevant. As a major problem of a Markov Chain Monte Carlo approach is caused by the presence of several peaks in the probability distribution, which forces it to stay in a small region of the space for a long period of time, these informations would obviously be useful when such an approach is used to approximate the posterior probabilities mentioned above, with prior on the rates. With a similar Bayesian framework, our observations can also be useful to develop a MCMC approach to approximate the posterior probabilities of duplication and loss rates given one (or more) gene tree, and that will be an alternative to the Expectation-Maximization approach of [5].

One fundamental application of such probabilistic analysis could be to detect and correct wrong gene trees. One of the major problems in using gene families trees is related to uncertainties and errors in such trees. Hahn illustrated this problem, in a parsimony framework, in [14], while [1] illustrates with the same fungal gene families we considered in a probabilistic framework. One possible approach to detect possible erroneous gene trees could then be to search the neighborhood of a given gene tree G (in terms of operations such as Nearest Neighbor Interchange [3] or the Tree Pruning and Regrafting [4]) and to see if some of its neighbors has a higher probability. This would require to design efficient algorithms to update efficiently the probability of a gene tree after such an operation is performed.

APPENDIX I
DEFINITIONS

Let $\sigma : L(G) \rightarrow L(S)$ be the function that maps each leaf of G to the unique leaf of S with the same label. The *LCA-mapping* $M : V(G) \rightarrow V(S)$ maps each vertex u of G to the unique vertex $M(u)$ of S such that $\Lambda(S_{M(u)})$ is the smallest cluster of S containing $\Lambda(G_u)$.

Given two cells (either a vertex or an edge) c and c' of a tree T, $c' \leq_T c$ (resp. $c' <_T c$) if and only if c is on the unique path from c' to $r(T)$ (resp. and $c \neq c'$), in that case c' is said to be a (resp. strict) descendant of c.

Definition 1: A reconciliation between a gene tree G and a species tree S is a mapping $\alpha : V(G) \rightarrow V(S) \cup E(S)$ such that

1) *(Base constraint)* $\forall u \in L(G), \alpha(u) = M(u) = \sigma(u)$.
2) *(Tree Mapping Constraint) For any vertex $u \in V(G) \setminus L(G)$,*
   a) *if $\alpha(u) \in V(S)$, then $\alpha(u) = M(u)$.*
   b) *If $\alpha(u) \in E(S)$, then $M(u) <_S \alpha(u)$.*
3) *(Ancestor Consistency Constraint) For any two vertices $u, v \in V(G)$, such that $v <_G u$,*
   a) *if $\alpha(u), \alpha(v) \in E(S)$, then $\alpha(v) \leq_S \alpha(u)$,*
   b) *otherwise, $\alpha(v) <_S \alpha(u)$.*

APPENDIX II
INPUT DATA

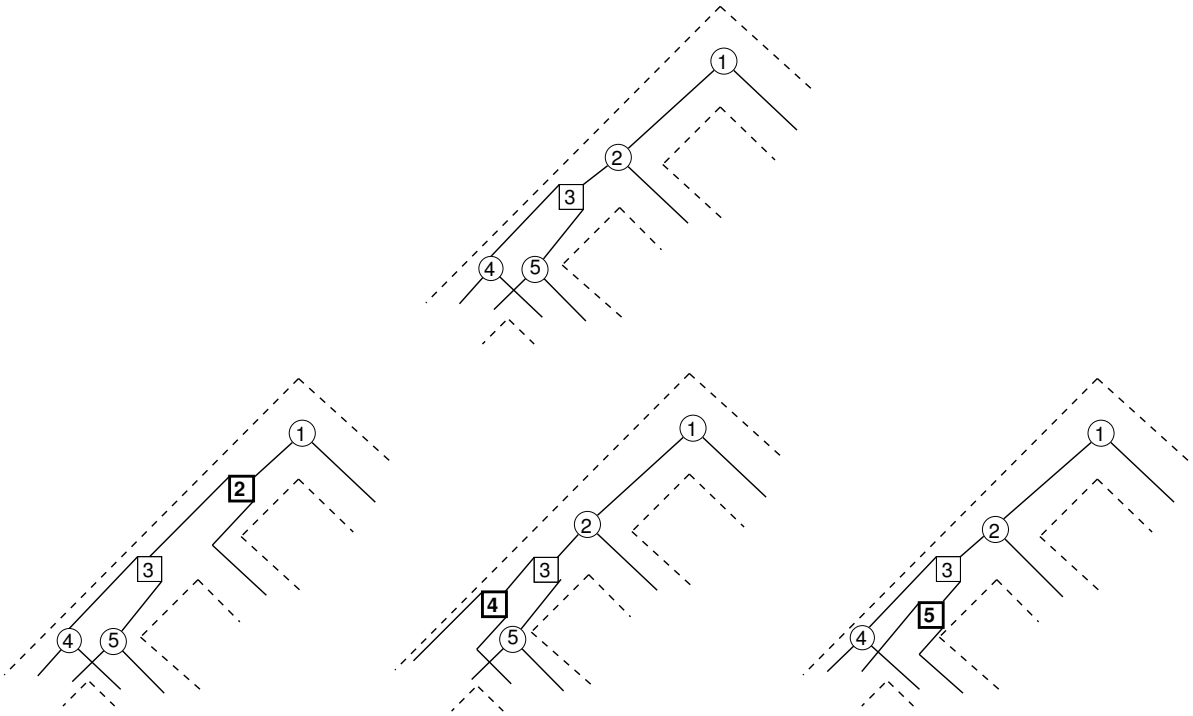

Fig. 15. The subtree of $\mathcal{T}(G, S)$ rooted at α_{min} for the trees G and S depicted in Figure 1. α_{min} and its children respectively are at the top and bottom of the figure. For each child, the vertex that has been moved upward is in boldface.

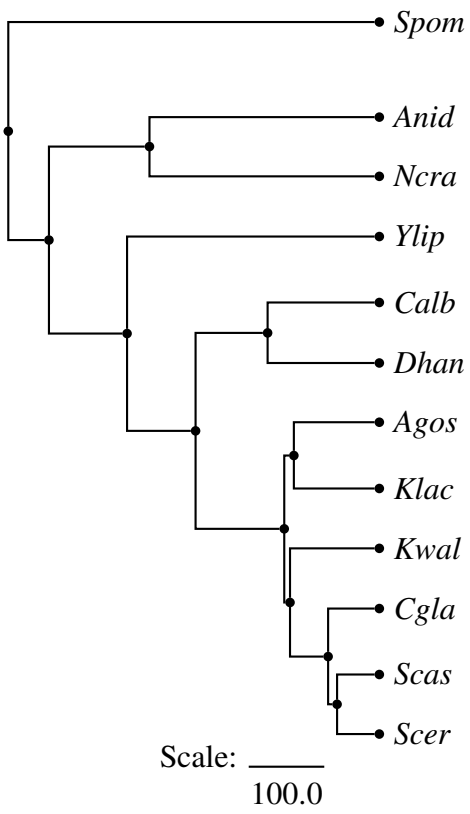

Fig. 16. The species tree S for the 12 fungal genomes, where divergence time is in Million Years.

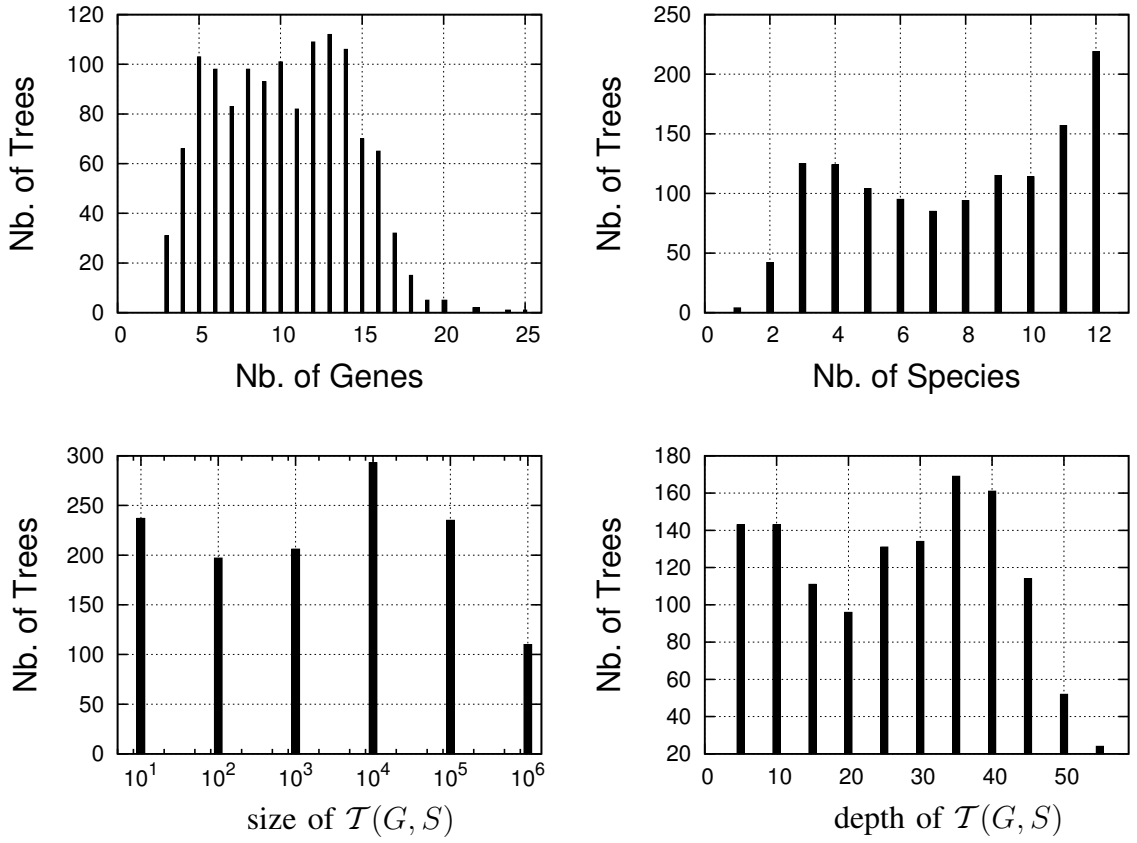

Fig. 17. Distribution of the 1278 real and A-trees G according to the number of genes and species present in G (**above**) and the size and depth of the space tree $\mathcal{T}(G,S)$ (**below**).

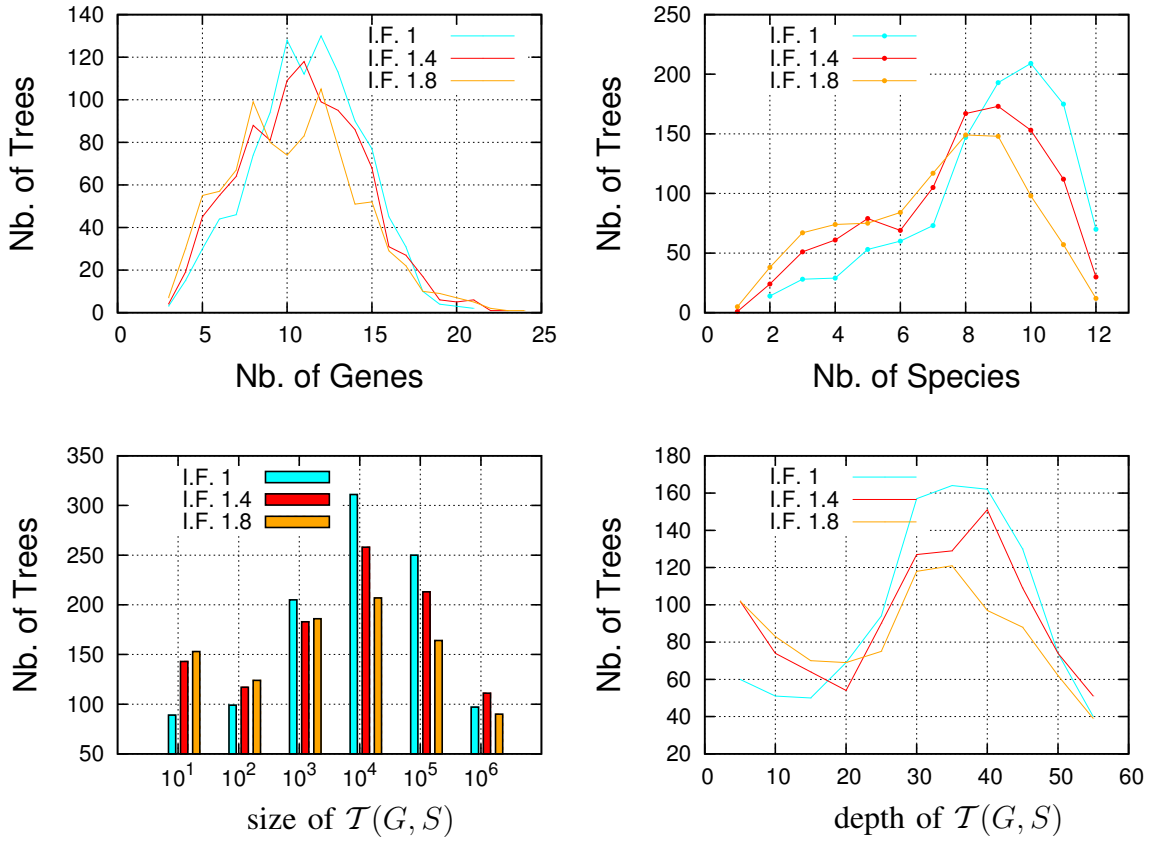

Fig. 18. For each I.F., distribution of the considered synthetic gene trees G according to the number of genes and species present in G (**above**) and the size and depth of the space tree $\mathcal{T}(G,S)$ (**below**).

ACKNOWLEDGEMENTS

C.C. and S.H. are supported, through the Discovery Grants program, by the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors would like to thank H. Philippe for providing the branch lengths of the fungi species tree.

REFERENCES

[1] Örjan Akerborg, Bengt Sennblad, Lars Arvestad, and Jens Lagergren. Simultaneous bayesian gene tree reconstruction and reconciliation analysis. *Proc. National Academy Sci. U.S.A.*, 106(14):5714–5719, 2009.

[2] Lars Arvestad, Jens Lagergren, and Bengt Sennblad. The gene evolution model and computing its associated probabilities. *J. ACM*, 56(2):1–44, 2009.

[3] Mukul S. Bansa, Oliver Eulenstein, and Andre Wehe. The gene-duplication problem: Near-linear time algorithms for nni based local searches. *IEEE/ACM Trans. Comput. Biol. and Bioinform.*, 6(2):221–231, 2009.

[4] Mukul S. Bansal and Oliver Eulenstein. An $\omega(n^2/\log n)$ speed-up of tbr heuristics for the gene-duplication problem. *IEEE/ACM Trans. Comput. Biol. and Bioinform.*, 5(4):514–524, 2008.

[5] Tijl De Bie, Nello Cristianini, Jeffrey P. Demuth, and Matthew W. Hahn. Cafe: a computational tool for the study of gene family evolution. *Bioinformatics*, 22(10):1269–1271, 2006.

[6] Paola Bonizzoni, Gianluca Della Vedova, and Riccardo Dondi. Reconciling a gene tree to a species tree under the duplication cost model. *Theoret. Comput. Sci.*, 347(1-2):36–53, 2005.

[7] K.P. Byrne and K.H. Wolfe. Consistent patterns of rate asymmetry and gene loss indicate widespread neofunctionalization of yeast genes after whole-genome duplication. *Genetics*, 175(3):1341–1350, 2007.

[8] J.P. Doyon, C. Chauve, and S. Hamel. Space of gene/species trees reconciliations and parsimonious models. *J Comput Biol*, 16(10):1399–1418, 2009.

[9] Walter M. Fitch. Homology - a personal view on some of the problems. *Trends Genet.*, 16(5):227 – 231, 2000.

[10] M. Goodman, J. Czelusniak, G.W. Moore, R.A. Herrera, and G. Matsuda. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Syst. Zool.*, 28:132–163, 1979.

[11] PawełGórecki and Jerzy Tiuryn. Dls-trees: a model of evolutionary scenarios. *Theoret. Comput. Sci.*, 359(1):378–399, 2006.

[12] A. De Grassi, C. Lanave, and C. Saccone. Genome duplication and gene-family evolution: the case of three oxphos gene families. *Gene*, 421(1-2):1–6, 2008.

[13] Dan Graur and Wen-Hsiung Li. *Fundamentals of Molecular Evolution second edition*. Sinauer Associates, Sunderland, MA., 1999.

[14] Matthew W. hahn. Bias in phylogenetic tree reconciliation methods: implications for vertebrate genome evolution. *Genome Biol.*, 8:R141, 2007.

[15] Harvey. *C++ How to Program (5th Edition) (How to Program)*. Prentice Hall, 2005.

[16] Olivier Jeffroy, Henning Brinkmann, Frédéric Delsuc, and Hervé Philippe. Phylogenomics: the beginning of incongruence? *Trends Genet*, 22(4):225–31, Apr 2006.

[17] David G. Kendall. On the generalized "birth-and-death" process. *Ann. Math. Statistics*, 19:1–15, 1948.

[18] Jian Ma, Aakrosh Ratan, Brian J. Raney, Bernard B. Suh, Louxin Zhang, Webb Miller, and David Haussler. Dupcar: Reconstructing contiguous ancestral regions with duplications. *J. Comput. Biol.*, 15(8):1007–1027, 2008.

[19] Roderic D. Page. Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. *Syst. Biol.*, 43:58–77, 1994.

[20] Michael Sanderson and Michelle McmMahon. Inferring angiosperm phylogeny from est data with widespread gene duplication. *BMC Evolutionary Biology*, 7(Suppl 1), 2007.

[21] Bengt Sennblad and Jens Lagergren. Probabilistic orthology analysis. *Syst. Biol.*, 58(4):411–424, 2009.

[22] J. Thorne, H. Kishino, and I. Painter. Estimating the rate of evolution of the rate of molecular evolution. *Mol. Biol. Evol.*, 15:1647–1657, 1998.

[23] M.J. van Hoeck and P. Hogeweg. Metabolic adaptation after whole genome duplication. *Mol. Biol. Evol.*, 2009. To appear. DOI:10.1093/molbev/msp160.

[24] Ilan Wapinski, Avi Pfeffer, Nir Friedman, and Aviv Regev. Natural history and evolutionary principles of gene duplication in fungi. *Nature*, 449:54–61, 2007.

[25] K.H. Wolfe and D.C. Shields. Molecular evidence for an ancient duplication of the entire yeast genome. *Nature*, 387(6634):708–713, 1997.

CHAPITRE 8

PRÉSENTATION DU TROISIÈME ARTICLE

8.1 Détails de l'article

Branch-and-Bound approach for parsimonious inference of a species tree from a set of gene family trees

Jean-Philippe Doyon et Cedric Chauve

Accepté dans *Software Tools and Algorithms for Biological Systems*

Research Book to be published by Springer

8.2 Partage du travail

M. Chauve et M. Doyon ont défini le plan et rédigé l'article. M. Doyon a implémenté la méthode, s'est occupé des expériences et a défini le plan des expériences avec M. Chauve. M. Doyon a développé les algorithmes, les preuves et la majorité des composantes théoriques. M. Chauve a aidé M. Doyon à rédiger certaines preuves.

Branch-and-Bound approach for parsimonious inference of a species tree from a set of gene family trees

Jean-Philippe Doyon[1,3] and Cedric Chauve[2⋆]

[1] LIRMM, Université Montpellier2 and CNRS, UMR 5506 - CC 477, 161 rue Ada 34392 Montpellier Cedex 5, France.
[2] Department of Mathematics, Simon Fraser University, 8888 University Drive, V5A 1S6, Burnaby (BC), Canada,
[3] DIRO, Université de Montréal, CP6128, succ. Centre-Ville, H3C 3J7, Montréal (QC),Canada,

Abstract. We describe a Branch-and-Bound algorithm for computing a parsimonious species tree given a set of gene family trees. Our algorithm can compute a parsimonious species tree for three cost measures: number of gene duplications, number of gene losses, and both combined. Moreover, to cope with intrinsic limitations of Branch-and-Bound algorithms for species trees inference regarding the number of taxa that can be considered, our algorithm can naturally take into account predefined relationships between sets of taxa. We test our algorithm on a dataset of eukaryotic gene families spanning 29 taxa.

Keywords. Comparative genomics, Evolution and phylogenetics.

1 Introduction

Speciation is the fundamental mechanism of genome evolution, especially for eukaryotic genomes. However, other events can happen, that do not result immediately in the creation of new species but act as fundamental evolutionary mechanisms, such as gene duplication and loss [9][4]. Duplication is the genomic process where one or more genes of a single genome are copied, resulting in two copies of each duplicated gene. Gene duplication allows one copy to possibly develop a new biological function through point mutation, while the other copy often preserves its original role. A gene is considered to be lost when the corresponding sequence has been deleted by a genomic rearrangement or has completely lost any functional role (i.e. has become a pseudogene). (See [9] for example). Genes of contemporary species that evolved from a common ancestor, through speciations and duplications, are said to be homologs [7] and are grouped into a gene family. Such gene families are in general inferred using protein sequence comparison.

The availability of large datasets of gene families makes now possible to perform genome-scale phylogenetic analyses. A widely used approach, named Gene Tree Parsimony (GTP for short), is based on the notion of *reconciliation* between a gene tree and a species tree introduced in [8], and seeks a species tree with a minimum overall reconciliation cost with the whole set of input gene trees. Given a gene tree G and a species tree S for the corresponding taxa, the reconciliation cost is the minimum number of duplications, losses, or mutations (duplications plus losses) that is needed to explain the (possible) discrepancies between G and S. Computing a most parsimonious reconciliation between a given gene tree and a species tree can be done in linear time [22], but inferring a parsimonious species tree is an NP-complete problem for both duplication and mutation criteria [12], although fixed-parameter tractable algorithms have been described in [13, 20]. Hence, in most cases, studies based on GTP use either a brute-force approach when the number of taxa is low, as in [18], greedy heuristics [4, 5] or the local search approach with edit operations on species

⋆ Contact author

[4] Other genomic events such as lateral gene transfer, that occurs mostly in bacterial genomes, will not be considered here.

trees such as the Subtree Pruning and Regrafting (rSPR) [14, 1] or Nearest-Neighbour Interchange (NNI) [2]. Although such heuristics, especially local-search ones, are fast and proved to be effective on large datasets [21], they do not guarantee to infer an optimal species tree.

A Branch-and-Bound approach is a classical method when dealing with hard species tree inference problems, as it implicitly explores the space of species trees and guarantees to find an optimal phylogeny for a given criterion. The optimality of such an approach requires that the objective function is nondecreasing during a downward phase of the exploration space and its effectiveness relies significantly on the required time to compute the cost measure associated to a newly visited species tree given the previous one. This method has been used [10] for evolutionary criteria such as Maximum Parsimony and Maximum Likelihood (the reader is referred to [6] for a brief overview), but it has not been considered up to now for the GTP. In the present work, we present a Branch-and-Bound algorithm that guarantees to find a species tree S with the minimum cost for a given gene tree G, and works for the three usual criteria (duplications, losses, and mutations). Our algorithm relies on a new way to explore the space of species trees that allows to update efficiently the cost (for the three considered costs) of a partial species tree and to naturally account for prior knowledge of the species phylogeny, and then process datasets that span a significant number of taxa.

The plan of the paper is as follows. In Section 2, formal notations are defined. In Section 3, we describe our Branch-and-Bound algorithm. In Section 4, we apply our algorithm on a dataset of 1111 gene families from 29 animal genomes taken from the TreeFam database [11, 17].

2 Preliminaries

Gene trees, species trees, forests. Except when indicated, any considered tree is rooted, binary, unordered and leaf-labeled. For simplicity, we consider that each leaf label is an integer. For a given tree T, let $V(T)$, $r(T)$, $L(T)$, $\Lambda(T)$ and $I(T)$ respectively denote its vertex set, its root, its leaf set, its label set (the set of integers that appear at its leaves), and its internal vertex set (that is $V(T) \setminus L(T)$). The depth of a vertex is the length of the unique path to $r(T)$ and the height of T, denoted $h(T)$, is its maximal depth. We will adopt the convention that the root is at the top of the tree and the leaves at the bottom.

Given two vertices u and v of T, $u \leq_T v$ (resp. $u <_T v$) if and only if v is on the unique path from u to $r(T)$ (resp. and $u \neq v$); in such a case, u is said to be a *(resp. strict) descendant* of v. For a vertex u of T, we denote by u_1 and u_2 its children (when $u \notin L(T)$), by $p(u)$ its parent, by $s(u)$ its sibling (when $u \neq r(T)$), and by T_u the subtree of T rooted at u. It is important to point out that because T is an unordered tree, the children u_1 and u_2 of an internal vertex u of T are interchangeable, that is u_1 may arbitrarily be selected as the unique children of u that respects a given constraint. The distance between two vertices u and v of a tree T, where $u <_T v$, is denoted $d_T(u, v)$ and is the number of vertices on the path from u to v in T, excluding u and v.

A *forest* $\mathcal{T}$ is a set of trees. The notations $V(\mathcal{T})$, $r(\mathcal{T})$, $L(\mathcal{T})$, $\Lambda(\mathcal{T})$ and $I(\mathcal{T})$ have the same definitions as for single trees, except that $r(\mathcal{T})$ is the set of roots of all the trees in $\mathcal{T}$. A forest is said to be ordered if there is a total order on the trees it contains. Given two trees S_1 and S_2 of a forest, we denote by $(S_1 + S_2)$ the trees obtained by joining S_1 and S_2 under a common (binary) root x (i.e. creating two edges from x to the roots of S_1 and S_2).

A *species tree* S is a tree such that each element of $\Lambda(S)$ represents an extant species and labels exactly one leaf of S (there is a bijection between $L(S)$ and $\Lambda(S)$). A *species forest* $\mathcal{F}$ is simply a set of trees with disjoint label sets. A *gene tree* G is a tree such that $\Lambda(G) \subseteq \Lambda(S)$ (each leaf of G represents an extant gene that belongs to a species of $\Lambda(S)$).

Reconciliation between a gene tree and a species tree. A reconciliation between a gene tree G and a species tree S maps each internal vertex of G onto a vertex of S and induces an evolutionary history in term of gene duplications and losses. The Lowest Common Ancestor mapping (LCA-mapping), that maps a gene u of G onto the most recent species of S that is ancestor of all genomes that contain a gene descendant of u, is the most widely used mapping. It depicts a parsimonious evolutionary process for each of the three usual combinatorial criteria, which is also the unique parsimonious scenario for the number of losses and the number of mutations [5], while there can be several parsimonious reconciliations for the number of duplications.

Definitions 2 to 4 below define how to read the different costs associated to the reconciliation between a given gene tree G and a given species tree S.

Definition 1. The LCA-mapping between a gene tree G and a species tree S, denoted $M_S : V(G) \to V(S)$, is defined as follows: given a vertex u of G, $M_S(u)$ is the unique vertex x of S such that $\Lambda(G_u) \subseteq \Lambda(S_x)$ and either x is a leaf of S, or $\Lambda(G_u) \not\subseteq \Lambda(S_{x_1})$ and $\Lambda(G_u) \not\subseteq \Lambda(S_{x_2})$. $\diamond$

Definition 2. An internal vertex $u \in I(G)$ is a duplication if $M_S(u) = M_S(u_1)$ and/or $M_S(u) = M_S(u_2)$. The duplication cost of the reconciliation between G and S is $d(G,S) = \sum_{u \in I(G)} d(u,S)$, where $d(u,S)$ has value 1 if and only if $u \in I(G)$ is a duplication and 0 otherwise. $\diamond$

Definition 3. The loss cost of the reconciliation between G and S is $l(G,S) = \sum_{u \in I(G)} l(u,S)$, where $l(u,S)$ is defined as follows

$$l(u,S) = \begin{cases} 0 & (1) \text{ if } M_S(u) = M_S(u_1) = M_S(u_2); \\ d_S(M_S(u_1), M_S(u)) + 1 & (2) \text{ if } M_S(u_1) \neq M_S(u) \text{ and } M_S(u_2) = M_S(u); \\ d_S(M_S(u_1), M_S(u)) + d_S(M_S(u_2), M_S(u)) & (3) \text{ if } M_S(u_1) \neq M_S(u) \text{ and } M_S(u_2) \neq M_S(u). \end{cases}$$

$\diamond$

Definition 4. The mutation cost of the reconciliation between G and S is $m(G,S) = l(G,S) + d(G,S)$. $\diamond$

Note that some internal vertices of G correspond to duplications for any given species tree S. Such vertices, that are called *apparent duplications* (also sometimes *forced duplications*) are defined as the vertices u such that $\Lambda(G_{u_1}) \cap \Lambda(G_{u_2}) \neq \emptyset$. Internal vertices u such that $|\Lambda(G_u)| = 1$ are obviously apparent duplications, and then, from now, without loss of generality, we consider that such vertices are replaced by a single leaf, which implies that there is no extant gene u (leaf) of G that has the same label as its sibling $s(u)$.

The LCA-mapping between a gene tree G and a species forest $\mathcal{F}$, denoted $M_{\mathcal{F}} : V(G) \to V(\mathcal{F})$, is defined similarly to the case of a single species tree (see Definition 1). For a given vertex $u \in V(G)$, $M_{\mathcal{F}}(u) = M_S(u)$, if there is a species tree S of $\mathcal{F}$ such that $M_S(u)$ is defined. Otherwise, $M_{\mathcal{F}}(u)$ is said to be undefined (which is denoted $M_{\mathcal{F}}(u) = \varnothing$ from now).

The goal of the current work is the design of an exact method to solve the following optimization problem, given a cost measure c (either d, l, or m) for the reconciliation between a gene tree and a species tree.

Minimum C Species Tree Problem
Input. A gene tree forest $\mathcal{G} = \{G_1, \ldots, G_k\}$
Output. A species tree S such that $\sum_{i=1}^{k} c(G_i, S)$ is minimized.

3 A Branch-and-Bound algorithm for the GTP

We now describe our Branch-and-Bound algorithm. We assume that there are n taxa, denoted by $\{1, 2, \ldots, n\}$, and denote by $\mathcal{K}^n$ the set of all possible species trees on these n taxa. Without loss of generality, we describe our algorithm for a single gene tree G.

The algorithm is based on the exploration of a rooted tree denoted $\mathcal{T}^n$, where each vertex corresponds to a forest of species trees, and such that each internal forest corresponds to an incomplete species tree for n species, and each leaf forest to a complete species tree S of $\mathcal{K}^n$. The Branch-and-Bound explores this tree and each time it visits a forest denoted $\mathcal{F}$, it computes a lower bound on the cost $c(G, S)$ (where $c = l$ or $c = d$) of any species tree $S \in \mathcal{K}^n$ located in $\mathcal{T}^n_{\mathcal{F}}$, that is the subtree of $\mathcal{T}^n$ rooted at $\mathcal{F}$. To ensure the optimality of this approach, such a lower bound has to respect the following definition.

Definition 5. Let $\pi : \mathcal{K}^n \to \mathbb{N}$ be an objective function that we seek to minimize. A function $\omega : V(\mathcal{T}^n) \to \mathbb{N}$ is a Consistent Lower Bound (CLB) for π if and only if (1) it is non-decreasing along the path that starts at $r(\mathcal{T}^n)$ and ends at any leaf $\{S\}$ of $\mathcal{T}^n$ and (2) $\omega(\{S\}) = \pi(S)$. $\diamond$

Given a CLB, denoted $c(G, \mathcal{F})$, for the considered cost $c(G, S)$, then $\mathcal{T}^n_{\mathcal{F}}$ is explored if and only if $c(G, \mathcal{F}) < c(G, S_{min})$, where $S_{min} \in \mathcal{K}^n$ is the best solution found since the beginning of the exploration of $\mathcal{T}^n$ and is updated when a species tree with a lower cost is found. Such a Branch-and-Bound guarantees to find an optimal species tree S_{min} such that $c(G, S_{min})$ is minimum.

The plan of this section is as follows: first, we formally define the space tree $\mathcal{T}^n$ and give important combinatorial properties that are central in the design of the Branch-and-Bound; second, we describe an algorithm that updates the LCA-mapping $M_{\mathcal{F}} : V(G) \to V(\mathcal{F})$ for the current visited forest $\mathcal{F}$ given the mapping of the previously visited forest; third, we define the two CLBs for the costs $d(G, S)$ and $l(G, S)$; and finally, we explain how $\mathcal{T}^n$ can easily be adapted to account for prior knowledge on the species phylogeny.

Combinatorial structure of $\mathcal{T}^n$ (i.e. the species trees space exploration tree). The main structural feature of $\mathcal{T}^n$ is that a child $\mathcal{F}'$ of an internal forest $\mathcal{F}$ is defined by joining two of its trees under a (new) vertex (thus forming a new clade). This architecture is different from the classical one used in a Branch-and-Bound approach for phylogenetic inference, where the exploration starts with a tree with two leaves and one internal node, an then iteratively add a leaf and an edge until a complete tree is obtained. The advantage of the architecture of $\mathcal{T}^n$ over the classical one is that it is more adapted to efficiently compute the LCA-mapping during its traversal, which is essential to rapidly explore the space and solve our problem. Definition 7 formally describes the architecture of $\mathcal{T}^n$, illustrated by Figure 1, and Property 1 shows that it is an appropriate structure for the exploration of $\mathcal{K}^n$. Below, given a species tree S over $\{1, \ldots, n\}$, $min(\Lambda(S))$ denotes the minimum label of the leaves of S. We also define an order on the trees of a forest as follows.

Definition 6. Given two trees S and S' of a forest $\mathcal{F}$, $S \prec_{\mathcal{F}} S'$ (resp. $\preceq_{\mathcal{F}}$) if and only if $min(\Lambda(S)) < min(\Lambda(S'))$ (resp. $\leq$). $\diamond$

Definition 7. $\mathcal{T}^n$ is an ordered and rooted tree where each vertex is an ordered species forest $\mathcal{F}$ on $\{1, \ldots, n\}$, with a distinguished tree called the *branching tree* $\beta(\mathcal{F})$. The branching structure of $\mathcal{T}^n$ is defined below.

1. The root forest $r(\mathcal{T}^n)$ is the forest composed of n trees $\{S_1, \ldots, S_n\}$, where S_i is the tree reduced to a single vertex labeled i, and its branching tree is the tree S_1.

2. Each leaf of $\mathcal{T}^n$ is a forest $\mathcal{F}$ containing a single tree that is a species tree from $\mathcal{K}^n$.
3. A forest $\mathcal{F}_x$ is a child of an internal forest $\mathcal{F}$ if and only if there exists two trees S_{x_1} and S_{x_2} in $\mathcal{F}$, with $S_{x_1} \prec_{\mathcal{F}} S_{x_2}$, such that
   (a) $\mathcal{F}_x = \mathcal{F} - \{S_{x_1}, S_{x_2}\} \cup \{S_x\}$, where $S_x = (S_{x_1} + S_{x_2})$ is the branching tree of $\mathcal{F}_x$,
   (b) and either $S_{x_2} = \beta(\mathcal{F})$ or $\beta(\mathcal{F}) \preceq_{\mathcal{F}} S_{x_1}$.

Finally, the children of an internal vertex (i.e. forest) $\mathcal{F}$ of $\mathcal{T}^n$ are totally ordered as follows: if $\mathcal{F}_x$ and $\mathcal{F}_y$ are two children of $\mathcal{F}$, where the corresponding branching trees are respectively $S_x = (S_{x_1} + S_{x_2})$ and $S_y = (S_{y_1} + S_{y_2})$, then, $\mathcal{F}_x$ precedes $\mathcal{F}_y$ if and only if either i) $S_{x_1} \prec_{\mathcal{F}} S_{y_1}$ or ii) $S_{x_1} = S_{y_1}$ and $S_{x_2} \prec_{\mathcal{F}} S_{y_2}$. ◇

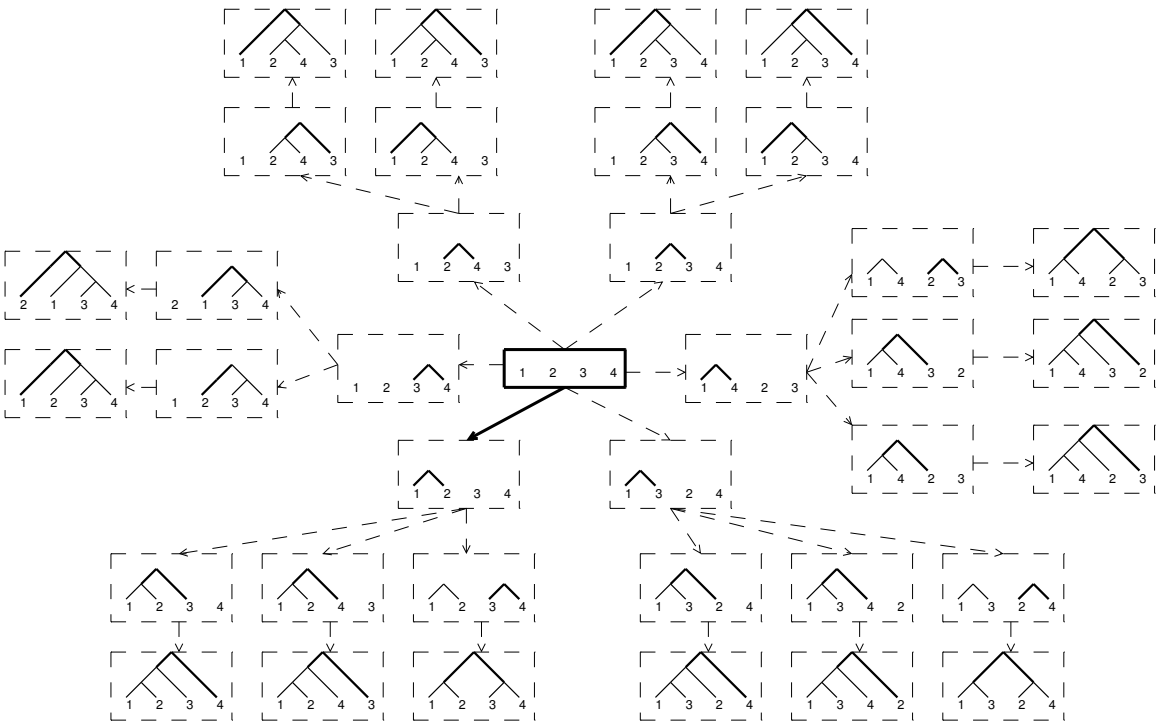

Fig. 1. For $n = 4$, representation of the space tree $\mathcal{T}^n$, where each square represents a forest, an arrow indicates one of its child, the root $r(\mathcal{T}^n)$ is the darkest square and its first child (according to the total order) is represented by the darkest arrow. The children of a forest are ordered according to the anti-clockwise direction. The darkest edges of a forest $\mathcal{F}_x$ indicate the two subtrees S_{x_1} and S_{x_2} of $\mathcal{F}$ that are joined together to form the new tree S_x of $\mathcal{F}_x$.

Property 1 below follows immediately from the structure of $\mathcal{T}^n$ described in Definition 7, and in particular from the order on the trees in a forest, that ensures that no two different paths from the root can lead to the same species forest.

Property 1. The tree $\mathcal{T}^n$ is such that (1) there are no two nodes that represent the same species forest; (2) $L(\mathcal{T}^n) = \mathcal{K}^n$; (3) its height is $\Theta(n)$; and (4) the number of children of each internal vertex is bounded by $O(n^2)$.

The general principle of our Branch-and-Bound algorithm is to visit $\mathcal{T}^n$ starting at its root, and then to recursively visit the children of the starting vertex forest $\mathcal{F}$ according to the order described

in Definition 7. We now explain how a subtree of $\mathcal{T}^n$ can be explored in time linear in the number of species forests it contains. There are two key points:

- To visit the children of an internal vertex $\mathcal{F}$ according to the order defined in Definition 7, it is sufficient that the trees of $\mathcal{F}$ are ordered according to $\prec_{\mathcal{F}}$.
- To maintain the order $\prec_{\mathcal{F}}$ in constant time when visiting a child forest $\mathcal{F}_x$ (of $\mathcal{F}$) with $S_x = (S_{x_1} + S_{x_2})$ as its branching tree, S_{x_2} is removed from the ordered forest and S_x replaces S_{x_1} (as $min(\Lambda(S_x)) = min(\Lambda(S_{x_1}))$). And then, when the traversal goes back to $\mathcal{F}$ after visiting the subtree $\mathcal{T}^n_{\mathcal{F}_x}$, the ordered forest $\mathcal{F}$ can be retrieved (in constant time) assuming that both the position of S_{x_2} in $\mathcal{F}$ and $S_x = (S_{x_1} + S_{x_2})$ (the used branching tree) were previously saved, which can be implemented easily using lists and pointers.

Together that the height of $\mathcal{T}^n$ is in $\Theta(n)$, this proves the following result.

Proposition 1. *The complete exploration of a subtree $\mathcal{T}$ of $\mathcal{T}^n$ can be implemented to run in time $\Theta(|V(\mathcal{T})|)$ and space $\Theta(n)$.*

We finally introduce some combinatorial properties on the architecture of $\mathcal{T}^n$ that will be used to define a CLB for the cost $l(G, S)$. According to Definition 3, the cost $l(u, S)$ induced by an internal vertex u of G depends on the distance in S between $M_S(u)$ and $M_S(u_1)$ (resp. $M_S(u_2)$). Hence, the main idea behind a CLB for $l(G, S)$ resides on the definition of a CLB for the distance $d_S(M_S(u_1), M_S(u))$ (resp. $d_S(M_S(u_2), M_S(u))$). Formally, considering a forest $\mathcal{F}$ of $\mathcal{T}^n$, a non-root vertex u of G, and any species tree $S \in \mathcal{K}^n$ that is located at a leaf of $\mathcal{T}^n_{\mathcal{F}}$, the question is as follows: how can a CLB for $d_S(M_S(u), M_S(p(u)))$ be efficiently computed during the traversal of $\mathcal{T}^n$ along the path that connects $r(\mathcal{T}^n)$ and $\mathcal{F}$? To define such a CLB, we introduce *incremental forests*.

Definition 8. Let $\mathcal{F}$ be a forest of $\mathcal{T}^n$ and $\mathcal{F}_x$ one of its children, whose branching tree is $S_x = (S_{x_1} + S_{x_2})$. Given a non-root vertex u of G, if the mapping $M_{\mathcal{F}}(u)$ is defined in either S_{x_1} or S_{x_2} and $M_{\mathcal{F}_x}(p(u))$ is not defined, then $\mathcal{F}_x$ is said to be an incremental forest for u. ◇

It is easy to see that each incremental forest for u located between $r(\mathcal{T}^n)$ and $\mathcal{F}$ (including $\mathcal{F}$) corresponds to an increment of one on $d_S(M_S(u), M_S(p(u)))$. If $d_{\mathcal{F}}(u)$ denotes the number of such incremental forests, then the property below immediately follows from the usual LCA-mapping (between G and S) and Definition 7.

Property 2. Given a leaf forest $\{S\}$ of $\mathcal{T}^n$ and a vertex u of $V(G) \setminus \{r(G)\}$, $d_S(M_S(u), M_S(p(u))) = d_{\{S\}}(u)$, and then $d_{\mathcal{F}}(u)$ is a CLB for $d_S(M_S(u), M_S(p(u)))$.

Updating the LCA-mapping. Let $\mathcal{F}$ be an internal forest of $\mathcal{T}^n$, and $\mathcal{F}_x$ be one of its child forest and the next one to be visited during the traversal of the space tree. The main issue here is to detect as efficiently as possible the vertices u of G for which the LCA-mapping was undefined in $\mathcal{F}$ but is now defined in $\mathcal{F}_x$. Definition 9 below describes the smallest forest of subtrees of G that contains all these vertices, Figure 2 depicts a simple example of this architecture, and then we explain how it can be explored in linear time.

Definition 9. Let $\mathcal{F}$ be an internal forest of $\mathcal{T}^n$ and $\mathcal{F}_x$ be one of its child, where $S_x = (S_{x_1} + S_{x_2})$ is the branching tree (see Definition 7). $\mathcal{G}_x$ denotes the forest of subtrees of G such that its root set is $r(\mathcal{G}_x) = \mathcal{M}_{\mathcal{S}_x} = \{u \in V(G) \setminus \{r(G)\} : M_{S_x}(u) \neq \varnothing \text{ and } M_{S_x}(s(u)) = \varnothing\}$ and its leaf set is $L(\mathcal{G}_x) = \mathcal{M}_{\mathcal{S}_{x_1}} \cup \mathcal{M}_{\mathcal{S}_{x_2}}$. ◇

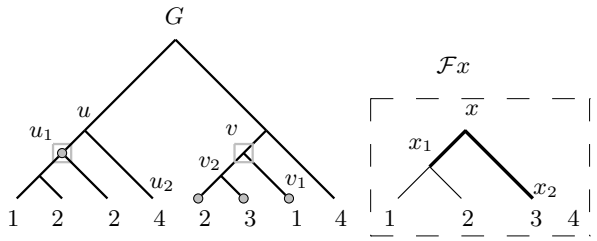

Fig. 2. Representation of the forest $\mathcal{G}_x$, given a gene tree G and a forest $\mathcal{F}x$ of $\mathcal{T}^n$, where $n = 4$. The vertices of G that are in $r(\mathcal{G}_x)$ (resp. $L(\mathcal{G}_x)$) are represented by a grey square (resp. circle). The two trees S_{x_1} and S_{x_2} of $\mathcal{F}$ used to define the branching tree S_x of $\mathcal{F}_x$ are indicated by larger edges. Because u_1 is mapped in S_x and u_2 is not, u_1 is both a leaf and a root of $\mathcal{G}_x$.

The main problem for the exploration of $\mathcal{G}_x$ is that it is not explicitly defined when the forest $\mathcal{F}_x$ is visited during the traversal of $\mathcal{T}^n$. However, as $\mathcal{F}$ is the previous forest and the parent of $\mathcal{F}_x$, we can assume that both sets $\mathcal{M}_{\mathcal{S}_{x_1}}$ and $\mathcal{M}_{\mathcal{S}_{x_2}}$ are computed, together with the LCA-mappings of their vertices, and that the leaf set $L(\mathcal{G}_x)$ is available (see Definition 9). The traversal of the whole forest $\mathcal{G}_x$ is done by a bottom-up approach in such a way that each vertex of $\mathcal{G}_x$ located at a given depth in G is visited before any vertex located at a lower depth. Such a traversal ensures that when a vertex u of $\mathcal{G}_x$ is visited, its mapping $M_{\mathcal{F}_x}(u)$ is defined (in S_x), and if the mapping $M_{\mathcal{F}_x}(s(u))$ of its sibling is not defined, then $s(u)$ is not in $\mathcal{G}_x$ and u is a root of this forest. This observation is formally stated in the Property 3 below, which also describes the calculations that are required during the traversal of $\mathcal{G}_x$ in such a way that the root set $r(\mathcal{G}_x)$ and the LCA-mappings of its vertices are computed.

Property 3. Let u be the current visited vertex during the bottom-up traversal of $\mathcal{G}_x$ and assume that its mapping $M_{\mathcal{F}_x}(u)$ is defined. If $M_{\mathcal{F}_x}(s(u))$ is defined and is in S_x, then both $s(u)$ and $p(u)$ are in $\mathcal{G}_x$ and $M_{\mathcal{F}_x}(p(u)) = x$. Otherwise, neither $s(u)$ nor $p(u)$ is in $\mathcal{G}_x$ and $u \in \mathcal{M}_{\mathcal{S}_x}$ $(= r(\mathcal{G}_x))$.

Proposition 2. *Let $\mathcal{F}_x$ be a child of an internal forest $\mathcal{F}$ of $\mathcal{T}^n$, with $S_x = (S_{x_1} + S_{x_2})$ as the branching tree, and suppose that $\mathcal{M}_{\mathcal{S}_{x_1}}$ and $\mathcal{M}_{\mathcal{S}_{x_2}}$ are given with their vertices ordered in decreasing order of their depth in G. The bottom-up traversal of $\mathcal{G}_x$, which computes the set $\mathcal{M}_{\mathcal{S}_x}$ (with the same order described above) and the LCA-mapping $M_{\mathcal{F}}(u)$ for each vertex $u \in V(\mathcal{G}_x)$, can be implemented to run in time $\Theta(|V(\mathcal{G}_x)|)$ and space $O(|V(\mathcal{G}_x)|) + \Theta(|V(\mathcal{F}_x)|)$.*

Proof. The bottom-up traversal of $\mathcal{G}_x$ is briefly described below.

1. Let Q be a queue (with the usual order) of vertices of G, initialized with $\mathcal{M}_{\mathcal{S}_{x_1}}$ and $\mathcal{M}_{\mathcal{S}_{x_2}}$.
2. A first loop on the vertices of Q is performed.
   (a) Let Q' be an empty queue.
   (b) A second loop over all the vertices u located at the front of Q and at the same depth in G, where the calculations described in Property 3 are done and $p(u)$ is inserted in Q' if and only if $p(u) \in V(\mathcal{G}_x)$.
   (c) All the vertices of Q' are moved at the front of Q.

As line 1 is done in $\Theta(|L(\mathcal{G}_x|))$ time, and all operations of Property 3, as well as line 2c, are done in constant time, the expected time complexity is immediate. The expected space complexity follows immediately from the traversal of $\mathcal{G}_x$ described above.

Definition and computation of the two CLBs. For both duplication and loss criteria, we formally define a cost for a forest $\mathcal{F}$ of $\mathcal{T}^n$, prove that it is a CLB for the considered criterion, and explain how it is computed in linear time when a child forest $\mathcal{F}_x$ of $\mathcal{F}$ is visited. Below, $S_x = (S_{x_1} + S_{x_2})$ denotes the branching tree of $\mathcal{F}_x$, $I'(G)$ denotes the subset $\{u \in V(G) \setminus L(G) : M_{\mathcal{F}}(u) \neq \varnothing\}$, and when the context is unambiguous, S refers to the species tree of $\mathcal{F}$ where the mapping $M_{\mathcal{F}}(u)$, of a vertex $u \in I'(G)$, is defined. We denote by k_G the number of apparent duplications in G, by $k_{G,\mathcal{F}}$ the number of such apparent duplications whose LCA-mapping is defined in $\mathcal{F}$, and by $d_{G,\mathcal{F}}$ the number of internal vertices of G that are duplications according to $\mathcal{F}$, that is $d_{G,\mathcal{F}} = \sum_{u \in I'(G)} d(u, S)$.

Definition 10. The duplication cost between a forest $\mathcal{F}$ of $\mathcal{T}^n$ and a gene tree G is denoted $d(G, \mathcal{F})$ and is defined as follows: $d(G, \mathcal{F}) = d_{G,\mathcal{F}} + k_G - k_{G,\mathcal{F}}$. ⋄

Together with Definition 2 and the fact that each apparent duplication $u \in I(G)$ is such that $d(u, S) = 1$ for any species tree $S \in \mathcal{K}^n$, $d(G, \mathcal{F})$ is obviously a CLB for $d(G, S)$. The inconvenient of this CLB, when considering only the vertices of G that are not apparent duplication, is that it requires the LCA-mapping of such a vertex to determine if it is a duplicated gene for any species tree located below the considered forest $\mathcal{F}$. Moreover, it is important to assess the efficiency of this CLB against the one that does not take advantage of the constant cost induced by the apparent duplications (such a CLB is equal to $d_{G,\mathcal{F}}$). Recall that all k_G apparent duplications are considered in the cost $d(G, \mathcal{F})$ of any forest $\mathcal{F}$ of $\mathcal{T}^n$, which means that the CLB described in Definition 10 can be reduced solely to the non-apparent duplications. Formally, if $\mathcal{F}$ is the current visited forest and S^* is the best species tree found since the beginning of the traversal of $\mathcal{T}^n$, then $\mathcal{F}$ is pruned if and only if $d_{G,\mathcal{F}} - k_{G,\mathcal{F}} \geq d(G, S^*) - k_G$. In other words, the advantage given by the k_G apparent duplications present in G can not be evaluated by how large k_G is, as the efficiency of the CLB $d(G, \mathcal{F})$ to cut the subtree $\mathcal{T}^n_{\mathcal{F}}$ depends on the number of non-apparent duplications induced by $\mathcal{F}$ (i.e. $d_{G,\mathcal{F}} - k_{G,\mathcal{F}}$) and S^* (i.e. $d(G, S^*) - k_G$).

The corollary below explains how the CLB $d(G, \mathcal{F})$ can be computed with the same complexities as for the traversal of $\mathcal{G}_x$ (see Proposition 2).

Corollary 1. *If the duplication cost $d(G, \mathcal{F})$ is given, then $d(G, \mathcal{F}_x)$ can be computed in time $\Theta(|V(\mathcal{G}_x)|)$ and space $O(|V(\mathcal{G}_x)|) + \Theta(|V(\mathcal{F}_x)|)$.*

Proof. Let $d(\mathcal{G}_x)$ denotes the number of vertex u of G that is a duplication according to S_x and such that its LCA-mapping is (resp. not) defined in $\mathcal{F}_x$ (resp. $\mathcal{F}$), that is $M_{\mathcal{F}_x}(u) = x$, $M_{\mathcal{F}}(u) = \varnothing$ and $d(u, S_x) = 1$. Hence, according to Definition 9, u is in $\mathcal{G}_x$. It is then immediate that $d(G, \mathcal{F}_x) = d(G, \mathcal{F}) + d(\mathcal{G}_x)$, where $d(\mathcal{G}_x)$ is computed by the traversal of $\mathcal{G}_x$, and the expected complexities follow immediately from Proposition 2. To handle apparent duplications, detecting them can be done during a preprocessing phase, which implies that updating $k_{G,\mathcal{F}}$ when visiting a new forest can be done in time linear in the number of apparent duplications whose mapping is defined.

Definition 11. The loss cost between a forest $\mathcal{F}$ of $\mathcal{T}^n$ and a gene tree G is denoted $l(G, \mathcal{F})$ and is defined as follows: $l(G, \mathcal{F}) = \sum_{u \in I'(G)} l(u, S) + \sum_{u \in I(G) \setminus I'(G)} (d_{\mathcal{F}}(u_1) + d_{\mathcal{F}}(u_2))$. ⋄

From Definitions 3 and 5 and Property 2, Corollary 2 below is immediate.

Corollary 2. *$l(G, \mathcal{F})$ is a CLB for $l(G, S)$.*

Corollary 3. *If the loss cost $l(G, \mathcal{F})$ is given, then $l(G, \mathcal{F}_x)$ can be computed in time $\Theta(|V(\mathcal{G}_x)|)$ and space $O(|V(\mathcal{G}_x)|) + \Theta(|V(\mathcal{F}_x)|)$.*

Proof. Let $l(\mathcal{G}_x)$ denotes the number of vertex u of $\mathcal{G}_x$ such that either $l(u, S_x)$ corresponds to the second case of Definition 3 or $\mathcal{F}_x$ is an incremental forest for u (see Definition 8). It is then immediate that $l(G, \mathcal{F}_x) = l(G, \mathcal{F}) + l(\mathcal{G}_x)$, where $l(\mathcal{G}_x)$ is computed by the traversal of $\mathcal{G}_x$. Because both cases above can be verified in constant time ($\mathcal{F}_x$ is an incremental forest for u if and only if u is both a leaf and a root of $\mathcal{G}_x$), the expected complexities follow immediately from Proposition 2.

Proposition 1, together with Corollaries 1, 2 and 3, gives the fundamental properties of the complexity of our Branch-and-Bound algorithm: (1) the exploration of the visited species forests requires a time linear in the number of these forests, (2) visiting a new species forest while updating the appropriate CLB requires a time linear in the number of vertices of gene forest whose LCA-mapping is updated and (3) the total space complexity is linear in the number of considered taxa. We do not have theoretical properties of the two CLBs we introduced, and we will assess how efficient they are to cut large subspaces of $\mathcal{T}^n$ experimentally in Section 4.

Accounting for prior knowledge on the considered species tree. In the context of some prior knowledge on the species tree for the considered n genomes, let S be a multifurcating (non-binary) species tree that describes such prior and $\mathcal{K}^n(S)$ be the subset of $\mathcal{K}^n$ that contains each (binary) species tree that is consistent with S. Such prior information can for example consists in well defined clades of taxa. The question that we address here is how to define an architecture that allows to exhaustively explore the subset $\mathcal{K}^n(S)$? Note that it can lead to a species tree that is not globally optimal, but is optimal among the species tree of $\mathcal{K}^n(S)$. The solution resides in the use of Definition 7 to exhaustively enumerate all the binary species trees for the m children of an internal vertex x of S that is not resolved. Formally, let $\{t_1, t_2, \ldots, t_m\}$ be the set of the m subtrees, where $m > 2$, rooted at the children of x. The original architecture $\mathcal{T}^m$ (see Definition 7) is used to exhaustively enumerate all the binary species trees for the unresolved vertex x of S as follows: $\mathcal{T}^m$ is rooted at the forest $\{t_1, t_2, \ldots, t_m\}$, and each leaf forest is composed of a single binary tree that contains one and only one leaf labeled by the subtree t_i, for each $1 \leq i \leq m$.

By recursively applying this process on all the unresolved vertices of S, it is easy to see that the induced architecture has $\mathcal{K}^n(S)$ as its leaf set. Moreover, given that all such unresolved vertices of S are ordered according to the prefix traversal of S, the time and space complexities for the exploration of this adapted architecture are the same as in Proposition 1.

4 Experimental results

We considered 1111 gene trees from the TreeFam database [11, 17], more specifically the ones of the TreeFam-B families that have been manually corrected by experts and contain gene families from 29 eukaryotic genomes. Table 3 (Appendix) lists the considered species, together with their abbreviations, and Figure 5 (Appendix) describes the size distribution of the gene trees. The corresponding reference species tree, denoted by S_0, is depicted in Appendix (Figure 4) and corresponds to the NCBI taxonomy tree [19], except that three nodes of the tree were considered as multifurcations due to different phylogenetic hypothesis regarding the corresponding clades (see [11, 17]).

Study of a subset of 8 *species.* First, to gain some insight on the whole space of species trees for a given dataset, we selected $n = 8$ species from the 29 considered genomes (see Figure 4), removed from the gene trees all genes from other species, and performed an exhaustive exploration of the 135135 species trees. The aim is to study the shape of the space $\mathcal{K}^n$ according to the three combinatorial criteria we consider in order to evaluate their performance for phylogenetic inference. For a species tree S of $\mathcal{K}^n$, $c(G, S)$ denotes the considered cost (either l, d, or m), S_{min} (resp. S_{max})

a tree of $\mathcal{K}^n$ that minimizes (resp. maximizes) this cost and $c_{norm}(G,S)$ the normalized cost over the range of possible values and defined as follows: $c_{norm}(G,S) = (c(G,S) - c(G,S_{min}))/(c(G,S_{max}) - c(G,S_{min}))$. The similarity between S and the species tree S_0 is evaluated by the classical Robinson and Foulds distance between phylogenetic trees [15, 16], denoted $RF(S_0,S)$. For the three costs, Figures 3 (left) below depicts the distribution of each tree $S \in \mathcal{K}^n$ according to the normalized cost $l_{norm}(G,S)$.

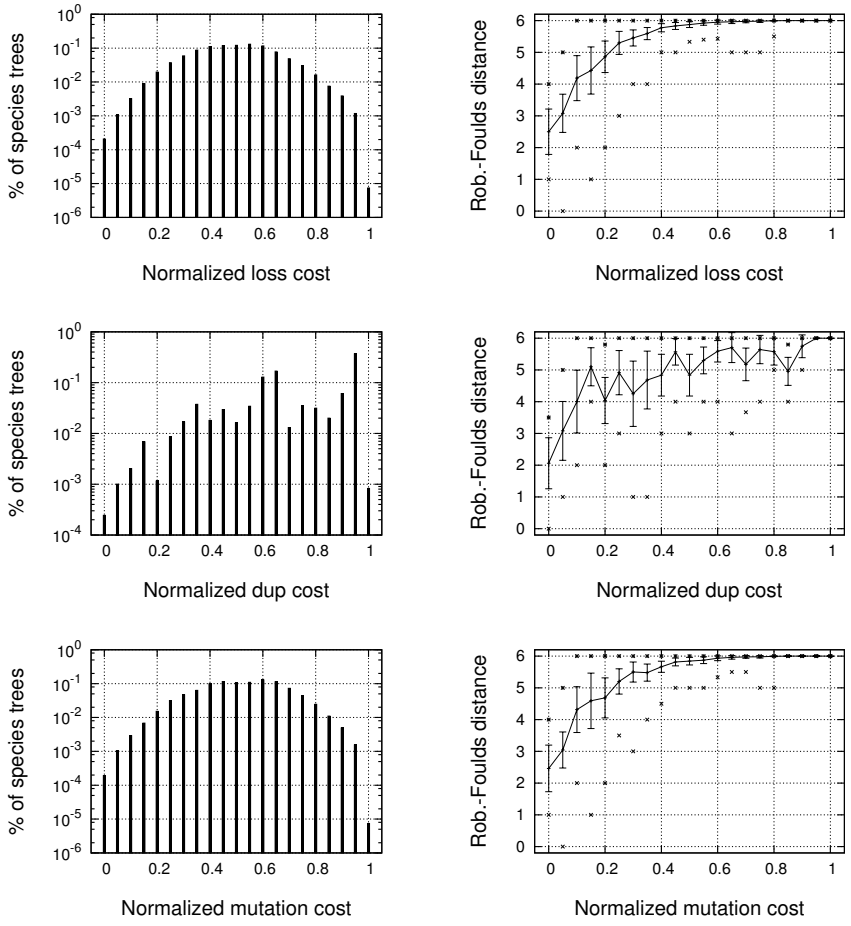

Fig. 3. (Left) The percentage of species tree S of $\mathcal{K}^n$ (y axis) such that $0.05\ i \leq c_{norm}(G,S) < 0.05(i+1)$ (x axis) and (Right) the average, standard deviation, and minimum and maximum values of the Robinson and Foulds distance with S_0 for each such tree. Top: loss cost. Middle: duplication cost. Bottom: mutation cost.

It appears clearly from Table 1 that, on this dataset, the duplication cost seems to be slightly better than the two other criteria, although in terms of normalized cost, the difference is relatively marginal. We can also observe that for the loss and mutation cost, the loss cost distribution is similar to a normal and almost symmetrical distribution with a mean located around 0.5, while the distribution for the duplication cost is less smooth. Over each species tree S of $\mathcal{K}^n$ with the same normalized cost $c_{norm}(G,S)$, we also computed the average distance $RF(S_0,S)$, and according to Figure 3 (right), the loss cost is clearly correlated with the R.F. distance to the tree S_0, which is not as clear with the duplication cost. Also note that, due to the fact that losses are more numerous than duplications, the properties of the mutation cost are very similar to the loss cost. Finally, we

	$c(G, S_{min})$	$c(G, S_{max})$	$c(G, S_0)$	$RF(S_0, S_{min})$
Loss	3577	35072	5226 (0.05)	3
Dup.	2229	6313	2425 (0.04)	1
Mut.	5812	41355	7651 (0.05)	3

Table 1. Minimum (col. 1) and maximum (col 2.) costs for the loss, duplication, and mutation criteria. Col. 3: costs of S_0, both absolute and normalized. Col. 4: R.F. distance between the optimal solution and S_0.

can notice that S_0 is close to the optimal species tree, both in terms of duplication and/or loss events, and in the RF distance.

Study of the whole 29 taxa dataset. Next, we attacked the problem of computing a parsimonious species tree for the 29 considered genomes. There are about 10^{36} possible species trees, and the Branch-and-Bound starting from the root of $\mathcal{T}^n$ was not completed after a few days of computation. There are two reasons for this problem: first, during the traversal of $\mathcal{T}^n$, the CLB of the newly visited forest is computed in linear time; second, the subtree of $\mathcal{T}^n$ induced by the pruned forests is not small enough for an exhaustive exploration.

We then decided to reduce the number of considered species trees by integrating prior information on the seeked species tree. For our experiments, we defined such prior information from the species tree S_0 as follows. A species subset denoted $\pi \subseteq \Lambda(S_0)$ is said to be consistent with a given gene tree G if and only if its intersection π' with $\Lambda(G)$ is consistent with each vertex $u \in V(G)$, that is either $\pi' \subseteq \Lambda(G_u)$, $\Lambda(G_u) \subseteq \pi'$, or $\Lambda(G_u) \cap \pi' = \emptyset$. Hence, the more a clade is respected among the considered gene trees, the more probable it is present in an optimal solution of the GTP problem. We found 19 clades, which can either be disjoint or included one into the other, that are consistent with a majority of the 1111 gene trees, and defined four species trees, denoted S_i for $i \in \{1, 2, 3, 4\}$, as follows: S_1 is consistent with all and only all the 19 clades, and for $i = 2, 3, 4$, one clade is removed from S_{i-1} to define S_i (see Appendix, Figures 6 and 7). Recall that $\mathcal{K}^n(S_i)$ denotes the subset of species trees that are consistent with a given tree S_i (see Section 3), then the five subsets of $\mathcal{K}^n$ are embedded as follows: $\mathcal{K}^n(S_i) \subset \mathcal{K}^n(S_{i+1})$, for $0 \leq i \leq 3$. Hence, $\mathcal{K}^n(S_0)$ (resp. $\mathcal{K}^n(S_4)$) is the smallest (resp. largest) set of species trees, and similarly is the induced space tree defined at the end of Section 3, which is used to solve the GTP problem on the considered reduced set of species trees. For S_1, S_2, S_3, and S_4, the number of possible species trees is respectively 127575, 893025, 9823275 and 29469825. For the three usual criteria, we applied the Branch-and-Bound to solve the GTP problem first on the smallest set $\mathcal{K}^n(S_0)$, and then the optimal solution for $\mathcal{K}^n(S_i)$, with increasing index i from 0 to 3, was then used as the first upper bound for the Branch-and-Bound applied on $\mathcal{K}^n(S_{i+1})$. For each criterion and each constrained species tree, the result are summarized in Table 2 below.

For the duplication criterion, the best solution found in $\mathcal{K}^n(S_1)$ and $\mathcal{K}^n(S_2)$ is the same and is depicted in Figure 8 (see Appendix A), and the Branch-and-Bound applied on $\mathcal{K}^n(S_3)$ (resp. $\mathcal{K}^n(S_4)$) was not completed after 4 days CPU time, and this is caused (as previously explained) by the slow rise of the CLB for the number of duplications according to the one for the losses. Moreover, among all the 20942 internal vertices, there is 3693 apparent duplications. For the loss and mutation criteria and the four sets $\mathcal{K}^n(S_i)$, the optimal solution is the same and is depicted in Figure 8. For both criteria, this means that the optimal solution for the GTP applied on the largest set $\mathcal{K}^n(S_4)$ can be found solely by applying the Branch-and-Bound on the smallest set $\mathcal{K}^n(S_1)$, although that its optimality status requires the use of $\mathcal{K}^n(S_4)$. The Robinson and Foulds distance between the two optimal solutions for loss (i.e. and mutation) (in $\mathcal{K}^n(S_4)$) and duplication (in

	Optimal cost in $\mathcal{K}^n(S_i)$		CPU time (in seconds)				
	S_0	$S_1 \dots S_4$	S_0	S_1	S_2	S_3	S_4
Loss	22464	21257	54	3147	7897	26444	59997
Mut	27691	26328	49	3944	10697	39742	94429
Dup	5140	4941	50	7296	32117	?	?

Table 2. For each criterion and each constrained species tree S_i, the **optimal cost** for the GTP problem applied on the set of allowed solutions $\mathcal{K}^n(S_i)$ and the **CPU time** used by the Branch-and-bound. For each of the three criteria, the optimal cost in $\mathcal{K}^n(S_i)$, for $i = 1, 2, 3,$ and 4, is the same. The '?' character indicates that the Branch-and-bound process was not terminated after 4 days, where the optimal solution of $K^n(S_2)$ was the best solution found so far for both process.

$\mathcal{K}^n(S_2)$) criteria is 4, while their distance with S_0 (i.e. the reference species tree) respectively is 5 and 3.

We applied Duptree [21], which is based on a Randomized hill climbing heuristic, on our 1111 gene trees to solve the GTP problem for the duplication criterion. First, when the topological constraints of S_4 are used to reduce the search space (that is $\mathcal{K}^n(S_4)$ as for our Branch-and-Bound), duptree found the same solution as our approach applied on $\mathcal{K}^n(S_2)$ (see Table 2) within solely 2 seconds and 6 rSPRs. Second, when the program is applied without prior knowledge on the species phylogeny, the same solution is found within 3 seconds and 8 rSPRs.

5 Conclusion

In the current work, we described a Branch-and-Bound algorithm that seeks a parsimonious species tree, given a set of gene trees, according to the number of duplications and/or losses. Up to now, two (resp. one) Fixed-Parameter Tractable algorithms exists for the duplication (resp. mutation) criterion [13, 20] (resp. [13]), and there is no exact approach for the loss criterion, which appears to be relevant for phylogenetic inference according to [4]. Whereas these FPT algorithms are not suitable for phylogenetic inference problems with several genomes, we demonstrated the applicability of our Branch-and-Bound on a large dataset of 29 eukaryotic genomes to obtain the optimal species tree given prior constraints on the species phylogeny. For both loss and mutation criteria, our approach found the optimal species tree on a real and large dataset. Moreover, as Duptree does not consider the number of duplications and losses, it can not be used to validate the optimal solution (for the mutation criterion) proposed in this work.

Our results show that the species phylogeny in TreeFam is very close (but different) to the optimal species tree. This suggests that near-optimal species trees are worth to be considered when looking for a species tree from gene families, and that methods that explore the neighborhood of a given species tree (here the optimal one) are pertinent. Local-search heuristics such as the ones described in [1, 2] are natural candidates. However, it is important to recall that such approaches, in order to assess the computational quality of the produced species trees, need to be complemented by methods that compute an optimal species tree (or an optimal among a large set of considered species tree), which we described here.

A possible direction for further research is to consider multiple gene duplication episodes, during which a large portion of an organism's genome is duplicated. Given a species tree S and a set of gene trees, [3] proposed the first exact and efficient algorithm that locates in S gene duplication events in such a way that the number of multiple gene duplication episodes is minimized. However, when the species phylogeny is unknown, inferring the most parsimonious species tree according to the multiple gene duplication criterion is still an open problem.

References

1. M.S. Bansal, J.G. Burleigh, O. Eulenstein, and A. Wehe. Heuristics for the gene-duplication problem: A $\theta(n)$ speed-up for the local search. In *Research in Computational Molecular Biology, 11th Annual International Conference, RECOMB 2007, Oakland, CA, USA, April 21-25, 2007, Proceedings*, volume 4453 of *Lecture Notes in Computer Science*, pages 238–252. Springer, 2007.
2. M.S. Bansal, O. Eulenstein, and A. Wehe. The gene-duplication problem: Near-linear time algorithms for nni-based local searches. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(2):221–231, 2009.
3. Mukul S. Bansal and Oliver Eulenstein. The multiple gene duplication problem revisited. *Bioinformatics*, 24(13):i132–i138, 2008.
4. C. Chauve, J.P. Doyon, and N. ElMabrouk. Gene family evolution by duplication, speciation and loss. *Journal of Computational Biology*, 15(8):1043–1062, 2008.
5. C. Chauve and N. ElMabrouk. New perspectives on gene family evolution: Losses in reconciliation and a link with supertrees. In *Research in Computational Molecular Biology, 13th Annual International Conference, RECOMB 2009, Tucson, AZ, USA, May 18-21, 2009. Proceedings*, volume 5541 of *Lecture Notes in Computer Science*, pages 46–58. Springer, 2009.
6. J. Felsenstein. *Inferring Phylogenies*. Sinauer Associates, 2003.
7. W.M. Fitch. Homology a personal view on some of the problems. *Trends in Genetics*, 16:227–231, 2000.
8. M. Goodman, J. Czelusniak, G.W. Moore, R.A. 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.
9. D. Graur and W.-H. L. *Fundamentals of Molecular Evolution*. Sinauer Associates, second edition edition, 1999.
10. M.D. Hendy and D. Penny. Branch and bound algorithms to determine minimal evolutionary trees. *Mathematical Biosciences*, 59, 1982.
11. H. Li and *et al.* Treefam: a curated database of phylogenetic trees of animal gene families. *Nucleic Acids Research*, 34(Database-Issue):572–580, 2006.
12. B. Ma, M. Li, and L. Zhang. From gene trees to species trees. *SIAM Journal on Computing*, 30(3):729–752, 2000.
13. Hallett M.T. and Lagergren J. New algorithms for the duplication-loss model. In *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology, RECOMB 2000, April 8-11, 2000, Tokyo, Japan*, pages 138–146. ACM Press, 2000.
14. R.D.M. Page. GeneTree: comparing gene and species phylogenies using reconciled trees. *Bioinformatics*, 14(9):819–820, 1998.
15. N.D. Patengale, E.J. Gottlieb, and B.M.E. Moret. Efficiently computing the robinson-foulds metric. *Journal of Computational Biology*, 14(6):724–735, 2007.
16. D.F. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53:131–147, 1981.
17. J. Ruan and *et al.* TreeFam: 2008 Update. *Nucleic Acids Research*, 36(Database issue):D735–D740, 2008.
18. M. Sanderson and M. Mcmahon. Inferring angiosperm phylogeny from est data with widespread gene duplication. *BMC Evolutionary Biology*, 7(Suppl 1):S3, 2007.
19. E.W. Sayers and *et al.* Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 37(Database issue):D5–D15, 2009.
20. Ulrike Stege. Gene trees and species trees: The gene-duplication problem is fixed-parameter tractable. In *Proceedings of the 6th International Workshop on Algorithms and Data Structures (WADS'99*, pages 166–3, 1999.
21. A Wehe, M.S. Bansal, J.G Burleigh, and O. Eulenstein. DupTree: a program for large-scale phylogenetic analyses using gene tree parsimony. *Bioinformatics*, 24(13):1540–1541, 2008.
22. L. Zhang. On a mirkin-muchnik-smith conjecture for comparing molecular phylogenies. *Journal of Computational Biology*, 4(2):177–187, 1997.

A Appendix. Additional tables and figures

Arath	Arabidopsis thaliana	Orysa	Oryza sativa
Schpo	Schizosaccharomyces pombe	Yeast	Saccharomyces cerevisiae
Schma	Schistosoma mansoni	Caebr	Caenorhabditis briggsae
Caeel	Caenorhabditis elegans	Caere	Caenorhabditis remanei
Aedae	Aedes aegypti	Anoga	Anopheles gambiae
Drome	Drosophila melanogaster	Drops	Drosophila pseudoobscura
Cioin	Ciona intestinalis	Ciosa	Ciona savignyi
Brare	Danio rerio	Fugru	Fugu rubripes
Tetng	Tetraodon nigroviridis	Gasac	Gasterosteus aculeatus
Oryla	Oryzias latipes	Xentr	Xenopus tropicalis
Chick	Gallus gallus	Mondo	Monodelphis domestica
Bovin	Bos taurus	Canfa	Canis familiaris
Mouse	Mus musculus	Rat	Rattus norvegicus
Macmu	Macaca mulatta	Human	Homo sapiens
Pantr	Pan troglodytes		

Table 3. The list of the 29 considered species, with their names (second and fourth columns) and their abbreviations (first and third columns).

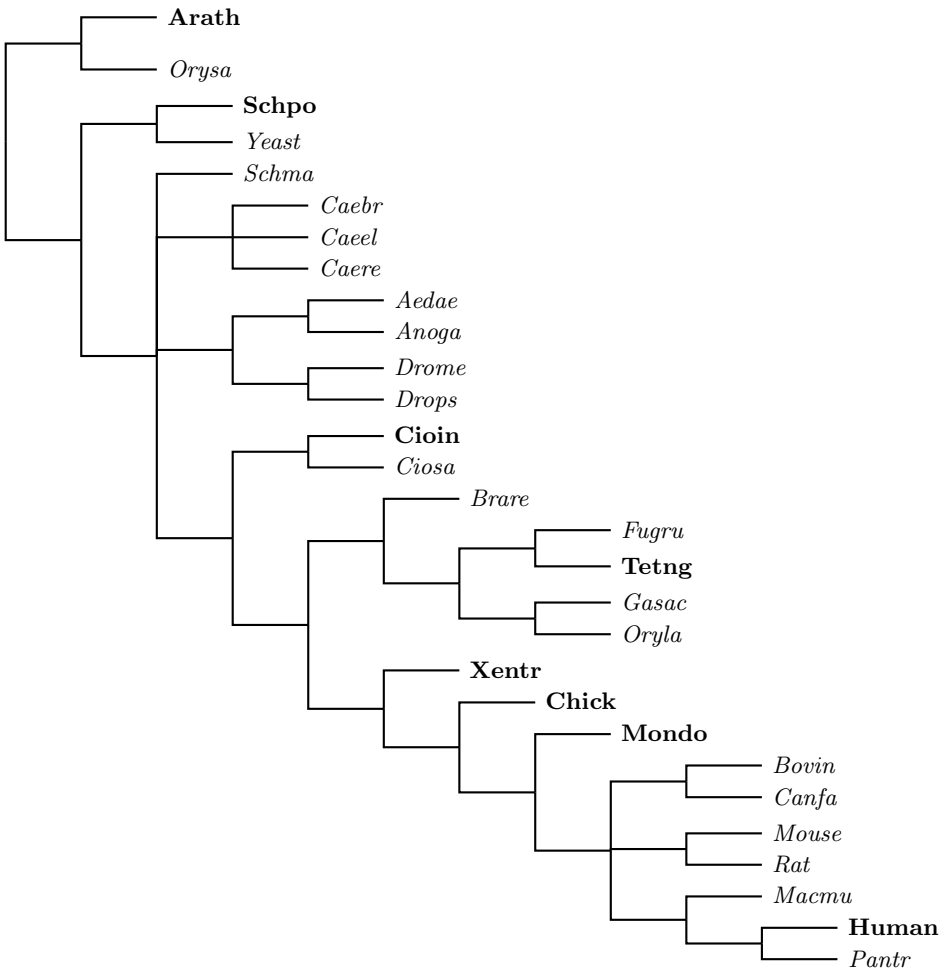

Fig. 4. The species tree for the 29 animals considered in the experiments. Species in boldface corresponds to the eight species used for an exhaustive exploration of the space $\mathcal{K}^n$, where $n = 8$ (See Section 4).

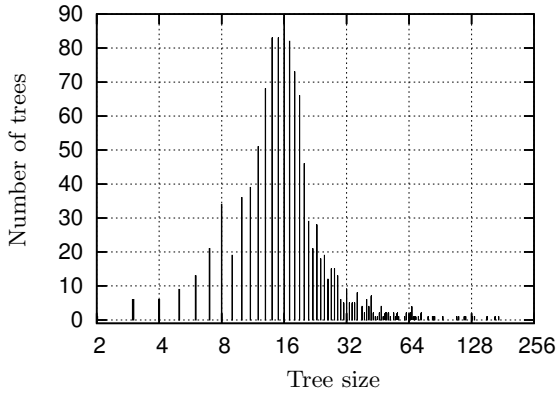

Fig. 5. Distribution of the 1111 gene trees (y axis) according to their number of leaves (x axis).

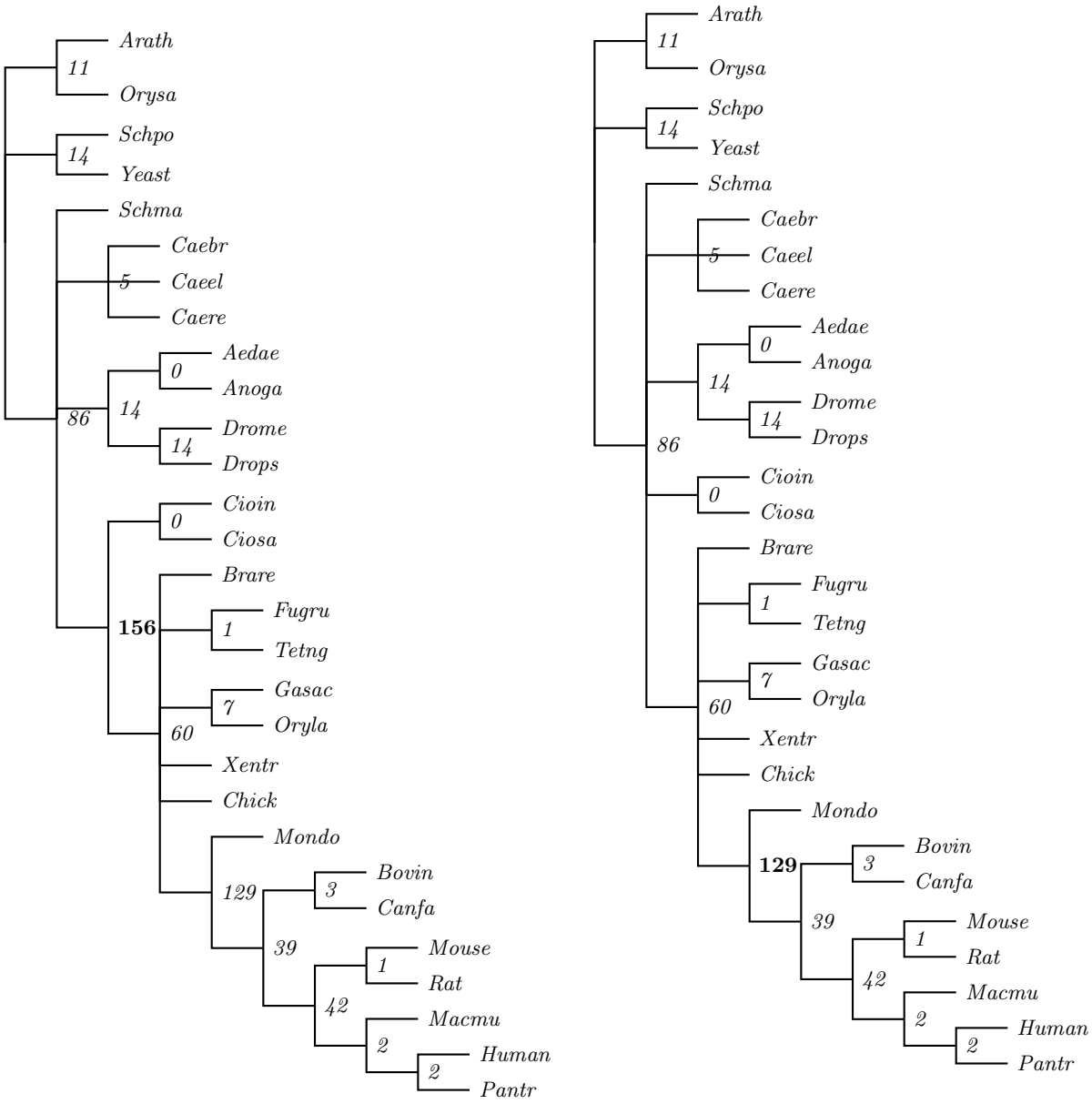

Fig. 6. Constrained species trees S_1 (**left**) and S_2 (**right**), where the integer beside each vertex indicates the number of trees among the 1111 considered ones that are not consistents with the corresponding clade and the integer in boldface in S_1 (resp. S_2) corresonds to the clade that is not present in S_2 (resp. S_3).

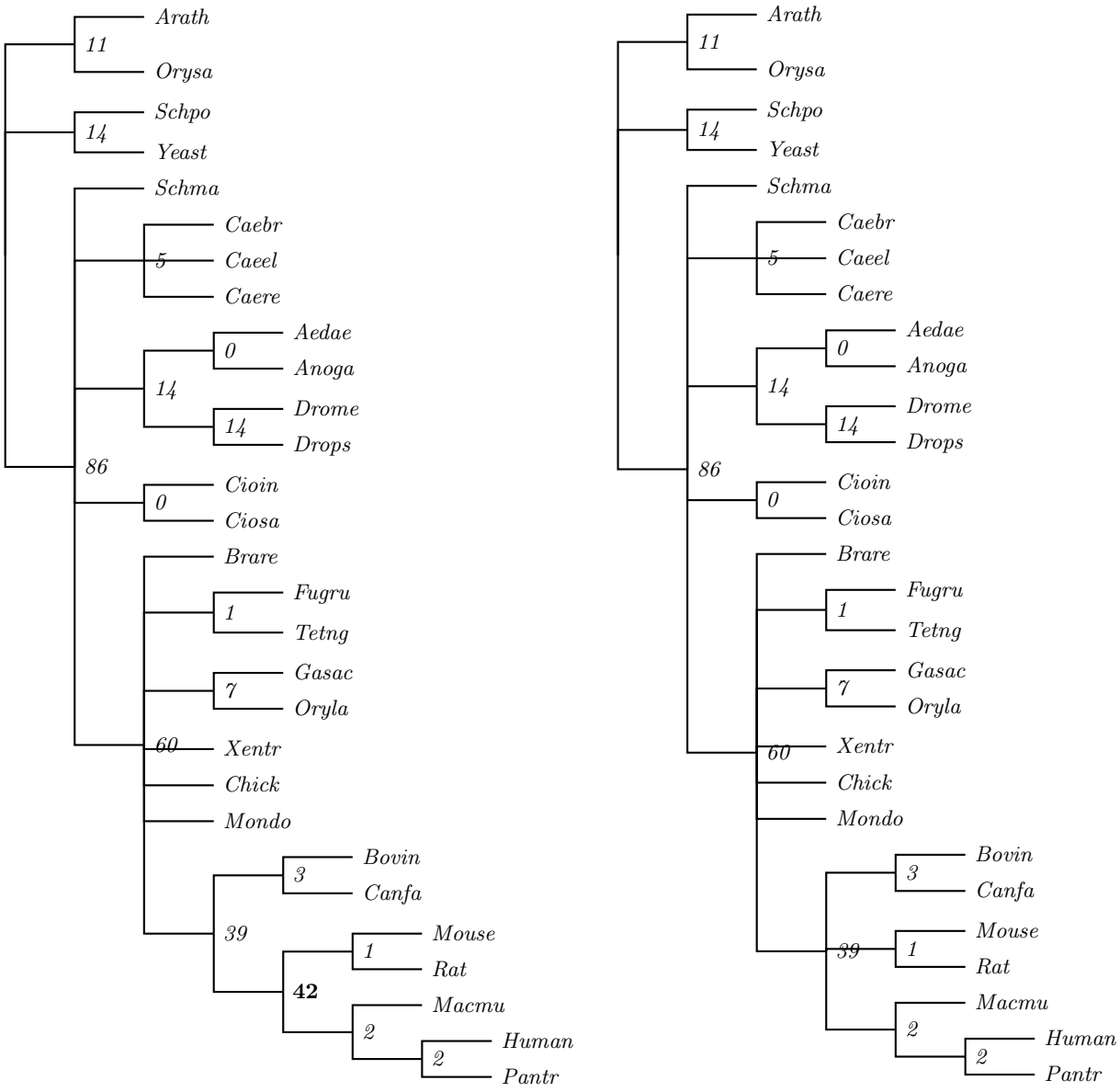

Fig. 7. Constrained species trees S_3 (**left**) and S_4 (**right**).

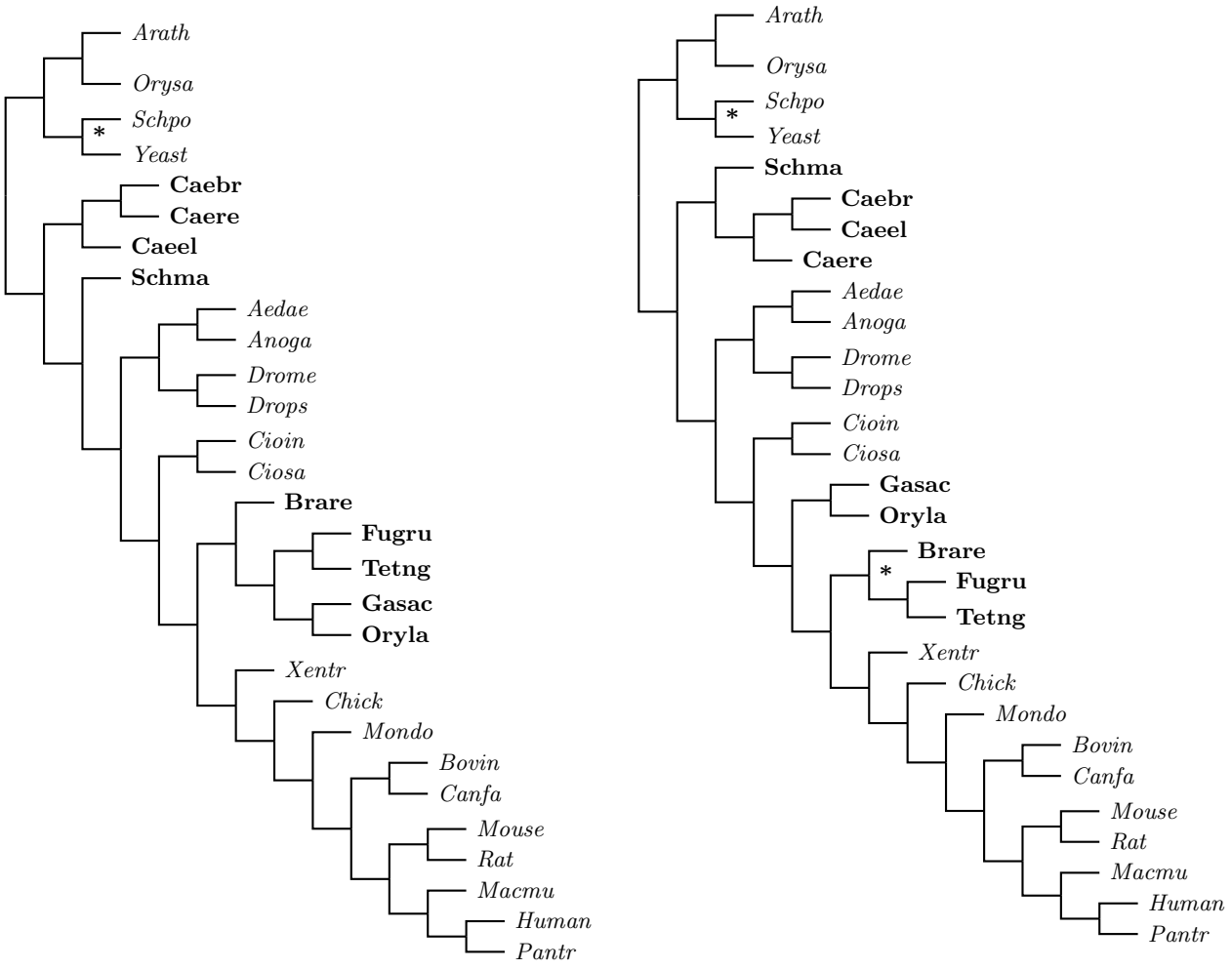

Fig. 8. Left (resp. right): optimal solution for the duplication (resp. loss and mutation) criterion. The character '*' indicates a clade that is misplaced according to the proposed phylogeny (Figure 4) and the taxa in boldface point out the disagreements between the two optimal solutions.

CHAPITRE 9

CONCLUSION

9.1 Modèles combinatoires de réconciliation

Grâce à notre modèle combinatoire de réconciliation entre un arbre de gènes et un arbre d'espèces, nous avons développé au chapitre 6 des algorithmes efficaces pour calculer le nombre de réconciliations, générer aléatoirement une réconciliation selon la distribution uniforme et explorer l'espace des réconciliations (sous-optimales). Nous avons aussi démontré comment l'algorithme de dénombrement s'applique aux réconciliations sous-optimales pour le coût de duplication, mais pas pour le coût de perte. La monotonie croissante du nombre de pertes étant déjà respectée, d'autres propriétés combinatoires devront donc être étudiées. Aussi, notre modèle s'applique à des arbres enracinés et binaires et ne prends en considération ni les duplications multiples, ni les transferts (surtout fréquents chez les procaryotes [12]). Nous décrivons ci-dessous les principales avancées théoriques des modèles de réconciliation qui considère ces aspects.

Lorsque l'arbre d'espèces seul est non-binaire, le couplage LCA défini en [90] permet de calculer en temps polynomial le nombre minimum de duplications et de pertes pour une réconciliation avec l'arbre de gènes. En utilisant ce couplage, notre définition de réconciliation et nos outils combinatoires peuvent être généralisés aux arbres d'espèces avec multifurcations. Comme nous l'avons vu au chapitre 6, l'exploration efficace des réconciliations sous-optimales requiert une monotonie croissante du coût considéré, il faudra donc vérifier si cette propriété est vérifiée par les coûts induits de ce couplage. Si l'arbre de gènes seul est non-binaire, calculer un arbre binaire, cohérent avec l'arbre original et de coût de duplication minimum est NP-complet [13]. Pour les duplications multiples de gènes [32, 43], un algorithme efficace calcule le minimum requis pour la réconciliation de plusieurs familles de gènes avec un arbre d'espèces donné [9]. Lorsque

l'arbre de gènes n'est pas enraciné, un problème naturel et résolu en temps linéaire est de calculer la racine dont la réconciliation LCA minimise le coût considéré [21].

Contrairement aux modèles de réconciliation avec duplications et pertes de gènes, calculer une réconciliation parcimonieuse est NP-complet [39, 87] lorsque les transferts sont considérés. La raison d'une telle dichotomie est qu'un transfert induit le même temps [1] pour les deux espèces impliquées et qu'une réconciliation doit être cohérente selon les temps induits par les transferts. Si cette contrainte n'est pas exigée, le problème est résolu en temps polynomial [87]. Sinon, deux méthodes sont proposées. La première est une structure nommée *graphe d'espèces* [39, 45], où les transferts sont possibles seulement pour les paires d'espèces jointes par une nouvelle branche ajoutée à l'arbre d'espèces. La deuxième méthode utilise un arbre d'espèces où les événements de spéciation sont datés [59, 69]. Pour les modèles avec duplications et pertes, seulement la réconciliation LCA est parcimonieuse pour les trois critères, alors que cette unicité est perdue pour un modèle avec transferts. C'est pourquoi les outils combinatoires de notre modèle devront être développés pour étudier cet espace des réconciliations parcimonieuses avec transferts. Selon un modèle combinatoire non-ambigu [2] défini pour un arbre d'espèces daté, l'algorithme présenté en [28] calcule en temps polynomial une réconciliation parcimonieuse et pourra servir de base pour étudier cet espace.

9.2 Modèles probabilistes de réconciliation

Nous avons présenté au chapitre 7 les premières analyses à grande échelle de l'espace des réconciliations et de leur probabilités postérieures. Nos résultats montrent que la réconciliation LCA est généralement la plus probable et qu'une sous-arborescence de petite taille couvre la masse probabiliste de toute l'espace. Ceci nous a permis de calculer efficacement et précisément les probabilités de l'arbre de gènes et des réconciliations les plus probables. Nous avons aussi observé quelques arbres de gènes, générés avec

[1] C'est-à-dire durant l'évolution représentée par la phylogénie d'espèces.
[2] Contrairement aux autres modèles [59, 69, 87].

des taux réalistes, pour lesquels la réconciliation LCA n'est pas la plus probable. En calculant efficacement la réconciliation la plus vraisemblable (voir le théorème 3.4) et en l'utilisant comme racine de l'arborescence, notre approche demeure appropriée même pour ces arbres.

Il faudra ensuite détecter les arbres de gènes dont l'espace des réconciliations contient plusieurs maximums locaux, caractériser les données d'entrées pour lesquelles les probabilités estimées par notre approche risquent d'être moins précises et évaluer leur fréquence afin de décider s'il faut leur accorder une attention particulière. Autrement dit, si de tels arbres de gènes ont une probabilité exacte très faible et sont minoritaires parmi l'ensemble des arbres générés sous les mêmes conditions, alors notre méthode pourra être utilisée dans la plupart des cas. Enfin, connaître la forme de l'espace de recherche et le nombre de maximums locaux sont des atouts importants pour étudier la convergence et l'efficacité d'une approche MCMC.

Afin de compléter nos résultats, des expériences plus approfondies sont nécessaires. En particulier, varier les taux pour tout l'arbre d'espèces ou pour certaines de ses branches, varier les taux de duplication par rapport à ceux de perte et choisir des arbres d'espèces de taille et de forme (filiforme, parfaitement balancé, etc.) différentes. Pour ce qui est des résultats théoriques, il faudra développer des propriétés sur la récursion utilisée pour calculer la réconciliation la plus vraisemblable. Un résultat théorique démontrant que la réconciliation LCA est la plus probable, pour certains types de données d'entrées, serait une contribution importante au modèle probabiliste d'Arvestad *et al.* En plus de fournir une explication théorique de nos observations sur la masse probabiliste et une première caractérisation de jeux de données où elles ne seraient pas observées.

Selon un arbre d'espèces daté et un modèle d'évolution de gènes qui intègre l'évolution des séquences et des taux de substitution, de duplication et de perte (de gènes), Akerborg *et al.* [1] présente un MCMC pour estimer les probabilités conjointes de séquences moléculaires et d'un arbre de gènes. Pour des familles de gènes avec au plus 17 gènes ou avec exactement un gène par espèce, leurs expériences montrent qu'une ap-

proche minimisant le coût de réconciliation[3] propose généralement l'arbre le plus probable. Alors que pour la majorité des familles plus grandes, l'arbre calculé par cette approche de parcimonie n'est pas présent dans la distribution postérieure.

Les observations d'Akerborg *et al.* s'accordent avec nos résultats expérimentaux sur la probabilité postérieure de la réconciliation LCA. C'est pourquoi notre méthode peut servir à la détection et la correction d'erreurs dans un arbre de gènes dont le signal phylogénétique n'est pas clair. Une telle approche serait basée sur l'exploration du voisinage, où un nouvel arbre de gènes est valué selon soit la probabilité de la réconciliation LCA, soit la masse probabiliste d'une sous-arborescence de profondeur maximale. Pour développer une telle approche, il sera nécessaire d'étudier la complexité en temps pour mettre à jour la réconciliation LCA et sa vraisemblance après l'application d'un opérateur (NNI [11] ou SPR [8]) sur l'arbre de gènes. Enfin, selon un ensemble de familles de gènes, une approche MCMC pour estimer les probabilités postérieures des taux de duplication et de perte [14] serait basée sur des critères similaires.

Notre modèle probabiliste pour l'évolution d'une famille de gènes permet une variation des taux (de duplication et de perte) entre branches de l'arbre d'espèces (horloge moléculaire relaxée [29]) mais pas au cours du temps (hétérotachie [61]). Une telle hétérogénéité temporelle peut être dûe aux changements de taille de la population d'une espèce [62] et aux forces de sélection naturelle [95]. Les taux de duplication et de perte peuvent aussi varier entre différentes familles selon les fonctions des gènes [25]. Un modèle probabiliste de réconciliation considérant ces aspects sera plus réaliste face à l'évolution génomique.

[3] Cette méthode, nommée SYNERGY [93], propose un arbre de gènes (en plus de relations d'orthologie) en utilisant le coût de réconciliation, la similarité de séquences et la conservation du voisinage des gènes (synténie [34]).

9.3 Méthode exacte pour le problème GTP

Nos résultats expérimentaux sur les probabilités postérieures des réconciliations parcimonieuses justifient l'utilisation de la parcimonie pour inférer un arbre d'espèces selon un ensemble d'arbres de gènes. Nous avons proposé au chapitre 8 une méthode de "Branch-and-Bound" pour résoudre ce problème et nos résultats préliminaires sont positifs selon la similarité entre les arbres d'espèces proposés et la phylogénie de référence. Pour obtenir des résultats avec des temps de calcul raisonables, nous avons réduit la taille de l'arborescence d'arbres d'espèces en imposant des clades (d'espèces) respectés par une majorité d'arbres de gènes. Une approche moins drastique sera de réduire la taille d'une sous-arborescence durant le "Branch-and-Bound". Nous avons aussi observé que le problème est plus difficile à résoudre pour le coût de duplication.

De manière similaire au programme *DupTree* [94], notre "Branch-and-Bound" devra associer un poids à chaque arbre de gènes et le prendre en considération dans le coût total de réconciliation avec un arbre d'espèces. Pour un arbre de gènes donné, un tel poids serait directement proportionel aux certitudes de ses clades représentées par les valeurs de "bootstrap".

Le couplage LCA défini pour un arbre d'espèces non-binaire [90] (voir la section 9.1) peut être utilisé dans notre arborescence d'arbres d'espèces. Il faudra voir si le coût (de duplication, de perte ou de mutation) induit par ce couplage a une monotonie croissante lors d'une exploration descendante de l'arborescence, propriété nécessaire au "Branch-and-Bound" (voir le chapitre 8). Il faudra ensuite comparer l'efficacité de notre approche, selon les temps de calcul et les coûts calculés, en utilisant soit nos bornes inférieures (c'est-à-dire les CLBs), soit les coûts induits par ce couplage LCA.

BIBLIOGRAPHIE

[1] O. AKERBORG, B. SENNBLAD, L. ARVESTAD et J. LAGERGREN : Simultaneous bayesian gene tree reconstruction and reconciliation analysis. *Proc. National Academy Sci. U.S.A.*, 106(14):5714–5719, 2009.

[2] S. F. ALTSCHUL, W. GISH, W. MILLER, E. W. MYERS et D. J. LIPMAN : Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, October 1990. ISSN 0022-2836.

[3] L. ARVESTAD, A. C. BERGLUND, J. LAGERGREN et B. SENNBLAD : Bayesian gene/species tree reconciliation and orthology analysis using MCMC. *Bioinformatics*, 19 Suppl 1:i7–15, 2003. ISSN 1367-4803.

[4] L. ARVESTAD, A.-C. BERGLUND, J. LAGERGREN et B. SENNBLAD : A probabilistic model for the reconciliation of gene trees and species trees. Rap. tech., TRITA-NA-0221, 2002.

[5] L. ARVESTAD, A.-C. BERGLUND, J. LAGERGREN et B. SENNBLAD : Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. *RECOMB 2004*, p. 326–335, 2004.

[6] L. ARVESTAD, J. LAGERGREN et B. SENNBLAD : The gene evolution model and computing its associated probabilities. *J. ACM*, 56(2):1–44, 2009.

[7] N. BAILEY : *The Elements of Stochastic Processes with Applications to the Natural Sciences*. Wiley-Interscience, 1990.

[8] M. BANSAL, J. BURLEIGH, O. EULENSTEIN et A. WEHE : Heuristics for the gene-duplication problem : A $\theta(n)$ speed-up for the local search. *RECOMB 2007*, p. 238–252, 2007.

[9] M. S. BANSAL et O. EULENSTEIN : The multiple gene duplication problem revisited. *Bioinformatics*, 24(13):i132–138, 2008.

[10] M. S. BANSAL et O. EULENSTEIN : An $\omega(n^2/\log n)$ speed-up of tbr heuristics for the gene-duplication problem. *IEEE/ACM Trans. Comput. Biol. and Bioinform.*, 5 (4):514–524, 2008.

[11] M. S. BANSAL, O. EULENSTEIN et A. WEHE : The gene-duplication problem : Near-linear time algorithms for NNI based local searches. *IEEE/ACM Trans. Comput. Biol. and Bioinform.*, 6(2):221–231, 2009.

[12] R. G. BEIKO, T. J. HARLOW et M. A. RAGAN : Highways of gene sharing in prokaryotes. *Proceedings of the National Academy of Sciences of the United States of America*, 102(40):14332–14337, 2005.

[13] A. BERGLUND-SONNHAMMER, P. STEFFANSSON, M. BETTS et D. LIBERLES : Optimal gene trees from sequences and species trees using a soft interpretation of parsimony. *J Mol Evol*, 63(2):240–250, Aug 2006.

[14] T. D. BIE, N. CRISTIANINI, J. P. DEMUTH et M. W. HAHN : Cafe : a computational tool for the study of gene family evolution. *Bioinformatics*, 22(10):1269–1271, 2006.

[15] O. R. P. BININDA-EMONDS : The evolution of supertrees. *Trends in Ecology & Evolution*, 19(6):315 – 322, 2004. ISSN 0169-5347.

[16] G. BLIN, C. CHAUVE, G. FERTIN, R. RIZZI et S. VIALETTE : Comparing genomes with duplications : A computational complexity point of view. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 4(4):523–534, 2007. ISSN 1545-5963.

[17] P. BONIZZONI, G. D. VEDOVA et R. DONDI : Reconciling a gene tree to a species tree under the duplication cost model. *Theor. Comput. Sci.*, 347(1-2):36–53, 2005. ISSN 0304-3975.

[18] J. CASTRESANA : Selection of Conserved Blocks from Multiple Alignments for Their Use in Phylogenetic Analysis. *Mol Biol Evol*, 17(4):540–552, 2000.

[19] C. CHAUVE et N. ELMABROUK : New perspectives on gene family evolution : Losses in reconciliation and a link with supertrees. *RECOMB 2009*, p. 46–58, 2009.

[20] C. CHAUVE, J.-P. DOYON et N. EL-MABROUK : Gene family evolution by duplication, speciation, and loss. *J. Comput. Biol.*, 15(8):1043–1062, 2008.

[21] K. CHEN, D. DURAND et M. FARACH-COLTON : NOTUNG : a program for dating gene duplications and optimizing gene family trees. *J Comput Biol*, 7(3-4):429–447, 2000.

[22] H. CHIPMAN, T. HASTIE et R. TIBSHIRANI : *Statistical Analysis of Gene Expression Microarray Data*. Boca Raton, FL : Chapman and Hall, 2003.

[23] M. CSŰRÖS et I. MIKLÓS : A probabilistic model for gene content evolution with duplication, loss, and horizontal transfer. *RECOMB 2006*, p. 206–220, 2006.

[24] M. CSŰRÖS et I. MIKLÓS : Mathematical Framework for Phylogenetic Birth-And-Death Models. *ArXiv e-prints*, fév. 2009.

[25] M. CSŰRÖS et I. MIKLÓS : Streamlining and Large Ancestral Genomes in Archaea Inferred with a Phylogenetic Birth-and-Death Model. *Mol Biol Evol*, 26(9):2087–2095, 2009.

[26] F. DELSUC, H. BRINKMANN et H. PHILIPPE : Phylogenomics and the reconstruction of the tree of life. *Nat Rev Genet*, 6(5):361–375, May 2005.

[27] J.-P. DOYON, C. CHAUVE et S. HAMEL : Space of gene/species trees reconciliations and parsimonious models. *J. Comput. Biol.*, 16(10):1399–1418, 2009.

[28] J.-P. DOYON, C. SCORNAVACCA, G. J. SZÖLLŐSI, V. RANWEZ et V. BERRY : An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications, and transfers, submitted.

[29] A. J. DRUMMOND, S. Y. W. HO, M. J. PHILLIPS et A. RAMBAUT : Relaxed phylogenetics and dating with confidence. *PLoS Biol*, 4(5):e88, 03 2006.

[30] O. EULENSTEIN : Vorhersage von genduplikationen und deren entwicklung in der evolution. 1998.

[31] O. EULENSTEIN : A linear time algorithm for tree mapping. *Arbeitspapiere der GMD No. 1046, St*, p. 1046, 1996.

[32] M. R. FELLOWS, M. T. HALLET et U. STEGE : On the multiple gene duplication problem. *ISAAC 1998*, p. 347–356, 1998.

[33] J. FELSENSTEIN : Evolutionary trees from dna sequences : a maximum likelihood approach. *Journal of molecular evolution*, 17(6):368–376, 1981. ISSN 0022-2844.

[34] G. FISCHER, E. P. C. ROCHA, F. BRUNET, M. VERGASSOLA et B. DUJON : Highly variable rates of genome rearrangements between hemiascomycetous yeast lineages. *PLoS Genet*, 2(3):e32, 03 2006.

[35] W. M. FITCH : Homology - a personal view on some of the problems. *Trends Genet.*, 16(5):227–231, 2000. ISSN 0168-9525.

[36] B. FREY et D. DUECK : Clustering by passing messages between data points. *Science (New York, NY)*, 315(5814):972–976, 2007.

[37] W. R. GILKS, S. RICHARDSON et D. SPIEGELHALTER : *Markov chain Monte Carlo in practice*. Chapman & Hall/CRC, 1996.

[38] M. GOODMAN, J. CZELUSNIAK, G. W. MOORE, R. A. HERRERA et G. MATSUDA : Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Syst. Zool.*, 28:132–163, 1979.

[39] P. GÓRECKI : Reconciliation problems for duplication, loss and horizontal gene transfer. *RECOMB 2004*, p. 316–325, 2004.

[40] P. GÓRECKI et J. TIURYN : DLS–trees : a model of evolutionary scenarios. *Theor. Comput. Sci.*, 359(1):378–399, 2006. ISSN 0304-3975.

[41] D. GRAUR et W.-H. LI : *Fundamentals of Molecular Evolution second edition*. Sinauer Associates, Sunderland, MA., 1999.

[42] S. GRIBALDO et C. BROCHIER : Phylogeny of prokaryotes : does it exist and why should we care ? *Research in Microbiology*, 160(7):513–521, September 2009. ISSN 09232508.

[43] R. GUIGO, I. MUCHNIK et T. F. SMITH : Reconstruction of ancient molecular phylogeny. *Mol Phylogenet Evol*, 6(2):189–213, Oct 1996.

[44] M. W. HAHN, M. V. HAN et S.-G. HAN : Gene family evolution across 12 drosophila genomes. *PLoS Genet.*, 3:e197, 11 2007.

[45] M. HALLETT, J. LAGERGREN et A. TOFIGH : Simultaneous identification of duplications and lateral transfers. *RECOMB 2004*, p. 347–356, 2004.

[46] M. HALLETT et J. LAGERGREN : New algorithms for the duplication-loss model. *RECOMB 2000*, p. 138–146, 2000.

[47] T. HASTIE, R. TIBSHIRANI et J. FRIEDMAN : The elements of statistical learning : Data mining, inference, and prediction. 2001.

[48] J. P. HUELSENBECK, F. RONQUIST, R. NIELSEN et J. P. BOLLBACK : Bayesian Inference of Phylogeny and Its Impact on Evolutionary Biology. *Science*, 294 (5550):2310–2314, 2001.

[49] M. HURLES : Gene duplication : the genomic trade in spare parts. *PLoS Biol*, 2 (7), July 2004. ISSN 1545-7885.

[50] O. JEFFROY, H. BRINKMANN, F. DELSUC et H. PHILIPPE : Phylogenomics : the beginning of incongruence ? *Trends Genet*, 22(4):225–31, Apr 2006.

[51] T. H. JUKES et C. R. CANTOR : *Evolution of Protein Molecules*. Academy Press, 1969.

[52] D. G. KENDALL : On the generalized "birth-and-death" process. *Ann. Math. Statistics*, 19:1–15, 1948.

[53] L. B. KOSKI et G. B. GOLDING : The closest blast hit is often not the nearest neighbor. *J Mol Evol*, 52(6):540–542, June 2001.

[54] S. KUMAR : Molecular clocks : four decades of evolution. *Nat Rev Genet*, 6 (8):654–662, 2005.

[55] C. LANAVE, G. PREPARATA et C. SACCONE : Mammalian genes as molecular clocks ? *J Mol Evol*, 21(4):346–350, 1984-1985.

[56] M. LARKIN, G. BLACKSHIELDS, N. BROWN, R. CHENNA, P. MCGETTIGAN, H. MCWILLIAM, F. VALENTIN, I. WALLACE, A. WILM, R. LOPEZ, J. THOMPSON, T. GIBSON et D. HIGGINS : Clustal W and Clustal X version 2.0. *Bioinformatics*, 23(21):2947–2948, 2007.

[57] H. LI et *et al.* : Treefam : a curated database of phylogenetic trees of animal gene families. *Nucleic Acids Research*, 34(Database-Issue):572–580, 2006.

[58] W.-H. LI : *Molecular Evolution*. Sinauer Associates, 1997.

[59] R. LIBESKIND-HADAS et M. A. CHARLESTON : On the computational complexity of the reticulate cophylogeny reconstruction problem. *J. Comput. Biol.*, 16(1):105–117, 2009.

[60] S. LLOYD : Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:128–137, 1982.

[61] P. LOPEZ, D. CASANE et H. PHILIPPE : Heterotachy, an Important Process of Protein Evolution. *Mol Biol Evol*, 19(1):1–7, 2002.

[62] M. LYNCH : Streamlining and simplification of microbial genome architecture. *Annual Review of Microbiology*, 60(1):327–349, 2006.

[63] B. MA, M. LI et L. ZHANG : From gene trees to species trees. *SIAM J. Comput.*, 30(3):729–752, 2001.

[64] J. MA, A. RATAN, L. ZHANG, W. MILLER et D. HAUSSLER : A heuristic algorithm for reconstructing ancestral gene orders with duplications. *RECOMB-CG*, p. 122–135, 2007.

[65] J. MACQUEEN : Some methods for classification and analysis of multivariate observations. *Procedings of the Fifth Berkeley Symposium on Math, Statistics, and Probability*, 1:281–297, 1967.

[66] W. P. MADDISON : Gene Trees in Species Trees. *Syst Biol*, 46(3):523–536, 1997.

[67] M. MAHAJAN, P. NIMBHORKAR et K. VARADARAJAN : The planar k-means problem is NP–hard. *WALCOM 2009*, p. 274–285, 2009.

[68] A. MCLYSAGHT, K. HOKAMP et K. WOLFE : Extensive genomic duplication during early chordate evolution. *Nat Genet*, 31(2):200–204, Jun 2002.

[69] D. MERKLE, M. MIDDENDORF et N. WIESEKE : A parameter-adaptive dynamic programming approach for inferring cophylogenies. *BMC Bioinformatics*, 11 (Suppl 1):S60, 2010. ISSN 1471-2105.

[70] A. S. NOVOZHILOV, G. P. KAREV et E. V. KOONIN : Biological applications of the theory of birth-and-death processes. *Brief Bioinform*, 7(1):70–85, 2006.

[71] R. D. PAGE : Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. *Syst. Biol.*, 43:58–77, 1994.

[72] R. D. PAGE et M. CHARLESTON : Trees within trees : Phylogeny and historical associations. *Trends in Ecology and Evolution*, 13:356–359, 1998.

[73] R. D. PAGE et J. A. COTTON : Vertebrate phylogenomics : reconciled trees and gene duplications. *Pac Symp Biocomput*, p. 536–47, 2002.

[74] R. PAGE : GeneTree : comparing gene and species phylogenies using reconciled trees. *Bioinformatics*, 14(9):819–820, 1998.

[75] P. PUIGBO, Y. WOLF et E. KOONIN : Search for a 'tree of life' in the thicket of the phylogenetic forest. *Journal of Biology*, 8(6):59, 2009. ISSN 1475-4924.

[76] F. RUSKEY : *Combinatorial Enumeration*. Book in preperation.

[77] A. RZHETSKY et M. NEI : Statistical properties of the ordinary least-squares, generalized least-squares, and minimum-evolution methods of phylogenetic inference. *Journal of Molecular Evolution*, 1992.

[78] N. SAITOU et M. NEI : The neighbor-joining method : a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, 4(4):406–425, 1987.

[79] M. SANDERSON et M. MCMMAHON : Inferring angiosperm phylogeny from est data with widespread gene duplication. *BMC Evolutionary Biology*, 7(Suppl 1), 2007. ISSN 1471-2148.

[80] B. SCHIEBER et U. VISHKIN : On finding lowest common ancestors : Simplification and parallelization. *SIAM J. Comput.*, 17(6):1253–1262, 1988.

[81] B. SENNBLAD et J. LAGERGREN : Probabilistic orthology analysis. *Syst. Biol.*, 58 (4):411–424, 2009.

[82] J. B. SLOWINSKI et R. D. PAGE : How should species phylogenies be inferred from sequence data ? *Biol*, 48:814–825, 1999.

[83] J. SPRING : Genome duplication strikes back. *Nat Genet*, 31(2):128–129, Jun 2002.

[84] U. STEGE : Gene trees and species trees : The gene-duplication problem is fixed-parameter tractable. *WADS 1999*, p. 166–183, 1999.

[85] D. L. SWOFFORD, G. J. OLSEN, P. J. WADDELL et D. M. HILLIS : Phylogenetic inference. *In D. M. Hillis, C. Moritz, and B. K. Mable, editors, Molecular systematics*.

[86] J. THORNE, H. KISHINO et I. PAINTER : Estimating the rate of evolution of the rate of molecular evolution. *Mol. Biol. Evol.*, 15:1647–1657, 1998.

[87] A. TOFIGH, M. HALLETT et J. LAGERGREN : Simultaneous identification of duplications and lateral gene transfers. *IEEE/ACM TCBB*.

[88] Y. Van de PEER : Are all fishes ancient polyploids ? *Journal of Structural and Functional Genomics*, 3:65–73(9), 2003.

[89] S. van DONGEN : Graph clustering by flow simulation. *PhD Thesis*, 2000.

[90] B. VERNOT, M. STOLZER, A. GOLDMAN et D. DURAND : Reconciliation with non-binary species trees. *J. Comput. Biol.*, 15(8):981–1006, 2008.

[91] J. VLASBLOM et S. WODAK : Markov clustering versus affinity propagation for the partitioning of protein interaction graphs. *BMC Bioinformatics*, 10(1):99, 2009.

[92] W. WANG, H. YU et M. LONG : Duplication-degeneration as a mechanism of gene fission and the origin of new genes in drosophila species. *Nat Genet*, 36(5):523–527, May 2004. ISSN 1061-4036.

[93] I. WAPINSKI, A. PFEFFER, N. FRIEDMAN et A. REGEV : Automatic genome-wide reconstruction of phylogenetic gene trees. *Bioinformatics*, 23(13):i549–558, 2007.

[94] A. WEHE, M. BANSAL, J. BURLEIGH et O. EULENSTEIN : DupTree : a program for large-scale phylogenetic analyses using gene tree parsimony. *Bioinformatics*, 24(13):1540–1541, 2008.

[95] Y. I. WOLF, I. B. ROGOZIN, N. V. GRISHIN et E. V. KOONIN : Genome trees and the tree of life. *Trends in Genetics*, 18(9):472 – 479, 2002. ISSN 0168-9525.

[96] J. ZHANG : Evolution by gene duplication : an update. *Trends in Ecology & Evolution*, 18(6):292–298, June 2003. ISSN 01695347.

[97] L. ZHANG : On a mirkin-muchnik-smith conjecture for comparing molecular phylogenies. *J Comput Biol*, 4(2):177–187, Sum 1997.

[98] C. M. ZMASEK et S. R. EDDY : A simple algorithm to infer gene duplication and speciation events on a gene tree. *Bioinformatics*, 17(9):821–828, Sep 2001.