Phylogeny of dependencies and dependencies of phylogenies in genes and genomes

Wandrille Duchemin

N° d'ordre NNT: 2017LYSE1263

THÈSE de DOCTORAT DE L'UNIVERSITÉ DE LYON

opérée au sein de
l'Université Claude Bernard Lyon 1

Ecole doctorale ED341
Évolution Écosystèmes Microbiologie Modélisation - E2M2

Soutenue publiquement le 04/12/2017, par :

Wandrille DUCHEMIN

Phylogénie des dépendances et dépendances des phylogénies dans les gènes et les génomes

Directeurs de thèse: Éric Tannier et Vincent Daubin

Devant le jury composé de :

Berry Vincent, Professeur des Universités, Montpellier	Rapporteur
Roest Crollius Hughes, Directeur de Recherche, ENS	Rapporteur
Brochier-Armanet Céline, Professeure des Universités, Lyon 1	Examinateur
Abby Sophie, Chargée de Recherche CNRS	Examinateur
Daubin Vincent, Directeur de Recherche CNRS	Directeur de thèse
Tannier Éric, Chargé de Recherche INRIA	Directeur de thèse

UNIVERSITE CLAUDE BERNARD - LYON 1

Président de l'Université	**M. le Professeur Frédéric FLEURY**
Président du Conseil Académique	M. le Professeur Hamda BEN HADID
Vice-président du Conseil d'Administration	M. le Professeur Didier REVEL
Vice-président du Conseil Formation et Vie Universitaire	M. le Professeur Philippe CHEVALIER
Vice-président de la Commission Recherche	M. Fabrice VALLÉE
Directeur Général des Services	M. Alain HELLEU

COMPOSANTES SANTE

Faculté de Médecine Lyon Est – Claude Bernard	Directeur : M. le Professeur J. ETIENNE
Faculté de Médecine et de Maïeutique Lyon Sud – Charles Mérieux	Directeur : Mme la Professeure C. BURILLON
Faculté d'Odontologie	Directeur : M. le Professeur D. BOURGEOIS
Institut des Sciences Pharmaceutiques et Biologiques	Directeur : Mme la Professeure C. VINCIGUERRA
Institut des Sciences et Techniques de la Réadaptation	Directeur : M. le Professeur Y. MATILLON
Département de formation et Centre de Recherche en Biologie Humaine	Directeur : Mme la Professeure A-M. SCHOTT

COMPOSANTES ET DEPARTEMENTS DE SCIENCES ET TECHNOLOGIE

Faculté des Sciences et Technologies	Directeur : M. F. DE MARCHI
Département Biologie	Directeur : M. le Professeur F. THEVENARD
Département Chimie Biochimie	Directeur : Mme C. FELIX
Département GEP	Directeur : M. Hassan HAMMOURI
Département Informatique	Directeur : M. le Professeur S. AKKOUCHE
Département Mathématiques	Directeur : M. le Professeur G. TOMANOV
Département Mécanique	Directeur : M. le Professeur H. BEN HADID
Département Physique	Directeur : M. le Professeur J-C PLENET
UFR Sciences et Techniques des Activités Physiques et Sportives	Directeur : M. Y.VANPOULLE
Observatoire des Sciences de l'Univers de Lyon	Directeur : M. B. GUIDERDONI
Polytech Lyon	Directeur : M. le Professeur E.PERRIN
Ecole Supérieure de Chimie Physique Electronique	Directeur : M. G. PIGNAULT
Institut Universitaire de Technologie de Lyon 1	Directeur : M. le Professeur C. VITON
Ecole Supérieure du Professorat et de l'Education	Directeur : M. le Professeur A. MOUGNIOTTE
Institut de Science Financière et d'Assurances	Directeur : M. N. LEBOISNE

Résumé

L'évolution moléculaire, basée sur l'étude des données de séquençage, s'est imposée comme une approche majeure pour l'étude de l'Histoire des organismes vivants (notamment à travers les arbres phylogénétiques). Ses méthodes classiques reposent sur un découpage des génomes en entités supposées indépendantes : les gènes.

Or, les gènes n'évoluent pas indépendamment : l'histoire d'un gène s'inscrit au sein de l'histoire des espèces qui le portent. En outre, leur position le long des chromosomes fait qu'ils partagent des événements de mutations structurales (duplications, pertes de fragments chromosomiques) avec les gènes proches. Enfin, leur potentielle fonction biologique les amène à être influencés par (et à influencer en retour) l'évolution d'autres gènes.

Je montre que ne pas prendre en compte ces relations d'inter-dépendances évolutives (de coévolution) lors de l'inférence d'arbres de gènes résulte en une surestimation des différences entre les arbres des différents gènes ainsi qu'entre les arbres des gènes et l'arbre des espèces.

Des modèles permettent déjà d'intégrer la coévolution des gènes avec les espèces à la reconstruction des arbres de gènes . Par ailleurs, on connaît des modèles décrivant l'évolution des relations entre gènes, néanmoins sans intégrer ces informations à la reconstruction des arbres de gènes. Je reprend ces avancées et les combine au sein d'une méthode qui modifie les arbres de gènes selon un critère qui prend en compte les séquences ainsi que des relations de coévolution avec les espèces et d'autres gènes.

Cette méthode, appliquée à des mammifères et des champignons, permet de produire des histoires de gènes cohérentes entre elles.

Mots clefs: évolution moléculaires, phylogénie, coévolution, adjacence

Phylogeny of dependencies and dependencies of phylogenies in genes and genomes

Abstract

Molecular evolution, based on the study of sequencing data, established itself as a fundamental approach in the study of the history of living organisms (noticeably through the inference of phylogenetic trees).

Classical molecular evolution methods rely on the decomposition of genomes into entities that are supposed independent: genes. However we know that genes do not evolve independently: their potential biological function lead them to be influenced by (and influence) the evolution of other genes. Moreover, their position along chromosomes imply that they share events of structural mutations (duplication, loss of a chromosome fragment) with neighbouring genes. Similarly, a gene individual history inscribes itself in the history of the species that bears it.

I show that not taking into account this inter-dependency relationships (co-evolutionary relationships) during the inference of gene trees results in an overestimation of the differences between gene trees as well as between gene tree and species tree.

Modelling efforts these last year have allowed the integration of gene and species co-evolution information to the reconstruction of gene trees. Besides, researchers have proposed models describing the evolution of the relationships linking genes, but without integration of this information in the tree building process.

My works aim to combine these advances in a method that modify gene trees according to a criterion that integrates sequence information and information coming from co-evolution relationships.

This method, applied to mammals and fungi, leads to gene histories that are more congruent (simpler adjacency histories, longer events of loss or transfer, ...).

Keywords: molecular evolution, phylogeny, co-evolution, adjacency

Résumé étendu

Dans un contexte biologique, le terme «évolution» fait référence au processus qui sous-tend la transmission avec variation des traits héritables d'une génération à l'autre. Cette transmission est dite «avec variation» parce qu'elle implique des modifications (appelées mutations) de certain traits transmis. Une autre facette importante de l'évolution est ce qu'on appelle la «sélection» ou «sélection naturelle» qui fait référence au fait que les traits gênant la reproduction des individus les portant auront tendance à voir leurs fréquences baisser au sein des populations (puisque leurs porteurs ont, en moyenne, moins de descendance). A l'inverse, les traits favorisant la reproduction de leur porteurs auront tendance a augmenter en fréquence au sein des populations : ils sont sélectionnés. Le jeu des mutations, qui produisent une variation au sein des entités biologiques, et de la sélection, qui favorise la partie de cette variation qui améliore les chances de reproduction des individus, est à l'origine de l'accumulation de différences entre les lignées au fil des générations qui resulte en la diversification des formes de vie sur Terre. Ainsi, deux traits dans deux espèces différentes peuvent avoir pour origine un même trait qui existait au sein de l'espèce ancêtre des deux espèces mentionnées. Des traits partageant un ancêtre commun sont dit *homologues*. Usuellement, le processus de diversification d'un trait donné au cours de son histoire évolutive est représenté sous la forme d'un arbre phylogénétique (où les branchements successifs symbolisent la séparation de deux lignées). La reconstruction d'arbres capturant l'histoire de traits biologiques est située au cœur de cette thèse, et en particulier la reconstruction d'arbres phylogénétiques à partir de données moléculaires. Je vais maintenant décrire les méthodes d'analyse qu'ont pourrait appeler «classiques» en phylogénie moléculaire parce que mes travaux s'appuient en grande partie sur leurs concepts et parce que je cherche à remettre en question certaines des hypothèses qui sont courantes dans ces méthodes.

Il a été établi que l'ADN [1] présent au cœur des cellules vivantes constitue un *marqueur évolutif* de choix [Zuckerkandl and Pauling, 1965]: il est le support de la transmission génétique d'une génération à l'autre et porte à la fois les traces de cette transmission, mais aussi des innovations (les mutations) qui l'accompagne. Ainsi, en comparant les séquences d'ADN (des séries de A, T, G et C) de différents organismes,

[1] Acide Desoxy-RiboNucléique qui forme de longues molécules, appelés chromosomes et composé d'un assemblage de nucléotides pouvant prendre quatre formes : l'adénine (A) , la thymine (T), la guanine (G) et la cytosine (C).

on peut établir leurs relations de parenté, comprendre lesquels sont les plus proches les uns des autres ainsi que le nombre de mutations qui les séparent. Néanmoins, l'étude des méchanismes de l'évolution du matériel génétique a révélé que comparer les séquences d'ADN des organismes n'est pas un problème aisé, en grande partie parce que les séquences ne subissent pas seulement des mutations de contenu (un nucléotide est remplacé par un autre), mais aussi des mutations structurelles (des morceaux de séquence se déplacent, sont supprimés, dupliqués ou ajoutés).

Avant d'espérer reconstruire une histoire des espèces vivantes à partir de leur séquences d'ADN, il faut d'abord établir quels sont les morceaux de leurs séquences qui sont homologues entre eux. Ces relations d'homologies sont généralement basées sur un principe de similarité : on suppose que les séquences dérivant d'un ancêtre commun seront plus semblables entre elles qu'avec d'autres séquences. Cela implique aussi que les séquences détectées comme homologues n'auront pas subi de grandes mutations structurelles durant le laps de temps qui les sépare dans l'évolution (sans quoi on n'aurait pas pu établir l'homologie de ces séquences).

Dans un contexte d'étude de l'évolution, on appelle ces fragments d'ADN pour lesquels on peut établir des relations d'homologie des *gènes* et on nomme un groupe de gènes homologues entre eux une famille de gènes. Ainsi, le gène a fonction d'unité d'évolution : à chaque famille de gène correspond une histoire évolutive qui lui est propre, un arbre de gène.

Une fois des familles de gènes définies, il faut *aligner* les différentes séquences homologues, c'est à dire établir l'homologie non pas séquence à séquence, mais nucléotide à nucléotide. C'est une étape cruciale car d'elle dépendent directement les inférences des évènements de mutations qui serviront à reconstruire les phylogénies. Pour autant, aligner plusieurs séquences constitue un problème difficile (d'un point de vue de la complexité algorithmique) et en pratique on aligne les séquences à l'aide d'heuristiques, ce qui implique que les résultats puissent ne pas correspondre à nos attentes de qualité et impliquer des biais et/ou des incertitudes dans les phylogénies.

L'inférence d'une phylogénie à partir d'un alignement est, de nos jours, effectuée à partir de calculs de vraisemblance d'un modèle probabiliste d'évolution d'un nucléotide. Chaque position de l'alignement est alors considérée indépendamment de ses voisines, ce qui renforce l'importance de la qualité de l'alignement et amène à faire, au moins de manière implicite, l'hypothèse que les nucléotides d'un même gène évoluent indépendamment les uns des autres.

Or, cette hypothèse, si elle est utile pour inférer une phylogénie dans des temps de calcul raisonnables, ne permet pas d'accéder aux relations qu'entretiennent les différentes entités biologiques entre elles. En effet on observe que les forces de sélection qui s'exercent sur les nucléotides (ou d'autres entités biologiques) dépendent de l'état d'autres nucléotides (par exemple dans les codons la mutation d'un nucléotide en un autre n'aura pas le même impact, et ne sera donc pas sélectionné de la même manière, si l'état des deux autres nucléotides du codon implique que cette mutation entraînera un changement de l'acide aminé codé ou non).

On désigne par le terme de co-évolution ce concept selon lequel les contraintes évolutives s'exerçant sur une entité biologique dépendent de l'état (ou de la dynamique) d'autres traits. La co-évolution peut être observée à toutes les échelles du vivant: entre écosystèmes (par exemple, entre l'écosystème d'un lac et celui des terres qui l'entoure), entre espèces (par exemple, les proies et les prédateurs), entre individus et à l'échelle moléculaire entre protéines, gènes et nucléotides ou acides aminés (j'ai déjà évoqué l'exemple des codons). Mes travaux se focalisent sur l'unité d'évolution que forme le gène et je définis trois formes de coévolution qu'un gène peut entretenir avec d'autres entités.

La première forme de coévolution est celle qui unit le gène aux nucléotides de son alignement. On peut considérer chaque nucléotide comme une entité propre (une forme de mini-gène) qui coévolue avec ses nucléotides voisins de manière si forte qu'ils peuvent être représentés dans la même phylogénie. Ainsi je représente ces relations de «cophylogénie » entre les nucléotides comme une relation hiérarchique de coévolution entre le nucléotide (contenu) et le gène (contenant). Si l'on considère l'inférence de phylogénie, chaque nucléotide contient un peu d'information concernant son histoire, mais généralement pas assez pour la reconstruire entièrement. Mais parce qu'ils coévoluent, l'information des autres nucléotides d'un même gène nous informent sur l'histoire du nucléotide et à plus forte raison, du gène qui les contient tous.

La deuxième forme de coévolution est celle qui lie le gène aux espèces dans lesquelles il évolue. Ainsi lorsque les espèces se diversifient (spéciation) les gènes qu'elles contiennent subissent ces évènements aussi. A l'inverse, les gènes subissent des événements évolutifs qui leur sont propres et qui causent des différences entre leurs phylogénies et celle des espèces. Ainsi la duplication de gène provoque une augmentation du nombre de copies d'un gène au sein du même génome. A l'inverse, la perte de gène correspond à une diminution du nombre de copies d'un gène dans un génome. Le tri de lignée incomplet résulte de la diversité génétique de populations subissant des spéciations rapprochées dans le temps et amène les lignées d'un gène à se diversifier dans un ordre différent de celui des espèces. Enfin, le transfert horizontal correspond à l'intégration au sein d'un génome de matériel génétique venant d'un autre organisme, parfois très éloigné phylogénétiquement parlant. C'est une importante source d'innovation pour les organismes (en particulier les bactéries) parce qu'il permet l'acquisition soudaine de nouvelles fonctions. On appelle *réconciliation* le fait d'inférer, au sein d'un arbre de gène, quelles parties correspondent à des spéciations ou à d'autres événements évolutifs (on parle alors d'un arbre de gène réconcilié).

La troisième forme de coévolution est celle qui intervient entre gènes. Ainsi, les gènes dont la fonction est liée subissent une pression évolutive commune pour le maintien de cette fonction (ou exercent une pression l'un sur l'autre): ils coévoluent. De même, les gènes qui sont physiquement proches le long des chromosomes ont de plus fortes chances de partager un événement évolutif tel qu'une duplication, une perte ou un transfert (parce que techniquement ces événements arrivent à des fragments chromosomiques et pas à des gènes).

Ces trois formes de coévolution permettent d'établir un cadre dans lequel penser aux contraintes au sein desquelles les gènes évoluent et à l'information qui est à notre disposition pour reconstruire les phylogénies des gènes. L'inférence d'arbre «classique» présentée précédemment peut être vue comme n'exploitant que la première forme de coévolution. D'autres méthodes développées plus récemment exploitent aussi la seconde forme de co-évolution, cherchant une phylogénie qui implique une réconciliation vraisemblable (étant donné un arbre d'espèce et un modèle de réconciliation). En particulier, certaines méthodes sont en mesure d'exploiter l'information de la première et de la deuxième forme de coévolution conjointement, résolvant les incertitudes de l'une grâce à l'autre. Mes travaux visent à intégrer à la reconstruction des histoires des gènes de l'information venant de chacune des trois formes de coévolution, ce qu'aucune méthode ne fait à ce jour. En effet, ne pas prendre en compte la troisième forme de coévolution rend aveugle aux larges mutations structurelles (ainsi, on voit plusieurs duplications de gène indépendantes là où il n'y en a qu'une seule de grande taille) et amène à sous-estimer l'importance de la congruence entre les gènes liés (physiquement ou fonctionnellement).

Afin d'illustrer l'importance de la prise en compte des liens unissant les gènes lors de la reconstruction de l'histoire d'un groupe de gènes (et a fortiori d'un génome entier) j'expose les travaux que j'ai effectués dans le cadre de la reconstruction de l'histoire de ces liens entre gènes et les problèmes qu'on observe lorsque les histoires des gènes ont été inférées indépendamment les unes des autres. Je désigne comme *adjacence* un lien entre deux gènes. Les adjacences peuvent représenter diverses formes de lien entre gène (lien physique, fonctionnel, ...). A l'image des gènes, les adjacences ont leurs propres histoires et peuvent ainsi être transmises, créées (on parle de gain d'adjacence) ou perdues (cassure d'adjacence)[2].

Des méthodes existent déjà pour inférer des histoires d'adjacences étant données les adjacences entres gènes actuels et les arbres réconciliés de ces gènes. En particulier je m'intéresse à l'algorithme DeCo [Bérard *et al.*, 2012] qui infère les histoires d'adjacences qui minimisent un score basé sur le nombre de gains et de cassures. Depuis la publication originale en 2012, de nombreuses extensions ont été publiées pour cet algorithme, mais à chaque fois de manière indépendante. Ainsi une publication a permis de prendre en compte les réconciliations incluant des transferts horizontaux [Patterson *et al.*, 2013], une autre intègre l'inférence de nouvelles adjacences actuelles [Anselmetti *et al.*, 2015] et encore une autre permet l'échantillonage de solutions non parsimonieuses [Chauve *et al.*, 2015], mais aucune implémentation ne permet de faire les trois à la fois. Une partie significative de mes travaux a été d'intégrer les différentes extensions de l'algorithme DeCo au sein d'une implémentation unique, modulaire et cohérente[3]. Cette implémentation a donné lieu à une généralisation des formules de récursion servant au calcul du score des histoires d'adjacences afin de prendre en compte tout les cas provenant des interactions entre les différentes extensions et a aussi été l'occasion d'ajouter plusieurs nuovelles extensions et options à l'algorithme. Le programme obtenu, nommé DeCoSTAR, a donné lieu à une publication dans une revue scientifique, incluse dans ce document[4].

[2] Notons toutefois qu'une adjacence ne peut être présente que si les deux gènes qu'elle relie sont présents.

[3] Une version probabiliste de l'algorithme DeCo existe aussi [Semeria *et al.*, 2015]. Toutefois je ne l'ai pas intégrée avec les autres à cause de la différence entre méthodes de parcimonie et méthodes probabilistes d'une part, et à cause des ses limitations méthodologiques (comme l'impossibilité d'avoir plus d'une duplication à la suite par espèce par exemple) et computationelles (notamment lors du traitement des longueurs de branches) d'autre part.

[4] Tout le code produit a été intégré au sein de la suite de programmes ecceTERA [Jacox *et al.*, 2016] qui permet l'inférence d'arbres de gènes en optimisant un critère prenant en compte de l'information de la première et seconde forme de coévolution.

L'algorithme DeCo permet d'inférer les histoires des adjacences, et donc les adjacences ancestrales. Ainsi, si l'on considère les relations de voisinage le long d'un chromosome comme adjacences on peut avoir accès à l'ordre ancestral des gènes. C'est l'approche que j'ai mise en place au sein d'une étude ayant pour but la reconstruction de la séquence des chromosomes ancestraux d'une souche de la bactérie *Yersinia pestis* (en effet, pour avoir accès à la séquence ancestrale il me fallait d'abord avoir accès à l'ordre des gènes ancestraux). Cette étude a aussi fait l'objet d'une publication scientifique incluse dans ce document. Dans cet article, on met en évidence que le fait que les arbres et réconciliations des gènes aient été faits indépendamment les uns des autres mène à des inférences d'adjacences ancestrales erronées, en particulier l'inférence de chromosomes ancestraux non linéaires. On développe aussi l'idée que la linéarité des chromosomes ancestraux inférés peut être utilisée comme un critère pour corriger les arbres de gènes à l'origine de ces conflits.

Cette idée est intéressante car elle représente un cas où l'on utilise un critère issu de la troisième forme de coévolution (la linéarité des chromosomes ancestraux, qui dépend des adjacences) dans la reconstruction (ou plutôt ici la correction) des histoires individuelles des gènes. Les gènes dont l'histoire est ainsi corrigée se retrouvent avec des arbres et des réconciliations *localement* moins vraisemblable (dans le sens de critères qui leur sont propres comme leur alignement ou le nombre d'événements évolutif dans leur réconciliation), mais qui sont *globalement* plus vraisemblable (dans le sens d'un critère prenant en compte tout le génome, ici la linéarité des chromosomes ancestraux).

En développant cette idée d'un critère global (à l'échelle du génome) allant plus loin que la somme d'un ensemble de critères locaux (à l'échelle du gène), j'ai développé un score qui regroupe un ensemble de familles de gènes et permet de les évaluer sur la base d'information provenant des trois formes de coévolution. Ce score prend des valeurs d'autant plus petites que les histoires des familles de gènes sont en adéquation avec les données dont on dispose. Il prend la forme:

Score Global = *topologie* + *réconciliation* + *adjacence* + *co-événement*

La première partie du score, *topologie*, correspond à l'information de la première forme de coévolution et évalue l'adéquation de chaque arbre de gène à son alignement. En pratique, je n'utilise pas directement l'alignement des gènes, mais plutôt une distribution a posteriori d'arbres qui me permet d'estimer avec précision la vraisemblance d'un arbre en utilisant une méthode similaire à celle utilisée dans l'algorithme TERA [Scornavacca *et al.*, 2014].

La seconde partie du score, *réconciliation*, est égale à la somme des coûts des événements de duplication, perte et transfert de gènes observés dans les arbres réconciliés de chaque famille de gène. Ainsi elle correspond à l'information venant de la deuxième forme de coévolution.

La troisième et la quatrième partie du score sont des critères reposant sur la troisième forme de coévolution. *adjacence* correspond à la somme des coûts des histoires d'adjacences (inférées avec DeCoSTAR) entre les gènes considérés en termes de gains et de cassures d'adjacences. Ainsi des gènes voisins ayant des histoires compatibles entre elles (c'est à dire impliquant moins de gains et de cassures d'adjacences) sont favorisés. Enfin, *co-événement* est une mesure qui prend en compte le fait qu'on considère chaque événement de gène (par exemple une duplication de gène) indépendamment dans chaque arbre réconcilié alors qu'il est possible que les événements de deux arbres réconciliés dont les gènes seraient voisins correspondent en fait à un seul événement (par exemple, la duplication d'un large fragment chromosomique est observée sous la forme d'une duplication pour chacun des gènes impliqués). Les résultats de DeCoSTAR permettent de détecter les cas où des événements observés dans deux arbres réconciliés correspondent en fait à un événement partagé. En utilisant ces résultats je détecte des co-événements et je suis en mesure de corriger le fait qu'on les compte séparément lorsqu'on considère les familles de gènes indépendamment les unes des autres (comme c'est le cas dans *réconciliation*).

Ayant défini un score global pour un ensemble de familles de gènes reliées par des adjacences actuelles, je décris une stratégie pour trouver l'ensemble de topologies et d'arbres réconciliés (un par famille de gène) qui minimise le score global. Étant donné la taille de l'espace des solutions à explorer (la combinaison de tout les arbres et de toutes les réconciliations possibles pour chaque famille de gènes), une exploration exhaustive ne paraît pas envisageable en pratique. J'ai choisi une méthode qui, à partir d'une solution initiale (c'est-à-dire, une instance où chaque famille de gène a son arbre réconcilié et où les histoires d'adjacences ont été calculées), effectue des mouvements locaux pour optimiser graduellement le score (à la manière d'un échantillonneur de Gibbs). Le mouvement local correspond à la proposition d'un nouvel arbre réconcilié pour une famille de gènes (toutes les autres étant fixes par ailleurs). A ce nouvel arbre réconcilié correspond un changement dans les histoires des adjacences qui le lient aux autres familles, et donc aussi un changement dans les co-événements impliquant des gènes de cette famille. Ainsi, le nouvel arbre réconcilié implique un changement dans le score global. Si ce changement correspond à une diminution, alors on accepte le nouvel arbre réconcilié. Sinon on rejette le nouvel arbre et on garde l'ancien (ou alternativement, on accepte le nouvel arbre avec une probabilité qui dépend de l'augmentation du score, à la manière d'un recuit simulé et afin de permettre d'échapper à un optimum local). En enchaînant ainsi des mouvements locaux (à l'échelle d'une famille de gènes), on améliore petit à petit le score global afin de le faire tendre vers un optimum. Un programme permettant de calculer et d'optimiser le score global a été développé à partir du code de DeCoSTAR intégré à ecceTERA qui constitue alors pour moi une bibliothèque utile pour la manipulation des arbres, des réconciliations et des histoires d'adjacences. J'ai développé plusieurs méthodes permettant de faire une nouvelle proposition d'arbre réconcilié pour une famille de gènes (c'est à dire proposer un mouvement local).

La première consiste à choisir un arbre aléatoire pour la famille de gène, puis à le réconcilier de manière parcimonieuse (à l'aide de TERA). Cette méthode n'exploite aucune information de coévolution dans le choix du nouvel arbre.

Le deuxième méthode utilise l'information venant de la première forme de coévolution. Elle revient à échantillonner un arbre au sein de la représentation qu'on a de la distribution d'arbres a posteriori de la famille de gènes (qui elle-même représente le support venant de l'alignement). Comme précédemment, l'arbre échantillonné est ensuite réconcilié à l'aide de TERA.

La troisième méthode va un peu plus loin en intégrant de l'information de la première et de la deuxième forme de co-évolution. Elle s'inspire des formules de récurrence de TERA mais dépasse le cadre initial de calcul d'une solution parcimonieuse en y appliquant une transformation algébrique qui, après quelques ajustements (notamment pour garantir que la complexité de l'algorithme reste la même), permet d'échantillonner des arbres de gènes réconciliés avec une probabilité inversement proportionnelle à leur score joint *topologie* + *réconciliation* (pour reprendre les termes du score global). Le fait d'être en mesure de dépasser la parcimonie de TERA nous permet de proposer des solutions localement suboptimales, mais globalement optimales (en d'autre terme, la perte de vraisemblance de l'arbre ou de la réconciliation peut être compensée par des histoires d'adjacences plus parcimonieuses ou plus de co-événements).

Enfin, la quatrième méthode développée utilise de l'information venant des trois formes de coévolution. Pour ce faire, je pars de la troisième méthode proposée et j'y ajoute la présence d'un *arbre réconcilié guide* (l'arbre réconcilié d'un gène avec lequel la famille pour laquelle on fait une proposition coévolue) avec lequel il est possible de former des co-événements à un coût inférieur à ceux de l'événement indépendant. De cette manière, j'obtiens une méthode de réconciliation et de choix de topologie qui favorise les solutions qui contiennent des co-événements avec un autre arbre réconcilié. Ces deux dernières méthodes, qui correspondent à des nouveautés algorithmiques, ont été implémentées dans un programme à part entière.

La stratégie d'optimisation du score global a été appliquée à des jeux de données de mammifères (absence de transferts horizontaux) et de champignons (présence de transferts horizontaux). Ces applications permettent dans un premier temps de comparer les différentes méthodes de propositions développées, mais aussi d'observer les modifications qu'entraîne l'optimisation du score global sur les arbres réconciliés des gènes et les histoire des adjacences et enfin elles servent à obtenir des élements de réponse sur la viabilité des adjacences en tant que vecteur de signal évolutif et aussi sur la taille des larges événements évolutif (duplications, pertes et transferts)

J'observe tout d'abord que les méthodes que j'ai développé sont bien à même de trouver des arbres de gènes et des réconciliations qui font diminuer le score global. Cette amélioration concerne en particulier les parties du score ayant trait à la co-évolution entre les gènes (adjacences et co-événements), ce qui signifie qu'avec ces méthodes on augmente bien la congruence entre histoires de gènes. En outre, j'observe que l'amélioration du score global est liée à une augmentation de la taille (en nombre de gènes) des événements de duplication, perte et transfert (il s'agit donc ici des co-événements). Cela confirme mon idée selon laquelle en effectuant l'inférence des histoires de gène indépendamment on sous-évalue la taille de ces événements et il est donc particulièrement intéressant que ma méthode permette d'obtenir un point de vue alternatif qui prend en compte des attentes de co-évolution entre les gènes. Dans mes jeux de données, j'observe qu'en moyenne les pertes sont plus grandes que les autres événements (duplications ou transfert horizontaux). Dans le jeu de données de champignons, les transferts horizontaux sont détectés comme étant à la fois plus longs et plus nombreux que les duplications (à noter que les transferts ne sont vus comme plus long que les duplications qu'après l'optimisation du score global).

Dans le cas des jeux de données de mammifères, cette amélioration s'accompagne aussi d'une amélioration de mesures de linéarité des génomes ancestraux (dans le cadre de l'inférence de l'ordre des gènes ancestraux, la linéarité est un critère de qualité) ainsi que d'une augmentation du nombre de nouvelles adjacences extantes retrouvées (en effet le jeu de données de mammifère inclut des espèces dont le génome est mal assemblé et pour lesquelles il est souhaitable d'inférer de nouvelles adjacences reliant les contigs existants). Ces observations ne sont pas faites sur le jeu de données de champignons où je remarque plutôt une diminution globale du nombre de co-événements de réconciliations, en particulier les pertes ainsi qu'une légère augmentation du nombre d'événements de gains et de cassures d'adjacences.

En effet chez les champignons je considère une plus grande profondeur évolutive (l'arbre d'espèces utilisé pour les champignons correspond à un temps de divergence environ 5 fois plus long que celui couvert par l'arbre d'espèce des mammifères): les génomes de champignons ont donc eu plus de temps pour accumuler des réarrangements et les adjacences entre gènes apparaissent comme moins conservées (parce que les réarrangement modifient les adjacences). Ceci implique que chez les champignons les adjacences contiennent moins de signal nous renseignant sur l'histoire des gènes (ou du moins un signal qui ne va pas aussi profondément dans l'arbre d'espèce) que dans le jeux de données de mammifères. Cela peut expliquer la plus grande importance donnée à la parcimonie des réconciliations par rapport à celle des adjacences lors de l'optimisation des arbres de gènes des champignons.

En somme, j'ai décrit trois formes de coévolutions s'appliquant au gène: la coévolution avec sa séquence, avec les espèces et avec les autres gènes. Après avoir exploré les problèmes venant de ce qu'on ne prend en compte que la première et/ou la deuxième forme de coévolution lors de la reconstruction des histoires (phylogénie et réconciliation) des gènes, j'ai écrit un score global qui permet d'évaluer des histoires d'un ensemble de familles de gènes selon des critères propres à chacunes des trois formes de coévolution. J'ai aussi développé une méthode, basée sur des mouvements locaux (à l'échelle d'une seule famille), pour optimiser le score global, ainsi que plusieurs algorithmes pour effectuer les nouvelles propositions à la base des mouvements locaux. Enfin, j'ai montré à travers des applications à des jeux de données de mammifères et de champignons que cette méthode d'optimisation du score global peux effectivement amener des changements dans les arbres de gènes qui favorisent de plus long événements evolutifs ainsi que des génomes ancestraux plus lineaires.

Remerciements

Merci

C'est tout?
Oui!
Ah ! Non ! C'est un peu court, jeune homme !
On pouvait dire... oh ! Dieu ! ... bien des choses en somme...
En variant la cible, —par exemple, tenez :

Aux jury : Merci à eux d'avoir accepté de lire et d'évaluer mon manuscrit; à Sophie Abby, Céline Brochier-Armanet et en particulier à mes relecteurs Hugues Roest Crollius et Vincent Berry.

Aux directeurs : Merci à Vincent Daubin Éric Tannier de m'avoir encadré avec bienveillance durant ces trois années (et demi pour Eric si je compte mon stage). J'ai énormément apprécié travailler à vos cotés, d'avoir pu à la fois profiter de la confiance et de l'autonomie que vous m'avez accordées, mais aussi de votre supervision et du temps que vous m'avez consacrés à chaque fois que j'en ais eu besoin. J'espère avoir la chance de continuer à collaborer avec vous dans les années à venir.

Aux thésards (et ex-thésards) qui m'ont accompagnés durant ces 3 ans. Merci à Héloise , Fanny , Magali , Yoann , Adrian , Frédéric , Cécile , Laura , Anne , Pierre , Monique , Etienne , Maud (et j'en oublie bien d'autres, certainement) , ça a été un honneur et un plaisir de vivre cette expérience en même temps que vous. Félicitations à ceux qui ont déjà soutenu, bon courage à ceux pour qui la date de la soutenance approche. Mention spéciale à Fanny pour son appareil à gaufres qui a bien servi (et qui sert encore), À Héloïse, super co-bureau qui m'a évité de nombreux maux de têtes en m'expliquant les procédures administratives, À Adrian pour les soirées films, GoT et les nombreuses discussions et projets de ré-invention de la roue qu'on a eu, À Etienne, pour l'amitié qu'il me porte et les innombrables discussions et projets que nous eûmes, avons et aurons.

Aux membres du LBBE et en particulier ceux de mon équipe. J'ai eu la chance de profiter d'une ambiance de travail idéale, avec des collègues tout plus sympathiques les uns que les autres. Un remerciement particulier à Simon, Damien, Bastien, Vincent, Arnaud, Laurent, Christelle, Daniel et Blerina qui m'ont aidé dans ma recherche et mon enseignement. Je pense aussi aux pauses café quotidiennes et aux nombreuses discussions, souvent décousues, qu'on a eu avec Murray, Thomas, Jos, Dominique, pour n'en citer que quelques-un. Mention spéciale à Adil, collègue de bureau aux conversations intéressantes et avec lequel j'espère lancer de nombreux dés.

Aux coureurs du LBBE : Philippe, Simon, Murray, Guillaume, Rémi, ça a été un plaisir d'user mes semelles avec vous.

À la *team* LBBEer : c'était certainement l'expérience la plus proche de la biologie de paillasse que j'ai eu durant ma thèse. Merci aux courageux qui nous ont aidés à mettre en bouteille. Merci à ceux qui nous ont aidé à déguster. Et bon courage à la nouvelle génération de brasseurs.

Aux chercheurs en dehors du LBBE qui m'ont accompagnés dans ma recherche, merci à vous. En particulier Yann, Céline et Gergely.

Aux kiwis : un merci tout particulier à Murray et Pierre-Yves qui m'ont donné ma chance et m'ont fait confiance pour mon premier contact concret avec le monde de la recherche scientifique.

À Andrey et Grégory dont les incessantes rotations m'ont soutenues pendant la rédaction de ce document.

Et enfin, à la famille. À Marie-Odile, Guillaume et Louise pour avoir fait de moi ce que je suis et m'avoir supporté toute ma vie (enfin, presque, pour Louise). À Marie-Alix, dont l'affection et le soutien ont été essentiels ces dernières années.

Et à ceux que, sans aucun doute, j'oublie : merci.

Abbreviations

AA : Amino Acid

CCP : Conditional Clade Probabilities

DL : Duplication and Loss

DNA : DeoxyriboNucleic Acid

DTL : Duplication, Transfer and Loss

HGT : Horizontal Gene Transfer

ILS : Incomplete Lineage Sorting

LCA : Lowest Common Ancestor

ML : Maximum Likelihood

i.e. : *id est* , "that is"

e.g. : *exempli gratia* , "for example"

resp. : respectively

Contents

Chapter 1

Introduction : Molecular evolution and co-evolution

1.1 Molecular evolution

In a biological context, evolution refers to the processes underlying the transmission with variation of heritable traits of biological entities (species, population, gene) from generation to generation.

The transmission of heritable traits is said to be "with variation" because it can involve modifications (called mutations), making some traits of the descendant different from the ones of its ancestors.

The accumulation of these modifications over generations in different lineages is at the origin of the diversity of life on Earth. Thus, traits (*e.g.*, teeth) present in two different biological entities (*e.g.*, two species like human and mouse) may be different (*e.g.*, human teeth and mouse teeth differ) but nonetheless results from the transmission with mutation of the same, ancestral trait (*e.g.*, the teeth of the common ancestor of human and mouse). Traits sharing a common ancestor are said to be *homologous*.

The diversification pattern of homologous traits from a single common ancestor to its current biodiversity is commonly represented through a *phylogenetic tree*[1]

[1] section 1.1.3 contains many details about trees and the various technical terms surrounding them.

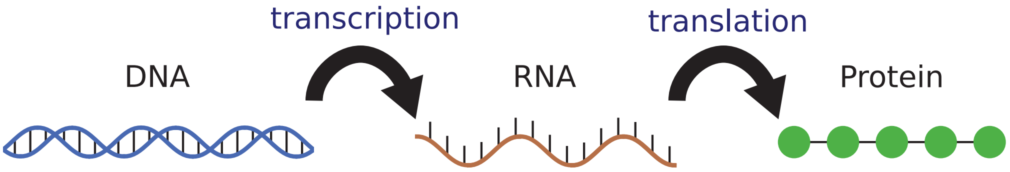


Figure 1.1: DNA is transcribed into RNA, which is translated into proteins.

1.1.1 Molecules as evolutionary markers

Situated at the heart of living cells, deoxyribonucleic acid (DNA) molecules encode the information necessary to the expression of the various organism traits: the genome.

DNA is composed of two strands coiled around each-other to form a double helix [Watson, J. D.; Crick, 1953]. Each strand is a chain of nucleotides, which are themselves composed of three elements: a phosphate group, a five carbon sugar and a nitrogenous base. These nitrogenous bases can be of four type in nucleic DNA (adenine (A), thymine (T), guanine (G) and cytosine (C)) and the sequence formed by the succession of these four bases is generally held to be the genetic information.

In the scientific community, DNA molecules are usually considered to be the medium by whom this genetic information is passed from one generation to another. Zuckerkandl and Pauling [1965] are usually credited with the establishement of DNA as *"documents of evolutionary history"* (or *semantides*), meaning that they carry information establishing the relatedness of the different organisms (*i.e.*, the similarities between the genome of different species) but they also carry the trace of the evolutionary processes (*i.e.*, the differences between the genome of different species).

Also mentioned by Zuckerkandl and Pauling [1965] as semantides, proteins, or polypeptides are chains of amino-acids and are synthesized in the cell by using some portion of DNA as a template [2] (this is called *translation*, see Figure 1.1), followed by subsequent transformations (*post-translational modification*) and have various roles in cell and organism physiology, from forming the cytoskeleton (the complex network of filaments that help govern cell shape), regulating cross-membrane transport (allowing various ions or bigger molecules such as glucose to enter or exit the cell), help synthesise various metabolites through enzymatic ability (such as the degra-

[2] the DNA must first by transcribed into ribonucleic acid (RNA)

dation of starch into simpler sugars by amylases), or regulate metabolism (such as insuline, an hormone implied in glycemy regulation).

Their functional role make them interesting markers to study, because tracking their evolution can help understand the apparition of numerous functions in living organisms (such as photosynthesis [Tavano and Donohue, 2006]). As such, polypeptides can and are also considered as molecular evolutionary marker. However, as they are not the primary carrier of the genetic information but are merely derived from it and can subsequently be modified, they carry an evolutionary information different from the one contained into their DNA counterpart. Moreover, the non-translated fraction of the genome contains information about the evolutionary history of genomes.

Changes in the DNA sequence, the basis of the signal of evolutionary history, can occur in a variety of ways, such as a modification during the replication or reparation of DNA. Independent from their origin, mutations can be of at least two types: substitutions, which correspond to changes in the *value* of a nucleotide in the sequence, and structural changes, which correspond to changes in the organization of the nucleotides between them.

Substitutions correspond to the replacement of a single nucleotide by another in the sequence. Figure 1.2 illustrates some scenarios of nucleotide substitution as well as establishes some vocabulary for the description of these scenarios: *single substitution* and *multiple substitutions* are quite self-explanatory; a *back substitution* correspond to a case where the scenario of substitution lead back to the original nucleotide (from a Guanine to a Guanine in the figure) ; *convergent substitutions* describe the phenomenon that occurs when two sequences evolving independently undergo scenarios of substitution that lead to similar differences with the original sequence (`s1` in the figure)[3].

Some structural mutations are illustrated in Figure 1.3 (these correspond to the ones that are most often modelled in evolutionary biology). Insertions relate to the addition of nucleotides in a sequence. They can range in size from a single nucleotide to whole chromosome regions. When the insertion is small, it is likely the result of an erroneous addition during DNA replication, but when the insertion is large the inserted region may come from another region of the genome, like another

[3] Note that this vocabulary is applied to substitutions here, but is also adapted where speaking about mutations in general.

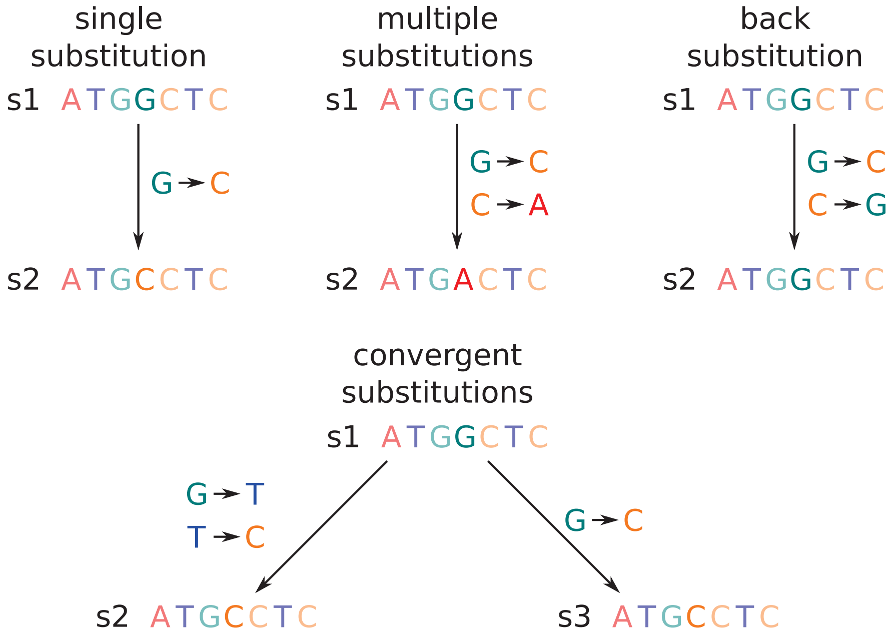


Figure 1.2: Different scenarios of substitutions in the evolution of the sequence `s1` into sequence `s2`; or into the parallel evolution of `s1` into sequences `s2` and `s3` (bottom row). Here, the mutations are always occurring on the same (highlighted) position (a Guanine in `s1`).

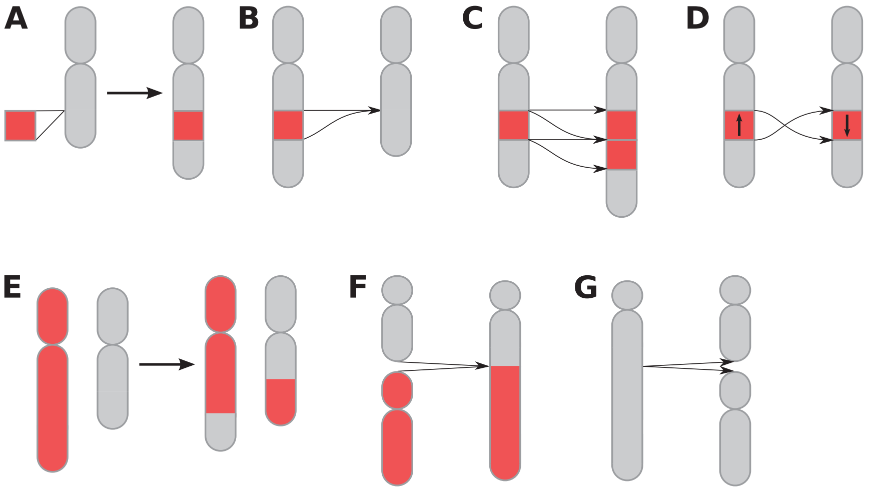


Figure 1.3: **A** Insertion of a region of the red chromosome in the gray one. **B** Deletion of the red chromosome region. **C** Duplication of the red chromosome region. **D** Inversion of the red chromosome region. **E** Translocation between the red and gray chromosomes. **F** Fusion of the red and gray chromosomes. **G** Fission of the gray chromosome.

chromosome. Deletions are about the disappearance of nucleotides from a sequence, and may also range in size from a single nucleotide to a whole chromosome region. They can be seen as complementary to the insertion, and insertions and deletions are often referred to together as *indels*. Examples of small indels are shown in Figure 1.4.

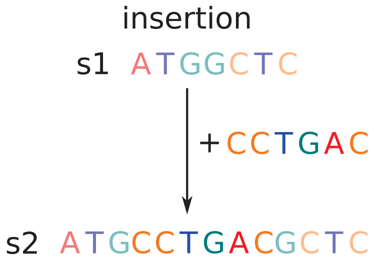


Figure 1.4: Insertion and deletion in nucleotides sequences.

Duplication is a mutation that copies a chromosome portion and inserts it elsewhere in the genome. It can also occur at all sorts of range, including the whole genome (and a cell replication may be seen as a duplication of the whole genome followed by the segregation of each copy in a different cell). Inversions consist in the reversal of a chromosome portion. Translocation is a process where two chromosomes exchange fragments. Finally, entire chromosomes may fusion or fission together.

A mutation occurring in the genome of a given individual in a population creates a new locus, or a new variant of a pre-existing locus in the population (a locus is a position on a chromosome). In either case this result is called an *allele*[4]. An allele begins by being present in only one genome/individual (the one where it occurred) and may, through reproduction, see the number of individual bearing it in the population vary[5].

A given allele may then be categorized in terms of its effect on the reproductive success of the organism that bears it. If the allele increases the reproductive success, it is said to be *beneficial*. On the contrary, an allele that decreases reproductive success (for instance, by causing a grave illness preventing the survival of the individual bearing it) is said to be *deleterious*. Finally, an allele that has no effect (or

[4] It follows that a given locus may have one or several alleles (*i.e.*, variants).

[5] In some multicellular organisms (such as animals), a mutation may only be passed to the next generation if it occurred in the germline of the organism (*i.e.*, a cell that will participate in reproduction).

no significant effect) on reproductive fitness is said to be *neutral*.

Beneficial alleles will tend to become more frequent in the population (*i.e.*, their bearers will on average reproduce more than others in the population), whereas *deleterious* alleles will tend to see their frequency decrease in the population, a process known as natural selection.

However this process of allele frequency variation is inherently stochastic and is thus subject to random variation, which is referred to as *genetic drift*. The strength of the genetic drift is usually related to the *effective population size*[6] : a high effective population size will mean a lower genetic drift, and vice-versa. The more genetic drift there is, the more a beneficial or a deleterious allele will be subject to random noise and the more the evolution of their frequency in the population will come close to that of a neutral allele.

1.1.2 Classical molecular phylogeny: how to get a phylogenetic tree

The following sections describe a classical pipeline of phylogenetic tree inference. It is not an exhaustive presentation of all possible methods or each step, but rather focuses on the most used ones, the most representatives ones.

NB: for vocabulary purpose, I presume that the sequence studied are DNA sequences. Similar observations can be done when using other sequences, such as polypeptides.

Homology detection

To unravel and understand the processes that lead to the current diversity of an ensemble of biological sequences from a ancestral single ancestor, one must first establish that these sequences have a common ancestor. That is, one must first determine homology at the sequence level (an example of homologous sequences is given in Figure 1.5).

At the scale of genomes, the question of homology determination becomes about splitting the genomes sequences in groups (families) of homologous sub-sequences.

[6] The effective population size is the size that an idealised (*i.e.*, a population that corresponds to the simplifying assumptions made in population genetic models, such as random mating or constant size) would need to have to represent the genetic diversity of the real population.

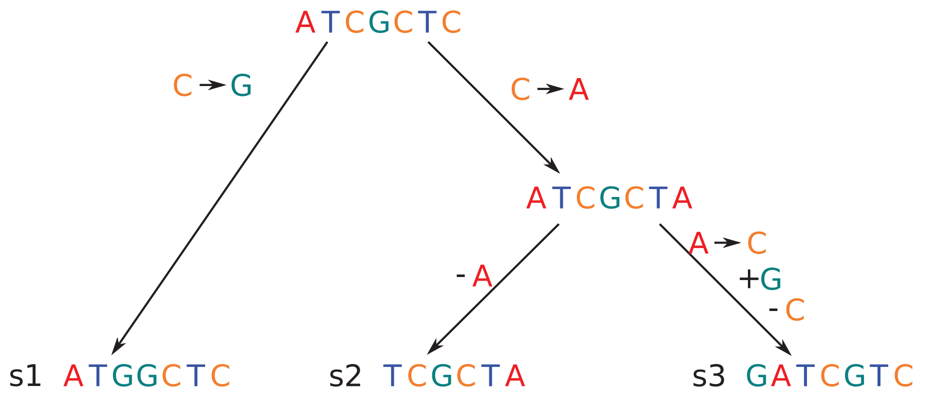


Figure 1.5: Three sequences (s1, s2 and s3) have evolved from the same, ancestral sequence: they are *homologous*.

In the absence of structural mutations, defining homology would just come down to grouping together chromosomes of different species. However in practice the relative positions of two nucleotides (or groups of nucleotides) can vary throughout evolution. One or the other (or both) could be moved to another position or chromosome, or be lost in a deletion.

So when looking for homologous sequences, one simultaneously looks for contiguous blocks of nucleotides that share a common evolutionary history. These contiguous blocks of nucleotides can be seen as forming a cohesive unit from the point of view of evolution and I call them *genes*[7] . Following this notation, I call a family of homologous sequences a *gene family*.

Detecting gene families is, in itself, a hard problem. It is done by comparing different sequences together under the hypothesis that if two sequences have, indeed, a common ancestor, then they should exhibit more similarity between them than with other sequences (that they are not homologous with).

Similarity between two sequences can be assessed with local alignment tools, such as BLAST [Altschul *et al.*, 1990] which provides a score of similarity significance (the e-value). Two sequences that share enough similarity (using a e-value cut-off for instance) over their whole lengths may be considered *putative homologs*. These can be converted into pairwise relationships between sequences, and these can, in turn, be used to cluster together groups of putative homologs (so that *in fine*, two sequences detected as putative homologs because of similarity may not be considered

[7] Not to be confused with other function-based definitions like the protein-coding gene or expressed gene. Here I use this term in its evolutionary definition.

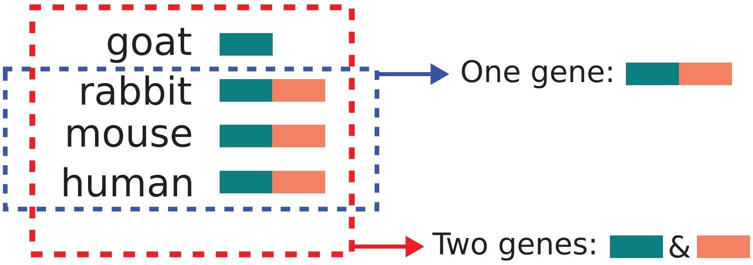


Figure 1.6: Two blocks of homologous nucleotides (in teal and salmon color) are neighbours in `human`, `mouse` and `rabbit`, but the `goat` only shows the teal block. A study including only `human`, `mouse` and `rabbit` (blue frame) will see the whole "teal and salmon" block as a single gene. In contrast, a study including all four species (red frame) will distinguish the whole teal gene and salmon gene.

as homologs, and *vice versa*). Going beyond pairwise similarity scores, more recent methods directly compare a single sequence to a group of sequences. Other methods combine the results of several approaches to refine homology search. See Fuellen [2008], Fujimoto *et al.* [2016] , Overbeek *et al.* [1999], for examples of homology detection methods and their relative strengths and advantages.

Note that homology relationships are transitive (if a is homologous to b and c, then b and c are homologous) and symmetric (if a is homologous to b, then b is homologous to c), so that defining homology between couples of sequences is enough to determine groups of homologous sequences.

Similarity based approaches make the implicit hypothesis that the sequences should not have diverged so much that they now share no more similarity than any two random (non-homologous) sequences. For this reason, detecting homology between sequences separated by a long evolutionary time can be a challenging task.

In the case of protein-coding sequences, amino-acid sequences may be preferred to nucleotide sequences for deep homology detection. This is due to the *genetic code degeneracy*[8]: a nucleotide substitution might not result in a change in the corresponding amino-acid, so the amino-acid sequences of two diverging genes may be more similar together than their nucleotide sequences.

An important point to make is that what I define as a gene (again, from an

[8] this refers to the fact that protein coding nucleotide sequences are organised in triplets (called codons). Each triplet correspond to an amino-acid, but there is more codons than amino acids (64 possible triplets, 20 amino-acids) so different codons can encode the same amino-acid. Furthermore, codons encoding the same amino-acid are often quite similar (they often share there first and second nucleotides) so that a nucleotide substitution might not result in a change in the corresponding amino-acid sequence.

evolutionary perspective) depends on the context of the study. As in Figure 1.6, what constitutes one gene when considering some species may constitute several when considering others. In general, the bigger the scope of a study in time is, the more genes will tend to reduce in size. To understand this, consider a block of contiguous nucleotides: the longer we let it evolve, the higher the chance that a structural mutation will break up this contiguous block into several, smaller, blocks.

However, despite the idea that structural mutations renders complex and contextual the detection of homologous families, many studies still defines their evolutionary units using predetermined arbitrary objects, in particular protein coding genes (this is the case in the HOGENOM data-base [Penel *et al.*, 2009] for instance). Such an approach causes problems, as these do not necessarily form good evolutionary units and any error made during the homology detection will have repercussions during the subsequent analyses. For instance, coding genes are known to sometimes fusion or fission. Consider the case where two genes (A and B) are found in a fusionned form AB in some organisms. Using coding gene as the unit of evolution means that 3 different families (A, B and AB) will be considered and then treated independently, completely obscuring the relationship between A (resp. B) and AB and leading to false inferences of the rates of gene apparition / disappearance or ancestral gene content. Similar problems occur because many proteins have been identified to be modular [Moore *et al.*, 2013]: they are composed of several blocks that rearrange themselves throughout evolution, in these cases, an approach finding the smallest blocks of homologous characters (as illustrated in Figure 1.6) is better adapted, such as the approach used to build the ProDom data-base [Kahn *et al.*, 2008].

Sequence alignment

The previous step detected homology between sequence fragments, sequence alignment aims to establish homology at the nucleotide (or amino-acid) scale. Here a given position in an amino-acid or nucleotide sequence is commonly called a *site*.

Among a gene family (*i.e.*, a set of homologous sequences), the different sequences may not have the same size (*e.g.*, the same number of sites) and in the alignment problem it is presumed to be because of structural mutations such as insertion/deletions (indels). A multiple sequence alignment is formed by the ensemble of its sequences where special characters have been inserted to symbolize indels

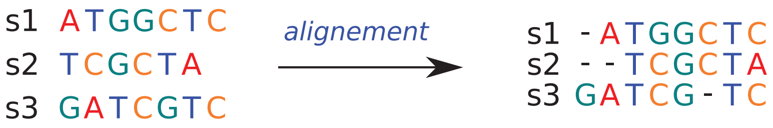


Figure 1.7: Three homologous sequences (s1, s2 and s3) and their alignment on the right. Gaps are figured with the - symbol.

(called *gaps* in this context) so that the different sequences have the same size and homologous nucleotides have the same position in the completed sequence (*i.e.*, the sequence with its gaps). An example of such an alignment is shown in Figure 1.7. The ensemble of the sites at the same position across the sequences of a multiple sequence alignment is commonly referred to as a column of the multiple sequence alignment. Consequently, building a multiple sequence alignment comes down to placing *gaps* in the different sequences it is composed of. Note that this definition of alignement inference leaves out all the other forms of structural mutations (such as duplication) which are not modelled in an intra-gene context, and which may lead to erroneous alignments.

Henceforward, I interchangeably use the terms "multiple sequence alignment" and "alignment".

This gap placement can be achieved through the optimisation of an arbitrary score that takes into account matches (declaring as homologous two sites bearing the same nucleotide / amino-acid), mismatches (declaring as homologous two sites bearing different nucleotides / amino-acids), gap opening (*i.e.*, the first position of a gap), and gap extension (the other positions). Exact algorithms exist to infer the alignments corresponding to the optimal score for two sequences (such as the Needleman-Wunsch algorithm [Needleman and Wunsch, 1970]). However for more than two sequences (and the vast majority of gene families have more than two) such exact approaches are intractable and thus methods to build a multiple sequence alignment rely on heuristics.

Aside from the fact that an heuristic solution means that the optimal alignment may never be reached (or at least never with certitude), the optimized metric is arbitrary and may not be realistic for the data under study. Indeed, remember that the scores that these algorithms optimize only take into account a predetermined number of events (substitution and small indels) and do not model a lot of more complex ones (like small inversions for instance), at the risk of missing them. Depending on the specificity of the dataset (such as the number of sequences, their size or similarity), different alignment algorithms, will be more or less suited to build a correct multiple sequence alignment [Pais *et al.*, 2014]. Additionally, a number of the parameters underlying alignment bear an evolutionary meaning and their value constitutes, in itself, an hypothesis on the evolutionary processes that shaped the data (for instance, the costs associated to a given type of mismatch compared to another). Because of this one should exercise caution toward a multiple sequence alignment and the extent to which it represents the real homology between sites.

Moreover, many multiple sequence alignment methods (see Hogeweg and Hesper [1984]; Edgar [2004]; Löytynoja and Goldman [2005] for instance) involve, at some point in their algorithm, a tree describing the relations between the sequences to guide the alignment. This tree, by necessity, is often itself built using very crude method as a proper phylogenetic tree reconstruction would necessitate an alignment (so that there is a form of co-dependency between alignment and tree reconstruction). As, in the context of phylogenetic reconstruction, the alignment is critical to the reconstruction of the phylogenetic tree (see next section), the idea that the alignment was built using a crudely constructed tree as a guide is yet another reason to be critical of the alignment and the phylogenetic tree it leads to.

Phylogenetic tree

This step aims to reconstruct the history of diversification of the sequences of a gene family (the gene family (bifurcating) tree), from the multiple sequence alignment of this gene family. For definitions of the tree-related vocabulary, see the section about phylogenetic tree jargon.

While there exists many methods to reconstruct a tree from a multiple sequence alignment, most currently used ones rely on a probabilistic model of sequence evolution. These methods can be categorized in two broad groups: maximum likelihood (ML) and Bayesian [9].

[9] There also exists distance and maximum parsimony approaches, however they are considered to be less statistically reliable than probabilistic methods (the probabilistic approaches are more computationally intensive than the distance approaches but this barrier has lowered with the advent of better computers and algorithms). Distance based approaches rely on a measure of *genetic distance* between the sequences of the alignment. These distances, summarized in a genetic distance matrix, are then used to build the phylogenetic tree. This is done either with a hierarchical clustering algorithm (see UPGMA [Sokal and Michener, 1958] and Neighbor-Joining [Saitou and Nei, 1987]) or by finding the tree which minimizes the distance between the *tree distance matrix* (where the distance is given by the sum of the lengths of the branches separating two leaves) and the

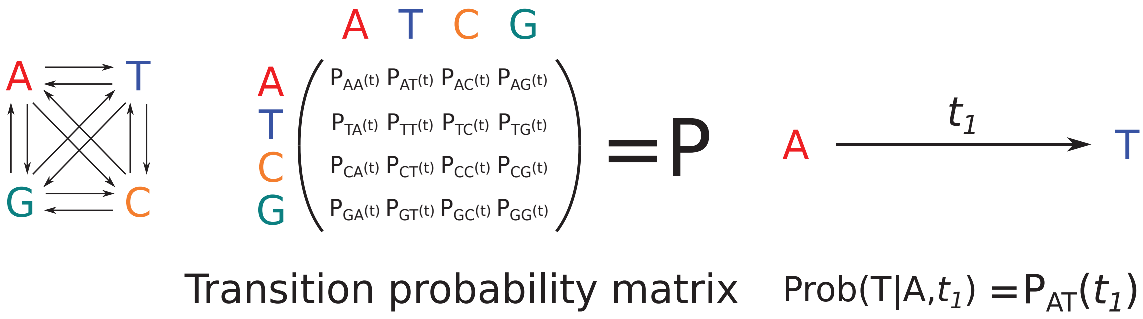

A
T
G
C
A T C G
A
T
C
G
P_AA(t) P_AT(t) P_AC(t) P_AG(t)
P_TA(t) P_TT(t) P_TC(t) P_TG(t)
P_CA(t) P_CT(t) P_CC(t) P_CG(t)
P_GA(t) P_GT(t) P_GC(t) P_GG(t)
=P
A
t1
T
Transition probability matrix
Prob(T|A,t1) =P_AT(t1)

Figure 1.8: A continuous Markov nucleotide substitution model. The probability, given the nucleotide `A` evolving for time t_1 to obtain `T`, can be computed with the matrix $P(t)$.

Probabilistic model of sequence evolution. A probabilistic model of sequence evolution, or more precisely in our case a *model of nucleotide substitution* describes a stochastic process of substitution for a single nucleotide (equivalent models exist for amino-acids). This process is usually represented as a Markov process [10] where the states are the different nucleotides, as shown in Figure 1.8 where in the substitution rate matrix P the value $P_{xy}(t)$ represents the transition probability from state x to state y in a given amount of time t.

Computing the likelihood. Consider an alignment D as well as a model M which includes the substitution model, tree topology, branch lengths (and any additional parameter).

To evaluate the fit of the model to the alignment we would like to access its probability given the data: $p(M|D)$.

Using Bayes theorem we get:

$$p(M|D) = p(D|M) \times \frac{p(M)}{p(D)}$$

Assuming that all instances of the model (*i.e.*, all different tree topologies, branch lengths and substitution model parameters) are *a priori* equally likely and that the probability of observing the data does not change with the model, $\frac{p(M)}{p(D)}$ is constant

input distance matrix [Fitch and Margoliash, 1967]. Maximum parsimony approaches try to obtain a tree that requires the minimal number of evolutionary events (in this context, a substitution or an indel) to explain the current alignment [Fitch, 1971]. Other, related, methods consider instead a linear combination of the different evolutionary events (*i.e.*, making some substitutions more costly than other) as the metric to optimize.

[10] For an introduction to Markov processes, see: Markov process visual interactive examples [last accessed 02-June-2017]. Note however that these models are discrete in time, while the one presented here is continuous in time.

across all instances of M[11] and that in turn we can write:

$$p(M|D) \ \propto \ p(D|M)$$

(where $\propto$ means "proportional to").

The probability of the model according to the data is proportional to the probability to observe the data according to the model.

This second probability is called the *likelihood* of the model and is noted $L(M) = p(D|M)$.

As $L(M) \ \propto \ p(M|D)$, an instance of M with a higher likelihood than another instance also has a higher $p(M|D)$ and so is a better fit to the data.

If we make the hypothesis that each alignment column is independent from the others we get:

$$L(M) = \prod_{a=1}^{m} p(D^a|M)$$

where m is the alignment size, D^a is the *a-th* column of the alignment. $p(D^a|M)$ is the likelihood of the model for the column D^a and is computed with:

$$p(D^a|M) = \sum_{x_0} \pi_{x_0} L_0(x_0)$$

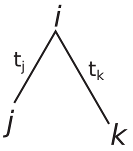
where, for a node i, $L_i(x_i)$ is the probability of observing data at the leaves of the subtree rooted in i, given that the nucleotide at i is x_i; 0 is the root of the tree; and π_{x_0} is the probability to observe nucleotide x_0 at the root of the tree[12].

If node i is a leaf:

$$\begin{aligned} L_i(x_i) &= 1 \text{ if } x_i \text{ is the observed nucleotide} \\ &= 0 \text{ otherwise} \end{aligned}$$

Otherwise:

$$L_i(x_i) = \Big[\sum_{x_j} P_{x_i x_j}(t_j) L_j(x_j)\Big] \times \Big[\sum_{x_k} P_{x_i x_k}(t_k) L_j(x_k)\Big]$$



where node i is the parent of nodes j and k, with associated branch lengths t_j and t_k (as in the diagram on the right).

Note that the computation of the likelihood essentially takes the form of a recursion on the tree topology[Felsenstein, 1981].

Note also that this model is applied to each column of the alignment indepen-

[11] Note that in the context of Bayesian methods this may not be the case and we have to account for *a priori* distributions on the model parameters, including topologies.

[12] These probabilities are additional parameters of the model of evolution that may also be inferred or fixed externally.

dently, each contributing independently to the overall likelihood. This is causing branch lengths in a phylogenetic tree to correspond to an *expected number of substitution per site* (however, tree branch lengths may also represent time in the case of *ultrametric trees*). This is also due to computational issues: considering each column of the alignment independently means that the complexity of the likelihood computing algorithm will (at worst) be proportional to the number of columns in the alignment (not doing so would mean considering the whole sequence at once, which corresponds to a state space that grows with the nuceotide alphabet size raised to power of the number of columns in the alignment). This however implies that when reconstructing the gene tree we have to make the hypotheses that 1) each branch of the tree evolves independently from the others 2) each position in the alignment evolves independently from the others (an hypothesis which is implicit to the model of *single* nucleotide substitution). As such these models cannot capture information about the interactions between nucleotides or gene lineages, despite the fact that these interactions are of particular interest, especially in a biological context where entities (be they species, genes or nucleotides) do not evolve in a vacuum, but in interaction with other biological entities (these ideas about interactions in evolution will be further developed in the next section).

Using the likelihood to evaluate candidate trees, the tree reconstruction algorithm must then explore the space of tree topologies[13] (and other evolutionary parameters, when applicable). If the number of leaves is low enough, this exploration can be exhaustive. However it quickly becomes intractable to do so and the algorithm must rely on heuristics to explore the tree space.

Maximum likelihood and Bayesian approaches both compute the likelihood as shown above, but maximum likelihood methods try to find the tree (and model parameters) with the maximum likelihood (as their name indicates) while Bayesian ones build a *posterior distribution* of trees (and models parameters): a distribution of trees where the frequency of an observed tree is proportional to its likelihood according to the alignment (and its *prior probability*).

The whole process used for phylogenetic tree inference is prone to error, may they come from the previous steps (homology detection and alignment), from the heuristics used for the tree space exploration, or the hypotheses of the chosen model

[13] Again, see the section about phylogenetic tree jargon for a definition of the number of possible topologies with a given number of leaves.

of evolution. As such, any phylogenetic tree obtained may be viewed as a single estimation of the history of a gene family, and some measure of the reliability of this estimation is desirable. Several methods have been developed to evaluate the certainty with whom a single tree represent the history of its alignment, or at least the robustness of the inference. This evaluation most often comes in the form of a *support value* associated with each clade or branch of the tree, representing the levels of confidence on different parts of the reconstructed phylogeny.

One of the methods to do so is the *bootstrap* method [Felsenstein, 1985] (although it is more an evaluation of the method robustness than certainty). It starts with the constitution of several *bootstrap samples* by choosing (with replacement) columns in the alignment (each bootstrap sample is an alignment the same size as the original one). A *bootstrap tree* is then estimated from each bootstrap sample, using the same method as the one used for the obtention of the original tree (the one inferred with the original alignment). The support value associated to a given branch (viewed are as a bipartition of the leaves of the tree) of the original tree corresponds to the number of times this branch was observed in the bootstrap trees. It ranges between 0 (the worst case, where this branch/bipartition was not seen in any of the generated bootstrap trees) and the number of generated bootstrap sample (where this branch was seen in all the bootstrap trees)[14].

Another procedure to assign support values to an estimated tree is to evaluate whether or not an internal branch length is significantly different from 0 (as, if it was 0 then it would be equivalent to removing the branch which would result in a multifurcating tree). For likelihood-based methods, this can be achieved via the procedure known as *likelihood ratio test* [Felsenstein, 1988].

Other methods can be used, such as the Shimodaira-Hasegawa test [Shimodaira and Hasegawa, 1999] which compares the likelihood of a set of trees.

Finally, as mentioned before, the Bayesian methods do not infer a single phylogenetic tree but a distribution of trees. A way to extract support value from such a distribution would be, for each possible bipartition, to assess the number of times it is observed in the distribution and divide it by the number of trees in the distribution [Larget, 1999]. This measure is called the *posterior clade probability*.

[14] Branches that separates a single leaf from the rest of the tree are present in every topology and their bootstrap value is thus always maximal. In practice, it is omitted.

1.1.3 Trees and tree jargon

This document contains numerous references to phylogenetic trees or object derived from them. Phylogenetic trees (henceforth referred to simply as trees) help, among other, define evolutionary relationship between different entities, can be used to determine times of divergence between them or characterise past, ancestral, entities. This section aims to describe the common vocabulary surrounding trees.

In phylogeny, a tree is often described as a set of *nodes*, linked together by *branches*, such that

- there is always a path between any two nodes (*i.e.*, they are connected together by one or several branches).
- there is only one path between any two nodes (*i.e.*, there is no cycles).

The nodes of a tree are usually divided into two sets: *leaves* and *internal nodes*. A *leaf* is a node of the tree that has only one neighbour (*i.e.*, is linked by only one branch). An *internal node* is a node that is not a leaf (*i.e.*, it has more than one neighbour).

Figure 1.9 shows two trees. Let us concentrate on the top tree first. It has 10 nodes (a, b, c, d, e, f, g, h, i and j). 6 of these are leaves (a, b, c, d, e and f) and 4 are internal nodes (g, h, i, j).

The top tree is said to be *unrooted*, which means that it has no *root* and cannot be used directly to determine if a node represent an ancestor or a descendant of another node.

By opposition, a *rooted* tree is a tree that possesses a *root*, which is a node that is determined to be the ancestor of all the other nodes in the tree. The bottom tree of Figure 1.9 is a rooted tree. Compared to the top tree, it has an additional node k that is defined as the root of the tree. The presence of the root orients the tree (the idea is that ancestry *flows* from the root to the leaves) and allows us to determine which node is an ancestor to which. Here, the leaf a is said to be the *child* of the internal node g. g, in turn, is the *parent* of a. b, who is also a child of g, is the *sister* of a.

We define the *ancestors* of a node as the set of nodes composed of its parent and the ancestors of its parent (*i.e.*, the parent of its parent, and so on, and so forth, up to the root). Here the ancestors of a are g, i and k.

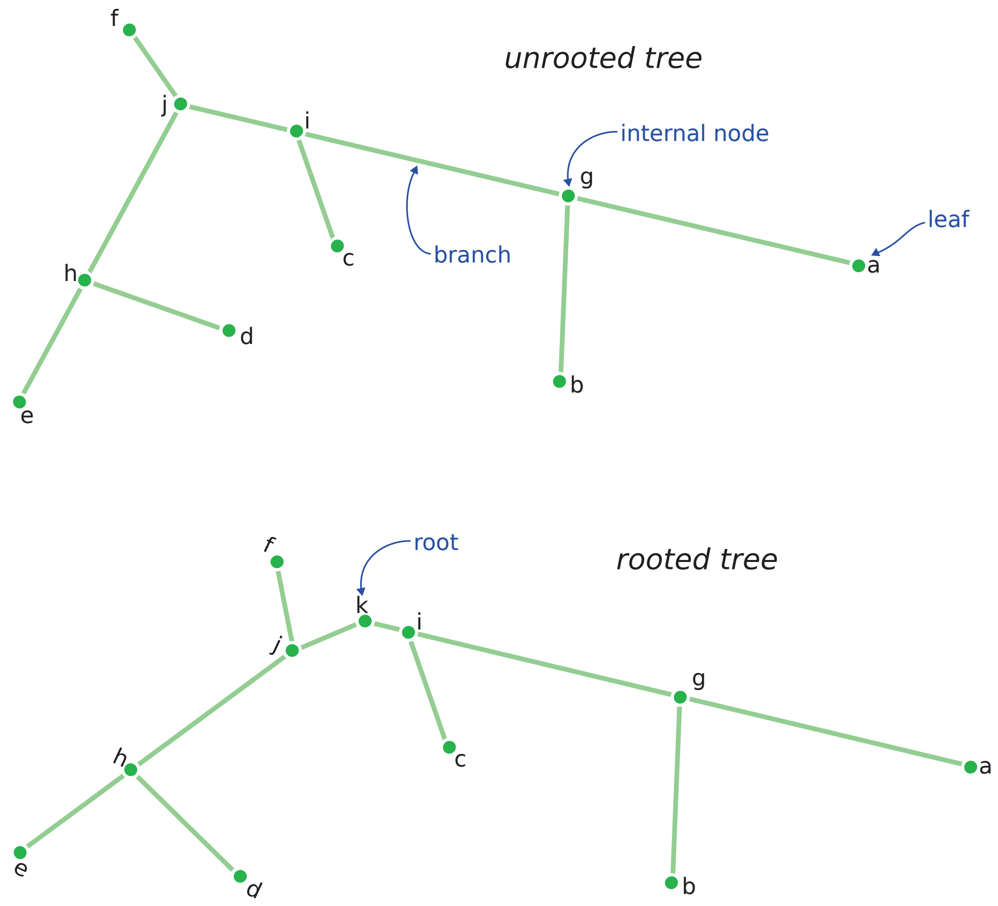


Figure 1.9: Two phylogenetic trees, one unrooted, the other rooted.

Conversely, the *descendants* of a node is composed of its children and their descendants. The descendants of node j are f, h, e, d.

The *Lowest Common Ancestor* (abbreviated *LCA*) of a set of nodes N is the node that is an ancestor to all node in N and is farthest from the root of the tree. The LCA of a and b is g. The LCA of c and a is i. The LCA of e, d and f is j.

The part of the tree composed of a given node n and all its descendants is called the *subtree* rooted at n.

Any branch in an unrooted tree can correspond to a *bipartition* of the leaves of the tree. For instance in the unrooted tree of Figure 1.9, the branch linking i and g corresponds to the bipartition $\{a, b \mid c, d, e, f\}$. I call a set of leaves a *clade* and say that it is present in an unrooted tree if there exist a bipartition in it that separates the clades from the rest of the leaves of the tree. In the unrooted tree of Figure 1.9, the clade $\{a, b\}$ is present in the tree, as well as $\{f, e, d\}$ or $\{c, b, a\}$. However the

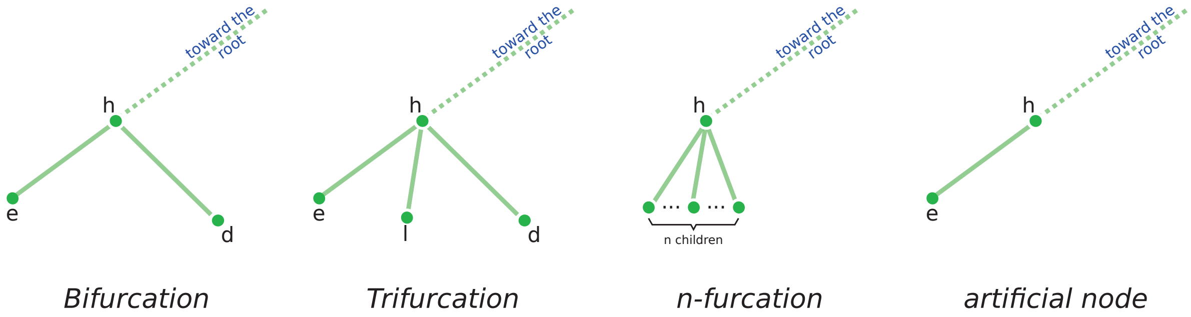


Figure 1.10: The node h is, from left to right : a bifurcation, two multifurcations and an artificial node.

clade $\{a, c\}$ is not present in the tree (because of the presence of leaf b, there is no branch that separates $\{a, c\}$ from all the other leaves).

For any clade C in a tree, there is a complementary clade that correspond to the leaves of the tree that are not in C (in other words, the other side of the bipartition). In Figure 1.9 the complementary clade of $\{a, b\}$ is $\{c, d, e, f\}$.

The two trees of Figure 1.9 are said to be *bifurcating* because their internal nodes are linked by exactly three branches (or two, in the case of the root). The internal nodes are said to be *bifurcations* in the tree. The term bifurcation refers to the idea that, in the rooted tree, any internal node has two branches linking to its children (the third branch links to its parent when it is not the root). When a node has more than two children, it is said to be *multifurcating*. A node with exactly 3 children is called a *trifurcation*; a node with n children is called a *n-furcation*, or a multifurcation of size n. Additionally, I call a node with only one child an *artificial node*, for reasons explained later. Figure 1.10 illustrates these concepts. Phylogenetic methods often (but not always) assume a bifurcating tree.

The different branches of a tree are often associated with *lengths* that describe a distance between the linked nodes. Additionally, the distance between two nodes that are not linked by a branch is the sum of the distances of the branches separating them. In the context of molecular evolution, this distance can take two forms: it may correspond to an amount of time or it may correspond to an *amount of evolution* which is often expressed in terms of substitutions per site (as shown in Figure 1.11). These concepts are illustrated in Figure 1.12, where it can be seen that evolutionary and time distances do not correspond to each other.

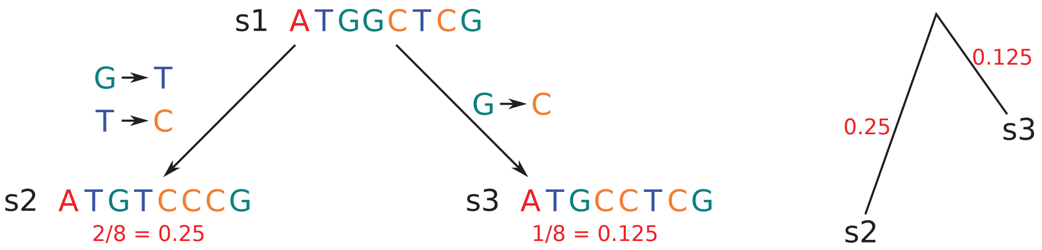


Figure 1.11: On the left, the evolution of the sequence s1 into sequences s2 and s3. The computation of the distance (in substitution per nucleotides) between s1 and s2 (respectvely, s3) are shown under s2 (respectively s3) in red. On the right, the corresponding tree of s2 and s3, with branch lengths in red.

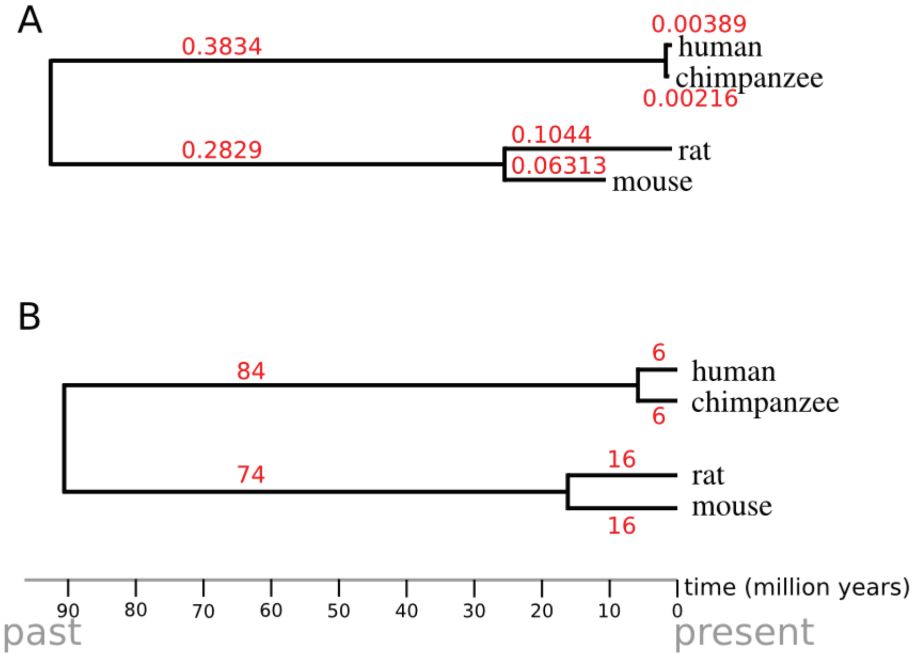


Figure 1.12: Two phylogenetic trees of the rat, human, mouse and chimpanzee with branch lengths in red. A. the branch lengths correspond to expected substitutions per site between the sequences of the 16S ribosomal protein of each species. B. the branch lengths correspond to time (expressed in million years) (divergence times were obtained from timetree.org *[last accessed 5th of August 2017]*

Indeed, using the numbers of Figure 1.12 again, between the rat and its last common ancestor with the mouse, 16 million years have passed while 0.1044 substitutions per sites accumulated, which gives a rate of $0.1044/16 = 6.525 \cdot 10^{-3}$ substitutions per site per million years. Compare this with the case of the human, for whom the rate of substitution in the branch that separates it from its common ancestor with the chimpanzee is $0.00389/6 = 6.483 \cdot 10^{-4}$ substitutions per site per million years. These differences are due to the fact that certain sequences accumulate mutations faster than others depending on the selective pressure exerted on them and the biology of their host organism.

When the branches of a tree correspond to time, the tree is said to be *ultrametric*.

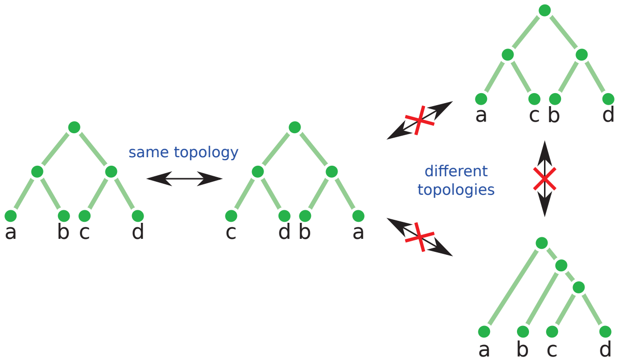


Figure 1.13: Similar and dissimilar topologies between rooted trees with 4 leaves

number of leaves n	unrooted T_n	rooted Tr_n
2	1	1
3	1	3
4	3	15
...	...	...
n	$T_{n-1} * (2n-5)$	$Tr_{n-1} * (2n-3)$
...	...	...
10	2 027 025	34 459 425
20	$\sim 2.22 * 10^{20}$	$\sim 8.20 * 10^{21}$

Table 1.1: Growth of the number of rooted and unrooted topologies with the number of leaves.

The *topology* of a tree refers to the way it successively links together groups of leaves in subtrees. In other words, the topology of a tree is its structure, without branch lengths. Trees with similar and different topologies are shown in Figure 1.13. Note that while the examples shown on the figure are rooted trees, the same concepts apply to unrooted trees. In a phylogenetics framework, two trees with different topologies (but the same set of leaves) tell two different stories of evolution and differentiation between biological objects.

For bifurcating trees, the number of possible topologies is determined by the number of leaves of the tree. For n leaves, I call the number of possible unrooted and rooted topologies respectively T_n and Tr_n. The evolution of these numbers is presented in Table 1.1. Note that The number of topologies grows exponentially with the number of leaves, which means that even for a relatively low number of leaves (say, 14), it is hard to iterate across all the possible topologies.

1.2 Co-evolution and the consequences of statistical independence in an interdependent world

Co-evolution may be defined as the phenomenon where two or more biological entities reciprocally affect each other's evolution.

In a model of evolution without co-evolution, biological traits evolve according to pressures determined by their environment. This environment may change, but is considered independent from the biological traits it exerts pressure on. In co-evolution, the environment of a given biological trait is itself subject to evolutionary pressures coming from this trait.

This term first appeared to describe the evolutionary interactions between communities of butterflies and the plants they feed on [Ehrlich and Raven, 1964]. They posited that the evolution of plants secondary substances (which may have poisonous or repellent properties) exerted a selective pressure on phytophagous insects which resulted in an adaptation to these secondary substances. In return, the role of phytophagous insects as "predators" of plants also exerted a selective pressure on plants.

From the study of populations interaction, co-evolution has been applied to other systems. For instance, it has been used in hypotheses about the emergence of important cell biology mechanisms [Lacey *et al.*, 1975; Gregory, 2001]. It has also been used to study the observed correlation between changes in quantitative traits or to describe the evolutionary interaction between biological molecules, such as proteins (this approach is called molecular co-evolution)[Codoñer and Fares, 2008].

Co-evolution occurs at all scales and appears to be an idea essential to the understanding of life. Indeed a model without co-evolution has a reductionist approach that recursively cuts down entities into smaller ones (genomes in genes, genes in nucleotides) and considers bigger entities as bags of smaller, independent, ones.

However, it appears that in biology many interesting phenomena, perhaps the most significant ones, do not concern the entities themselves but rather the interactions that they entertain between each others[Sapp, 1994; Soyer, 2012]. Such cases include symbiosis, protein interactions, metabolism / expression regulation, ecological community dynamics. In all these, an entity is not so much defined in terms of its internal qualities as it is by the sum of its relationships with other entities. Access to these relationships (and their histories) can be seeked through the modelling of co-evolutionary processes.

At the scale of genes and genomes, a concrete example of the limits of the reductionist modelling approach is that it does not allow the description of the sizes of chromosomal duplication events in vertebrates (as each gene is considered independently, one only sees single gene duplications). As we know that the majority of the genome is composed of repetitions, duplicated elements (for instance in humans, see de Koning *et al.* [2011]) and has undergone whole-genome duplications, we are effectively missing information on a very important process of genetic innovation. Such biological questions drive me to go beyond the "classical" approach that I described and consider the history of genes in the context of co-evolution.

In the following sections I will describe the gene in term of the co-evolutionary relationships it is part of, how these relationships are detected, how the history of these relationship may be inferred, and how these relationships may be used to gain information about a gene history.

1.2.1 Nucleotide and gene co-evolution

Nucleotides/amino-acid co-evolution

Co-evolution occurs at all scale of the living, including between different residues of the same gene. For instance, such co-evolution relationships can arise from a physical interaction that is needed to maintain a molecule (*e.g.*, a protein) 3D structure. In such relationship, the effect of a mutation in one residue depends on the current state of the other residue and vice-versa. This leads to correlated patterns of substitutions between different sites of an alignment.

These co-evolutionary interactions also imply that, at a given time, there only exists a fraction of the positions in a protein that have *acceptable* mutations (*i.e.*, a mutation that will not be selected against). The *covarion* (for concomitantly variable codons) model [Fitch and Markowitz, 1970] accounts for these effects of co-evolution by authorizing positions in the alignment to shift in and out of a state of invariance during their evolution. Here the co-evolutionary links between sites are not explicitly modelled : it is only the effects of the co-evolution that is emulated.

Models of codon evolution are another interesting models of evolution that imply a measure of co-evolution between sites of an alignment (see for instance Pouyet *et al.* [2016]). Indeed, each codon represents a group of three nucleotides whose evolution is particularly linked together : the transition probability from one nucleotide to another depends on whether or not this mutation leads to a change in the amino-acid sequence (and if it does, toward which amino-acid) and this depends on the state of the two other nucleotides of the codon.

Rather than using co-evolution information, many methods seek to detect and describe the co-evolutionary links between residues (both between residues of the same gene and of different genes) as they represent useful tools for the understanding of molecules functions and of the selective pressures acting on different sites.

Aside from methods based on directed mutagenesis experimentation or analysis of the three dimensional structure of proteins, computational methods can be crudely separated into two groups: the ones that do not account for phylogeny and the ones that do[15]. The first kind will seek correlated alignment columns and rely on measures such as mutual information content or entropy. The second recognizes that some of the observed correlation between alignment columns is caused by their evolution along the same phylogenetic tree (*i.e.*, they belong to the same gene) and seeks patterns of correlated substitution events (see Dutheil [2012] on the subject of accounting for phylogeny when detecting co-evolution between residues).

Codoñer and Fares [2008] propose the following decomposition of the covariance between two amino-acids:

$$C_{ij} = C_{phylogeny} + C_{structure} + C_{function} + C_{interactions} + C_{stochastic}$$

Where $C_{structure}$, $C_{function}$ and $C_{interactions}$ relate to co-variation due to the same selective force acting on both sites (respectively to maintain a protein structure, function, or interaction with another molecule). $C_{stochastic}$ relates to a noise that makes sites appear to be correlated when they actually are not. Finally, $C_{phylogeny}$

[15] See Codoñer and Fares [2008] for a review on these methods and the stakes of molecular co-evolution.

corresponds to the signal inherent to the fact that the sites share the same phylogeny.

For the purpose of gaining insight on the phylogeny of an object, I will focus on this idea that sites share a phylogeny. So rather than considering correlated events of substitutions, I am interested in correlated events of diversification (*i.e*, bifurcations in a tree)[16]. In the next section, the references to co-evolution I make should be understood to refer to the correlation of phylogenies.

Sequence co-evolution

In the previous section, I wrote that the alignment establishes homology relationships between the nucleotides of several homologous *genes*. Then the phylogenetic information contained in each column of the alignment is combined in a phylogenetic tree that describes a unique and coherent history for this whole group of nucleotides. Moreover, as mentioned before, each column of the alignment is considered separately during the phylogeny inference: each contributes independently to the likelihood of the tree (as well as the likelihood of the other parameters of the model of evolution).

As such, each nucleotide could be seen as its own mini-gene. All of these minigenes (*i.e.*, all positions of the alignment) are then supposed to co-evolve together to such a degree that they share the same phylogeny: the gene phylogeny. The coevolutionary relationship between each position in the alignment is thus supposed here to be total: the nucleotides may not have a different history (see however Boussau *et al.* [2009] for a model allowing for more than one tree for an alignment).

Having all the sites of a gene co-evolving together in such a manner comes down to seeing the gene a an **atom of evolution**: it forms a coherent, indivisible unit of evolution.

However, looking at an alignment, it can appear like two neighbouring columns may not have the same history. Cases where different columns of the same alignment display different trees may come from different sources.

- It is possible that the definition of the gene was erroneous: it groups positions that do not, in fact, share the same history.

[16] One could then speak about a state of co-phylogeny between sites to mean that they share the same phylogeny. However, this specific term has already been claimed by a scientific community to describe the host-parasite phylogenies relationship.

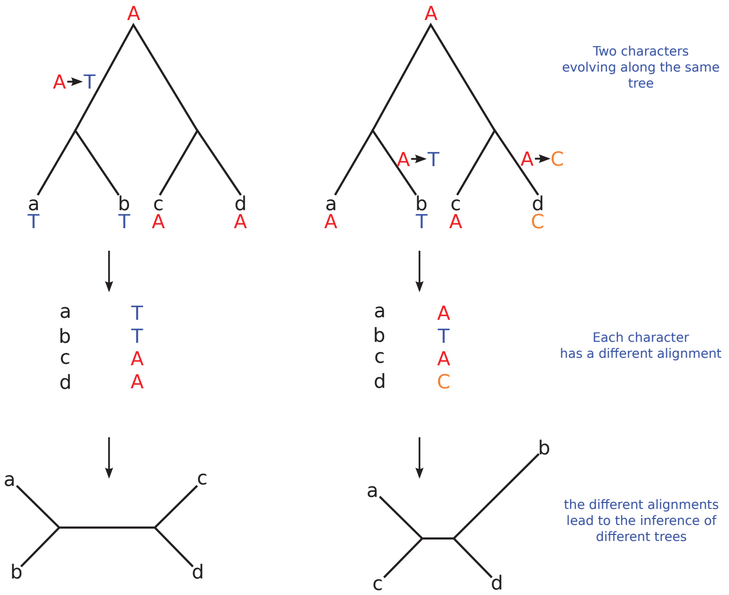

A
A→T
a
T
b
T
c
A
d
A
A
A→T
A→C
a
A
b
T
c
A
d
C
Two characters evolving along the same tree
a T
b T
c A
d A
a A
b T
c A
d C
Each character has a different alignment
a
b
c
d
a
b
c
d
the different alignments lead to the inference of different trees

Figure 1.14: Individual positions evolve along the same tree, but the substitutions they undergo produce patterns that do not reflect this tree. The two positions will then correspond to alignment columns which in turn, if we inferred a tree on each column separately, would lead to different *nucleotide trees*.

- The homology detection and alignment phase are prone to error and may lead to alignments segments where the aligned positions are not homologous.
- The process of evolution may lead to confusing patterns in extant sequences.

The third case may arise from stochastic processes like back mutations (a position mutated once, then mutated back to its previous state) or convergent evolution. Furthermore, they may be aggravated by differences in the selective pressures exerted in the different positions of the biological sequence (certain site may appear to change more slowly than other because each mutation they undergo is counter selected). These phenomena do not necessarily violate the view of the gene as an atom of evolution. However they are worth noting as they introduce uncertainties in the tree reconstruction process (as different columns of the alignment will support different trees, as shown in Figure 1.14).

The second case can also lead to more uncertainty in the tree reconstruction phase, but rather than denoting the production of different patterns of evolution from entities sharing the same history, it may rather come from the production of similar patterns from entities that do not share the same history, thus leading to similarity and spurious homology.

Finally, the first case is more problematic in the context of phylogeny as it means that the alignment cannot be represented by only one tree: there is more than one gene in the alignment. This can occur even when all the columns of the alignment are aligning homologous sites, if different parts of the sequences of interest followed different histories.

Such cases come close to the concept of modular protein evolution, which considers that proteins (*i.e.*, protein coding genes) do not constitute atoms of evolution, but rather are combinations of different subunits (modules) which are the atoms of evolution[Moore *et al.*, 2013].

For a proper phylogenetic analysis, an alignment should correspond to one gene only and steps for correcting such errors (*i.e.*, detecting segments of alignments with different evolutionary histories and splitting the alignment accordingly) should be undertaken (see Minin *et al.* [2005]; de Oliveira Martins *et al.* [2008] for instances of algorithms to detect segments of alignment with different phylogenetic signatures).

Grouping (neighbouring) nucleotides together in the same gene forces us to make the hypothesis that the grouped nucleotides completely co-evolve together, which may not be true. Only the nucleotide seems to constitute an irreducible unit of evolution. However a gene containing a single nucleotide often will not contain enough information on its own to reconstruct its entire evolutionary history. The same can be said from genes composed of multiple nucleotides: a complex history (*i.e.*, one involving many copies of the gene) may require a lot of signal to be robustly inferred, possibly more than what the gene contains.

From a practical point of view, one could argue that this is a question of balance and goal. The gene should be constituted so as to be atomic enough (*i.e.*, its nucleotides have the same, or almost the same, history) so as to have meaning from a phylogenetic point of view. And it should include enough nucleotides so as to contain enough phylogenetic signal to robustly infer its history.

As I take the stance that one should privilege the first criterion (atomicity of the defined gene), it becomes then interesting to ask ourselves if there is a way to still infer the correct gene phylogeny, despite a limited amount of information in its alignment. I thus have to consider other sources of information about the gene phylogeny.

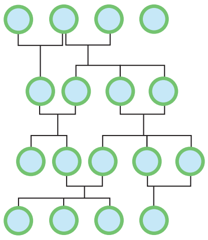


Figure 1.15: Parental relationships (black lines) between individuals (dots). The parental generation is represented above its children.

1.2.2 Species and gene co-evolution

Genes do not evolve independently in nature: they occur in genomes, which contain many genes. As such the events that the different genes of a genome undergo along their evolution have an influence on the evolution of the genome. Complementarily, events occurring to the genome (for instance, a *whole genome duplication*) also have an influence on the history of each of the gene it contains: there is a co-evolution between genomes and their constituent genes.

In order to describe the co-evolution of genes and genomes, I must first define the genome history.

While each individual possesses a different genome, individuals of the same species have very similar ones and will recombine them together (especially in species engaging in sexual reproduction).

This means that while it is possible to retrace the parental relationship between individuals of the same species (in a fashion similar to the building of a genealogical tree), such parental relationships are based over fine differences between their genomes and these relationships are not tree-like in shape, as represented in Figure 1.15.

However, when looking at broader time and biological scale, the differences between individuals of the same species are dwarfed by the differences between species. Moreover as individuals of different species do not normally exchange genetic material[17] the parental relationships between individuals of different species take the form of a tree, as can be seen in Figure 1.16.

Because of this, studies interested in processes happening over a long evolutionary time tend to summarize individual genomes into species genome. Species genomes

[17] See however the hybridization, and description of lateral gene transfer (later in this document) for instance of cases where individuals of different species exchange genetic materials.

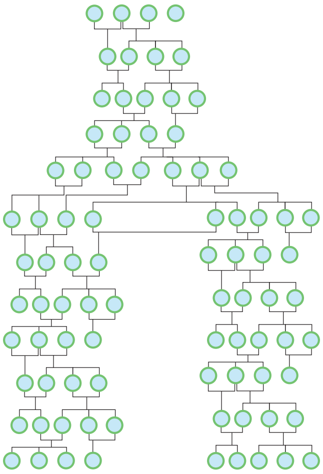


Figure 1.16: Parental relationships (black lines) between individuals (dots). The parental generation is represented above its children. Two sub-populations are not reproducing together, making the pattern of descent take the shape of a tree over time.

history are summarized into what is called a *species tree*.

Species tree

In a species tree, the leaves correspond to extant species and internal nodes correspond to the *speciation* of an ancestral species. Speciation describes the process by which new species are born. A model for speciations describes a population (of individuals of a given species) that separates into two sub-populations that reproductively isolate (*i.e.*, they gradually or abruptly stop reproducing together). As the two sub-populations reproduce less, their genetic material starts to evolve independently: they accumulate differences. These differences can, in turn, preclude sexual reproduction between individuals of the different sub-populations (in the case of sexual species), thus enforcing the reproductive isolation and independent accumulation of differences which will result in different species. The reproductive isolation may be due to a physical isolation (*e.g.*, the sub-populations are situated on different islands without the mean to cross water), or simply arise from within the population because of the mechanism of genetic polymorphism [Gavrilets, 2014].

Contrary to genes where it is possible to build a single linear multiple sequence alignment upon which the gene tree is inferred, it is not possible to infer a single alignment for whole genomes because of rearrangements[18].

However, because genes are parts of genomes, genomes-wide events such as speciation impact genes and are reflected in their phylogenies, so that it is possible to use the phylogeny of genes to reconstruct the history of the genomes that bear them.

But using any single gene tree, changing the gene labels into species labels and declaring it the species tree is not desirable. A gene may be situated on a sequence segment that has undergone an history different from the rest of the genome: this segment may have been deleted in a species, or duplicated in several copies in the genomes, or even have been transferred into this genome[19]. Moreover, an individual gene tree may be prone to error, because it derogates from the hypothesis made by the methods used to infer its phylogeny or simply because it does not contain the necessary signal for a robust tree reconstruction : its nucleotides have accumulated too few substitutions, or too many (leading to saturation). That is why species tree inference methods will combine the information of several gene trees: this reduces the (random) error associated with each individual gene tree by increasing the amount of data, while also reducing the error coming from possible differences between individual genes and species histories as it is expected that these differences are part of the history that most of the genes do not share.

One such type of methods is termed *concatenated alignment* (also referred to as *supermatrix* approach). The alignments of several genes families (containing one copy per species of interest) are concatenated head to tail. A single phylogenetic tree is then inferred on this concatenated alignment. In other methods, phylogenies are inferred independently for each gene family and then these trees are combined together to produce a *consensus tree* or a *supertree.* This second approach accounts for important differences in the substitution processes between individual gene families, but it asks the question of how to combine together the information of different gene trees. The most common methods to do this is are majority rule consensus [Adams, 1972] (for a comparison of concatenated-alignment and majority rule consensus, see Gadagkar *et al.* [2005]) and supertree methods [Gordon, 1986; Cotton and Wilkinson, 2007; Ranwez *et al.*, 2007] which make use of clades or bipartitions that the different trees have in common (not unlike the method to compute posterior clade

[18] Whole genome alignment would at best yield a group of multiple sequence alignments and is, in itself, an open question as exemplified by the Alignathon collaborative project[Earl *et al.*, 2014].

[19] I discuss further these events, where a gene phylogeny differ from the species phylogeny, in the next section.

probabilities mentioned earlier in the context of Bayesian phylogeny inference).

Yet other methods combine gene trees while accounting for their differences with the species tree in a more explicit manner.

Differences between the species tree and gene trees

As mentioned earlier, genes can have a phylogeny different from the phylogeny of their hosts species[20]. Such differences are caused by mutation events that only affect a portion of the genome where a gene copy is situated. Maddison [1997] described the discord between gene tree and species tree and identified a number of processes underlying it, shown in Figure 1.17. Note that this figure shows a gene tree drawn inside a species tree, symbolizing the fact that the genome of species contains genes.

Gene Duplication. (1.17 B) Gene duplication occurs when the segment of chromosome (or of a whole chromosome in the case of whole genome duplication) that bears the gene replicates in the genome. It leads to an increase in the number of copies of the same gene coexisting in the same genome.

There exists several reasons that explain that evolution retains what might seem like redundant information in the genome (the first of them being a purely neutral evolution point of view, where the fixation of the duplicated copies are only explained through random drift), I detail some of them below (for a more complete explanation of these, see Innan and Kondrashov [2010]).

Subsequent differential accumulation of mutations on the duplicated copies can lead to one of the copies acquiring a new function (*neofunctionalisation*) or, when the original gene performed several functions, a specialisation of each copy towards different tasks (*subfunctionalisation*) (this specialisation in one function may imply a respective loss of the other function by the copies, in which case both copies are now needed to perform the functions of the original, pre-duplication, gene copy).

In other cases, the number of gene copy might be directly beneficial to the organism bearing them. This is the case for expressed genes whose level of expression should be as high as possible (such as ribosomal genes, whose product are needed in large quantity) and is referred to as a *beneficial increase in dosage*. It is also possible that the redundancy of genetic information offers a protection against deleterious mutations: even if one copy of the gene is affected by such a deleterious mutation,

[20] Even barring any errors coming from the imperfect methods of tree inference.

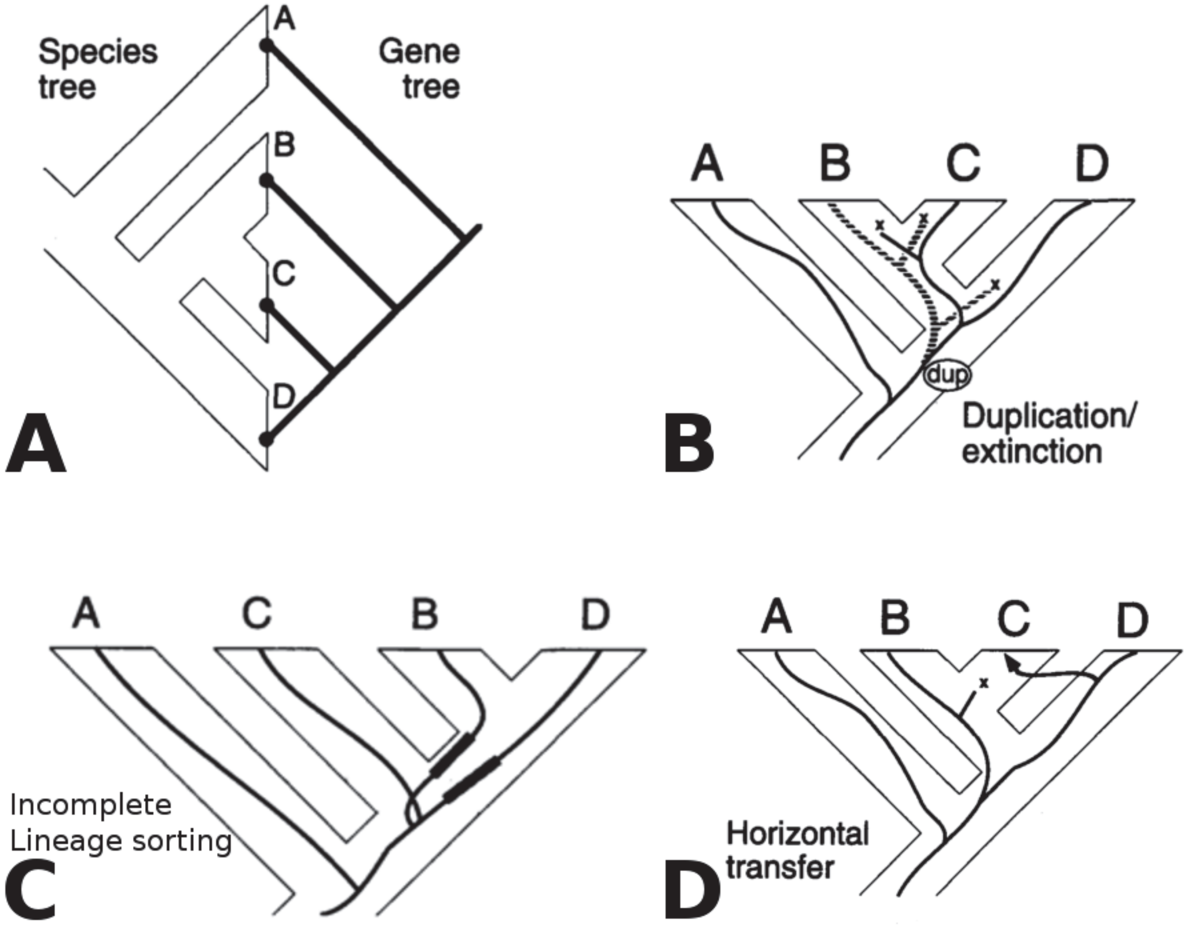


Figure 1.17: Figures 2, 3, 5 and 7 (partial) from [Maddison, 1997] which represent different process of discord between a gene tree and a species tree. **A** Species tree (on the left) of the four species A, B, C and D and gene tree (on the right) with one gene copy per species. Gene tree and species tree differ. **B** Gene duplication (represented by the "dup" tag; the duplicated lineages are indicated by the dashed line) and gene losses (represented by crosses) events can explain the history of the gene along the species tree. **C** Incomplete Lineage Sorting (ILS) implies the coexistence of of several gene lineages along a species tree branch (such lineages are shown as bold here). **D** Horizontal Gene Transfer (HGT) (represented using an arrow) from the genome of a species to the other.

the other will still be functional. This effect is stronger when mutation rates are high.

Gene Loss. (1.17 B) Gene loss occurs when the gene bearing chromosome segment is deleted from the genome (because of a large chromosomal deletion for instance). When considering the evolution of functional genes, a gene loss may also correspond to the pseudogenization of a gene copy.

Contrary to gene duplication, gene loss leads to a decrease in the number of gene copies in a given genome.

Incomplete Lineage Sorting (ILS). (1.17 C)

Consider the existence, at any given time in a species, for a given locus in the genome, of different alleles. These alleles have their own history in the population: they appear through mutation and their relative proportions change with each generation (possibly to the point of allele extinction). When speciation occurs, the alleles borne by individuals in the parent species will be sorted between the two children species. While in some cases the children species will inherit copies from all alleles present in the parent species, in some other cases they will only inherit a subset of these.

Incomplete Lineage Sorting occurs when this allele sorting mechanism give rises to a tree that is different from the species tree. Such a process is illustrated in Figure 1.18.

Horizontal Gene Transfer (HGT) . (1.17 D)

With the three sources of discord described previously (gene duplication, gene loss and incomplete lineage sorting) the gene history is different from the one of the species, but it is nevertheless contained in this species tree. The pattern of descent is said to be *vertical*, as each children always gets its genetic information from its parent. However, organisms have evolved ways to acquire genetic material from distantly related organisms. These processes are grouped under the term Horizontal Gene Transfer (also referred to as Lateral Gene Transfer), as they cause an organism to get some of its genetic material outside of the vertical transmission from parent to children. HGT is an asymmetrical phenomenon where one distinguishes the organism giving some genetic material (the *donor*), from the organism receiving it (the *recipient*).

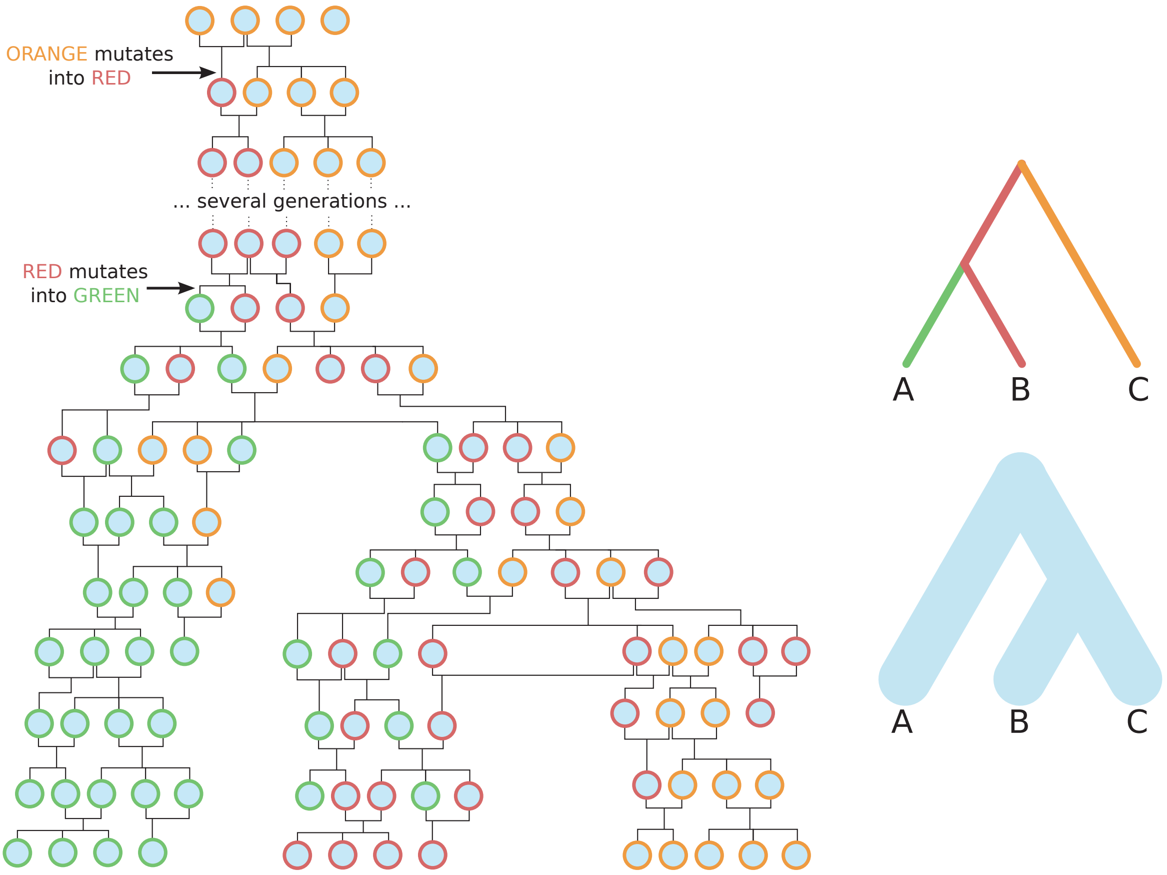


Figure 1.18: A representation of the process leading to ILS. A population with several alleles speciates several times (on the left). Imperfect repartition of the alleles between children species and allele frequency variations lead to a difference between the alleles tree (upper right) and the species tree (bottom right).

Substantial traces of HGT may be detected in nearly all bacterial genomes [Ochman *et al.*, 2000] (but also in archea [Diruggiero *et al.*, 2000] or eukaryotes such as fungi [Rosewich and Kistler, 2000]). This detection may be based on the composition of genomes gene repertoires: the presence of a given gene in only one of two closely related species may be explained by the loss of the gene in the second species (in which case the gene is supposed present in their ancestor) or by a transfer in the first species. Thus high level of difference between the gene repertoires of closely related species forces us to assume a huge ancestral genome size, or to accept that HGT is responsible for a large part of these differences [Daubin *et al.*, 2003b]. Similarly, HGT may be detected from patterns of difference between a gene tree and the tree of the species it evolves in (however precautions must be taken to avoid confusion of such patterns with the one generated by gene duplication and loss or ILS) [Daubin *et al.*, 2003a].

Three different mechanisms of DNA exchange between bacteria have been described [Ochman *et al.*, 2000; Thomas and Nielsen, 2005]: *Transformation*, *Transduction*, *Conjugation*.

Transformation refers to an uptake of DNA directly present in the milieu (*i.e.*, not inside another cell). It requires the recipient cell to be in a physiological state called *competence* (certain bacterial species are perpetually competent while for others this state is more transient) and can result in the transmission of DNA from very distantly related organisms.

Transduction refers to the introduction of new genetic material in a bacterium via a bacteriophage (a bacterium virus)[21]. Contrary to transformation, transduction will only occurs between specific species that can be infested by the same bacteriophage.

Conjugation involves a plasmid-mediated[22] DNA exchange between two cells in physical contact. The DNA received may be conserved in the form of a plasmid, but may also be integrated in the chromosome of the recipient species.

In addition to these methods, a number of mechanisms may results in lateral gene transfer[23]. Whatever the method used for the entry of foreign DNA in the recipient cell, this DNA must then be integrated in the recipient's genome for it to be successful transfer. This integration can take the form of the maintenance of the foreign DNA as a plasmid, but it may also be integrated to the recipient chromosomes. This integration may take the form of homologous recombination, in which case the new DNA replaces a pre-existing, homologous (and similar), sequence.

Homologous recombination, because of the similarity between the *old* and the *new* copy, is not likely to introduce novel functions into the genome of the recipient of the transfer. The other forms of HGT have, however, the potential to suddenly introduce novel functions in the recipient organism (relative to the evolution of

[21] As bacteriophages replicate in a bacterium (the donor), some of them may incorporate some of their host cell DNA inside their capside. These donor-DNA bearing bacteriophages may then infest a new cell (the recipient), thus injecting it with some DNA of the donor cell which can then be integrated within the recipient's genome (provided it does not die from the bacteriophage infestation).

[22] Plasmids are small circular DNA molecules present in bacterial cells (they can also be found in archea, as well as some eukaryotes such as yeast). They are separated from the chromosomal DNA and replicate independently.

[23] Or a lateral transfer signature. Such can be the case of ancient hybridization or introgression that transfer detection method may detect as repeated and important horizontal transfer with homologous recombination.

this gene by accumulation of successive random mutation) and HGT have been recognised as a driving force of the bacterial evolution [Ochman *et al.*, 2000].

As mentioned earlier, HGT stands out from the other sources of discord between species tree and gene tree as it represent a *non-vertical* component of evolution, the part of the gene tree that cannot be written *inside* the species tree, as shown in Figure 1.17 where duplication, loss (1.17 B) and ILS (1.17 C) are drawn inside the species tree, but horizontal transfer (1.17 D) shows a branch getting out of the species tree before getting back in.

This action of drawing the gene tree inside (or around, in the case of HGT) the species tree can be seen as a way to specify the events that happened along a gene phylogeny, and in which species these events happened. Such a specification is called a *reconciliation.*

Reconciliation

Reconciling a gene tree with a species tree comes down to associating the different parts of the gene tree (nodes, corresponding to extant or ancestral genes) to:

- a node of the species tree (*i.e.*, an extant or ancestral species)
- an evolutionary event (such as Speciation or a Duplication for instance)

The result of a reconciliation is called a *reconciled gene tree.*

A basic question in the study of reconciliation then is: given a species tree, a gene tree whose leaves are associated to extant species, and a set of events, can we infer the reconciled gene tree, as exemplified in Figure 1.19 with events of speciation, gene duplication, gene loss and horizontal gene transfer.

Solutions to instances of this question, varying in the framework used and/or the set of events considered, have been proposed over the years. Some consider only events of gene duplication and loss (both in a parsimony-based [Goodman *et al.*, 1979; Bonizzoni *et al.*, 2005], and in a likelihood-based framework [Arvestad *et al.*, 2003]), or only events of horizontal gene transfer and losses [Suchard, 2005; Boc *et al.*, 2010].

Others consider together events of duplication, loss and incomplete lineage sorting, such as Rasmussen and Kellis [2012] which builds on the multi-species coalescent (a framework who had been previously developed mainly for other purposes, including the inference of ancestral population sizes [Rannala and Yang, 2003]).

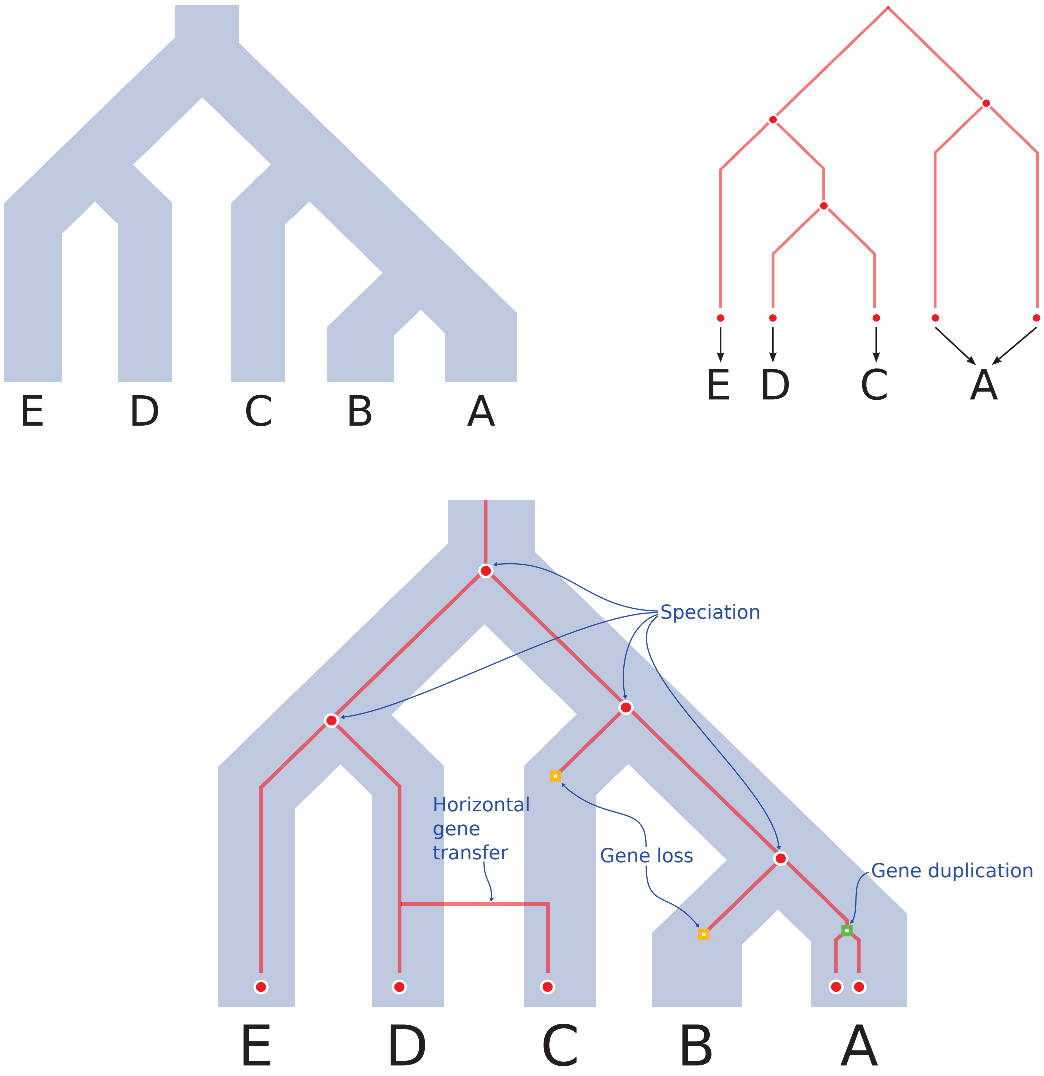


Figure 1.19: A species tree (upper left), a gene tree (upper right) and their reconciliation (bottom). The gene tree (upper right) has an association between its leaves and extant species of the species tree (A, B, C, D and E) (which means that the extant gene is found in this extant species); two genes are associated to species A, no gene is associated to B.
The reconciled gene tree (bottom) is represented *inside* the species tree. Speciations (and leaves) are represented as white-circled red dots; gene duplications are represented as green squares, gene losses are represented are orange squares; horizontal gene transfers are represented as a branch of the gene tree going out of the species tree, toward another branch of the species tree.

Yet others consider events of duplication, loss and horizontal gene transfers (see instances in a parsimony [Doyon *et al.*, 2010; Bansal *et al.*, 2012] or a likelihood [Szöllősi *et al.*, 2012; Tofigh *et al.*, 2011; Sjöstrand *et al.*, 2014] framework).

Stolzer *et al.* [2012] considers duplication, loss, horizontal gene transfers and ILS in a parsimony framework, but only when ILS is restricted to some, non-binary, nodes of the species tree.

Variants to this question include instances where the gene tree is considered non-binary [Lafond *et al.*, 2012], where the species tree is not binary [Vernot *et al.*, 2008], or where the species undergo hybridization events (the species are then not represented by a tree, but by a *species network*) [Than *et al.*, 2008].

During my work, I will generally not consider events of ILS and species hybridization and rather focus on models implying events of duplication, loss and transfer of genes.

Henceforward, I will refer to models considering only gene Duplication and Loss as DL models, and models considering Duplication, horizontal gene Transfer and Loss as DTL models.

Beyond the inference of a reconciliation given a gene tree and a species tree, a bigger question is to integrate the idea that species and genes are co-evolving entities and that understanding the evolution of one gives information on the other.

Reconciliation as a view of co-evolution

As mentioned earlier, genes and genomes co-evolve. Reconciliations, as they describe the relationship between a gene tree and species/genomes history, have a role to play in the deciphering of the history of this co-evolution.

The fact that this relationship is not a perfect one, and in particular the presence of horizontal gene transfer, has lead some to doubt the idea that a tree of species could be built or should be built [Doolittle, 1999; Dagan and Martin, 2006; McCann *et al.*, 2008]. Indeed, if the tree of species symbolizes the *vertical* component of evolution, then the presence of HGT, which represent the *horizontal* component of evolution, in the genes which are used to infer it will introduce many additional errors to the result. Moreover, if one considers that the history of genomes should be the history of the whole genome (as opposed to the history of vertical descent), then this history does not take the shape of a tree, but rather a network [Doolittle, 1999].

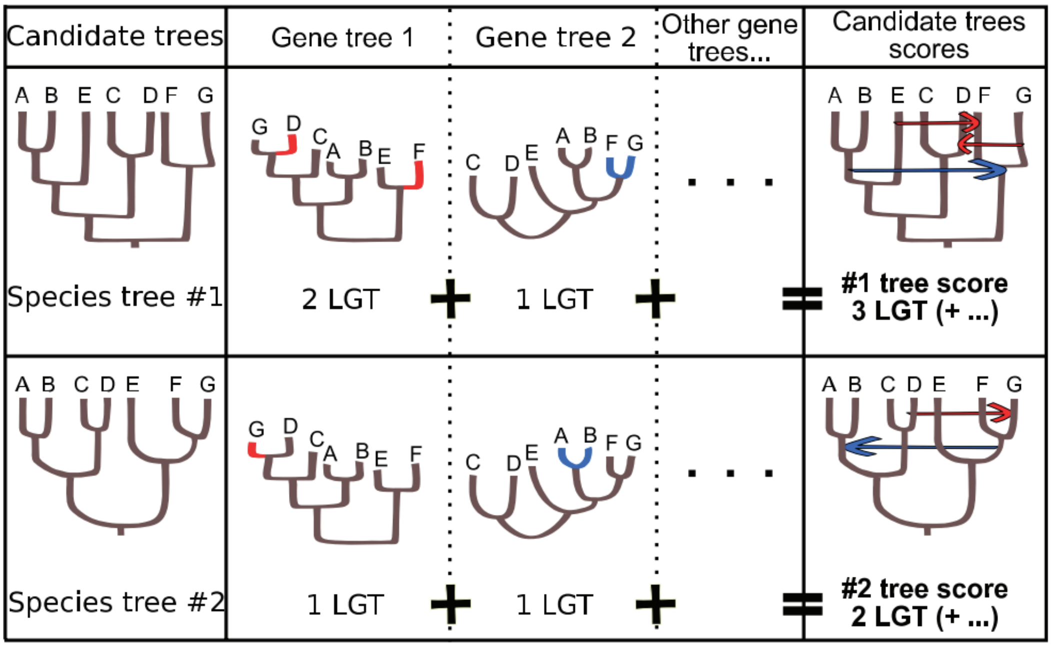


Figure 1.20: Figure 1 from Abby *et al.* [2012] showing the use of HGT events as a support for different species tree topologies.

While this second point mainly relies on what definition we give to "genome/species phylogeny", the first raises many valid concerns about our ability to infer a species phylogeny in the presence of HGT. To address this point, some studies (such as [Galtier and Daubin, 2008; Daskalakis and Roch, 2015]) have demonstrated that it was possible to find phylogenetic signal, and to reconstruct an accurate species phylogeny despite the presence of HGT.

On the same point, Abby *et al.* [2012] make the choice to reconstruct the species phylogeny, not *despite* transfers, but *using* "lateral gene transfer as a support for the tree of life". They do so by considering different species phylogenies in terms of the number of HGT they imply in the reconciliations of the genes they contain (as shown in Figure 1.20). This number of HGT events is then used as a score with whom the best species phylogeny is selected.

What is particularly interesting with this approach is that it favours a species phylogeny according to a metric of the level of co-evolution[24] it shows with its constituent genes.

[24] Considering that HGT is a source of discord between gene and species history, more HGT means a lower level of co-evolution between gene and species and vice-versa.

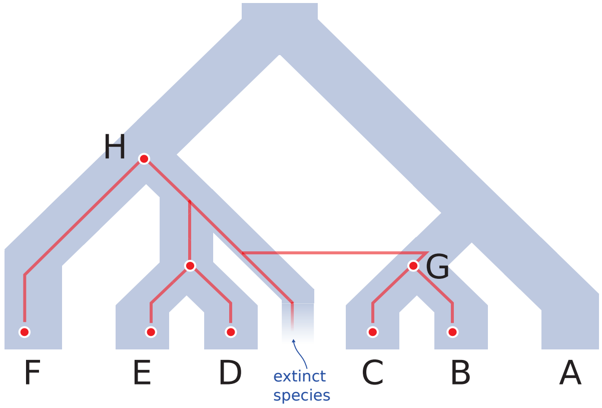


Figure 1.21: An horizontal gene transfer event originating in a species that later goes extinct. This particular event implies that the ancestral speciation H occurred before the ancestral speciation G.

This idea of using the discord between gene and species trees as a source of information to get better species tree have been exploited in the context of ILS (reviewed in Liu *et al.* [2009]), DL models (Wehe *et al.* [2008]; Hernandez-Rosales *et al.* [2012] for instance), TL models (DTL models without duplications) [Abby *et al.*, 2012] and DTL models [Szöllősi *et al.*, 2012]. In particular, an HGT event can serve as a testimony of the co-existence at some point in time of its donor and recipient species. This information can be used to infer relative dates to the speciations in the species tree [Szöllősi *et al.*, 2012], provided one accounts for the idea that the donor species may not have left any descendant in the used species tree (either because it went extinct or its descendant were not sampled as leaves in the species tree [Szöllősi *et al.*, 2013b]) as shown in Figure 1.21.

One can also consider the species tree as a fixed entity and use it to infer better gene trees. For instance, Lafond *et al.* [2012] consider a gene tree with multifurcations (that are typically obtained by collapsing branches of a gene tree with a poor support, thus symbolizing the parts of the gene tree that should be improved) and transform the multifurcations into series of bifurcations that minimize the number of gene duplications and losses they imply. Alternatively, given a gene tree Wu *et al.* [2013] or Bansal *et al.* [2015] search the space of the gene trees with a similar sequence support, looking for the one that minimizes a reconciliation score (Wu *et al.* [2013] used a DL model while Bansal *et al.* [2015] used a DTL model). Szöllősi *et al.* [2013a] goes further by considering jointly sequence support and species tree support (*i.e.*, the reconciliation) of the gene tree in a probabilistic framework. It does so by considering not directly the gene alignment, but rather a distribution of gene trees reflecting the alignment (typically an *a posteriori* distribution obtained from a Bayesian tree inference software). Szöllősi *et al.* [2013a] has seen an adaptation of its approach in a parsimony framework [Scornavacca *et al.*, 2014]. As this approach is of particular importance in my works, I will detail it in another chapter.

Finally, as a co-evolution relationship goes both way, some have attempted to jointly infer gene and species tree [Heled and Drummond, 2010; Boussau *et al.*, 2013], but at great computational cost.

The papers and methods cited here do not aim to give an exhaustive review, but rather some examples of what have been done in the field of reconciliation and associated problems[25]. I prefer to focus here on the idea of reconciliation as a way to describe the co-evolutionary relationship between a genome and its constituent genes through an alignment of the gene tree onto the species tree. This description then can (and has) be used as a mean to add information from the species (respectively gene) tree in the reconstruction of the gene (respectively species) tree.

1.2.3 Gene and gene co-evolution

After having described the co-evolution occurring between a given gene and the nucleotides it is composed of, and between a given gene and the genome it is part of, I will go on to describe a third form of co-evolution: the one between two genes in the same genome.

Two genes may co-evolve because they are interacting with each other when they fulfil their function. For instance, proteins that physically interact together need to maintain a degree of compatibility between their contact surfaces. Such interaction results in a degree of co-evolution which can not only be detected between proteins in direct interaction, but also (albeit to a lesser extent) between proteins without direct interaction but with interactions to the same proteins [Liang *et al.*, 2010]. Similarly, gene products that participate to similar cellular processes co-evolve because they may have to interact with similar compounds and/or to be expressed in a correlated fashion [Luo *et al.*, 2007; Villa-Vialaneix *et al.*, 2013].

Two genes may also co-evolve because they are physically close in a chromosome.

[25] For more extensive reviews, please consult Szöllősi *et al.* [2015] and Nakhleh [2013] on the topic of reconciliation inference, and Daubin and Szöllősi [2016] on the topic of HGT and the "Universal Tree of Life".

Two genes that are neighbours with each other will have a higher probability to be affected by events such as a segmental duplication or large deletion, than two distant genes.

Furthermore, the idea that we expect co-evolution to occur between genes that are neighbours on a chromosome is strengthened by the fact that co-localisation of genes along chromosomes have been shown to be correlated with contribution to a similar function in eukaryotes [Lee and Sonnhammer, 2003] and prokaryotes [Yanai *et al.*, 2002]; co-localisation has also been shown to correlate with co-expression [Hurst *et al.*, 2004]. These correlations are well illustrated in the prokaryotic operon structure (also observed in some eukaryotes [Blumenthal, 2004]) where genes contributing to the same function are clustered together in such a way that their expression is regulated by a single promoter [Jacob *et al.*, 1960]. Another illustration comes in the form of gene fusion, a process where two adjacent protein coding genes become one, which can be used to infer a functional association between genes [Promponas *et al.*, 2014].

Although in practice their effects are hard to distinguish because of the aforementioned correlations, it is interesting to note the difference between mechanisms where proximity on the genome implies a greater chance of being affected by the same, big, structural mutation (such as a duplication) and mechanisms where functional linkage implies a selective pressure favouring correlated mutations.

The co-evolution between two genes can be detected (and described) using their profile of absence/presence in different species. Co-evolution between two genes means that the disappearance/apparition of one will have an influence on the other, so it is expected that pairs of co-evolving genes show a higher correlation between their profile of absence/presence than random pairs of (non co-evolving) genes [Pellegrini *et al.*, 1999].

However, a simple comparison of profiles in extant species can fail to capture the complexity of the relationship between two entities. Because it does not account for the phylogeny of the species it consider, it fails to distinguish scenarios of correlated changes (here, gains / losses of gene) that recount a very different history of co-evolution. Dutheil [2012], while talking about co-evolving positions in a molecule, illustrates nicely this point in his first figure, shown here as Figure 1.22 where the same correlated patterns are shown to arise from two different scenarios, each of which implies a different number of correlated events (here, substitutions) and thus

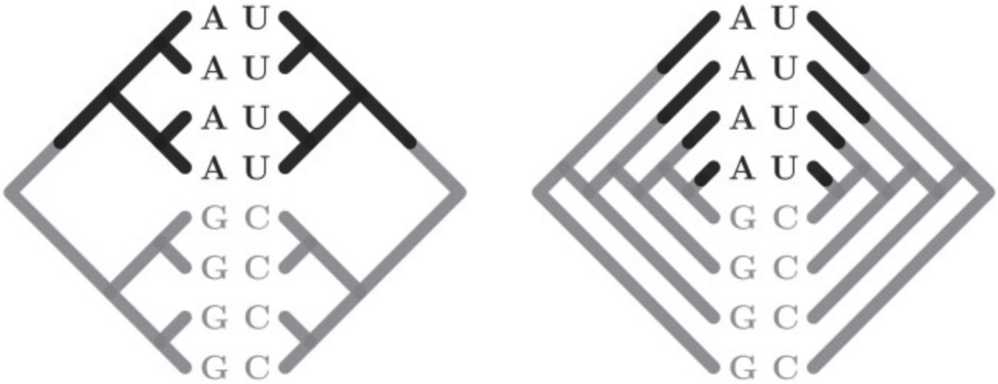


Figure 1.22: Figure 1 from Dutheil [2012] illustrating how two scenarios display the same correlated patterns (here, A associated with U and G associated with C). The correlation between extant profiles is the same for both scenarios, but the underlying evolutionary process is different. The leftmost scenario shows only 1 correlated substitution event (symbolized here as a transition from grey to black), while the one on the right implies 4 co-substitution events. With this in mind, the scenario on the right constitutes a stronger case for co-evolution as it is more unlikely to have happened by chance if the characters are independent.

provides a different signal of co-evolution.

Correspondingly, some methods look for signal of co-evolution by assessing the level of *congruence* between the phylogenetic tree of genes. This is the case of the *mirrortree* approach [Pazos and Valencia, 2001; Pazos *et al.*, 2005; Juan *et al.*, 2008] (see Figure 1.23) which considers the genes trees in the form of their distance matrix (the distance matrix of a tree contains pair-wise distances, either topological or weighted by the branch length, between each pair of leaves of the tree). They then compute the correlation between the two matrices. To be able to compare the matrices in such a fashion, they use a 1:1 association between the leaves of each gene tree (typically because they are in the same species) to ensure the matrices rows (and columns) match. Similar approaches, differing in the way the matrices correlation is assessed, have been developed such as the Congruence Among Distance Matrix test [Legendre and Lapointe, 2004], created for the general context of distance matrices comparison (they applied it to the comparison of whisky properties), but whose performance in the context of phylogeny have been assessed[Campbell *et al.*, 2011]. Yet other methods use a similar approach but consider gene tree distributions rather that individual gene trees [Arnaoudova *et al.*, 2010], in order to take into account

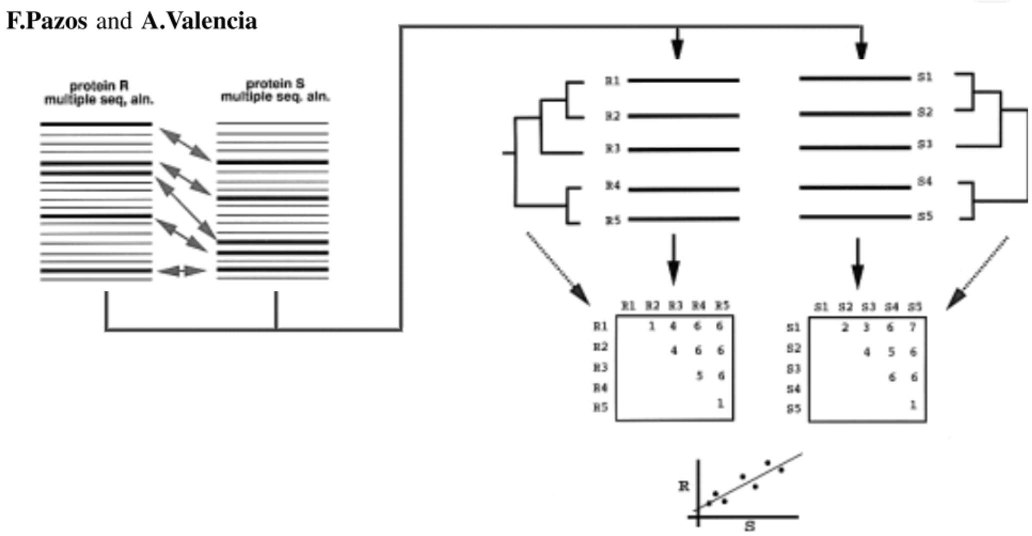


Figure 1.23: Illustration 1 from Pazos and Valencia [2001] showing the *mirrortree* procedure.

the uncertainty inherent to the tree reconstruction process.

While they offer a good way to assess whether or not two gene phylogenies look alike, these congruence approaches all suffer from the fact that they operate based on a 1 : 1 association between the leaves of the species tree. This bars the comparison of genes which have known a different history of duplication and loss (and transfer, when biologically relevant), or at least forces the user to make a decision on which leaves will be associated together, and which will be removed from the study. Another critic lies with the idea that these methods usually don't take into account the tree of species (although see Pazos *et al.* [2005] where the congruence between two genes is corrected by their congruence with the species tree, but who introduce the additional constraint that there must be exactly one gene copy per species) within whom the genes evolve. Hence two genes that do not co-evolve, but that both follow the species phylogeny strictly will be highly congruent despite an absence of co-evolution. Finally, these methods condense the view of the co-evolution between two genes in the form of a single statistic. They do not allow access to the history of the relationship uniting the genes, to the changes in this relationship across lineages (two genes may co-evolve in a species clade, but not in another).

A step in this direction can be seen in Barker *et al.* [2007] which proposes a

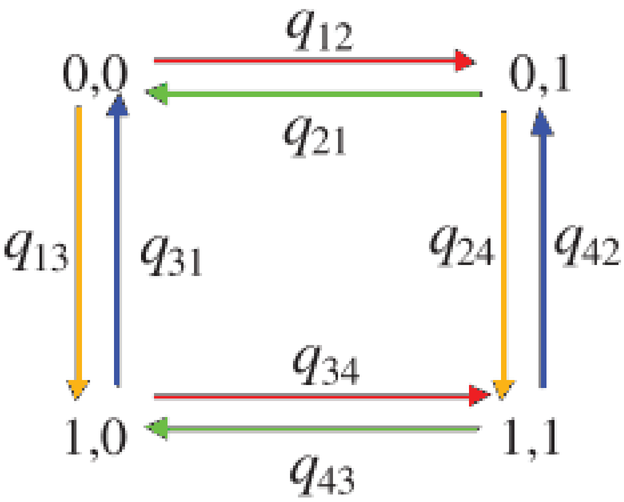


Figure 1.24: The models of Barker *et al.* [2007] as a graph with 4 states: $0,0$, $0,1$, $1,0$ and $1,1$ representing the four possibilities of absence/presence of two genes in a species (for instance $0,0$ means that both are absent while $1,0$ means that the first is present but the second absent.) The states are linked together by transition probabilities (q_{12} , q_{21} , ...) from one state to another. In the first model, each transition probability is free to take any value. But in the second model, independence between profiles is enforced by ensuring that transitions probabilities associated with an arrow of the same colors are equal (forcing, for instance, $q_{12} = q_{34}$ which means that the probability to gain the second gene is the same whether or not the first is present).

method to test for the non-independence of gains and losses of two genes. It does so by comparing two models of evolution of the presence/absence profiles of two genes across a species tree in a likelihood framework: one that allows for a bias toward correlated gains or losses, and the other that forces independence (represented in Figure 1.24; note that here, at it models presence/absence profiles, a gain in one family can only happen if there is no gene of that family already present). Additionally, this method can be used to infer the ancestral states of both profiles and pinpoint correlated (and non-correlated) events along the species tree, which comes closer to a description of the history of the relationship than what the congruence methods yield.

Other methods model more directly the relationship itself, which I will refer to as an *adjacency*. An adjacency is thus a binary (meaning that it is either present, or absent, without middle ground) relationship that links two genes.

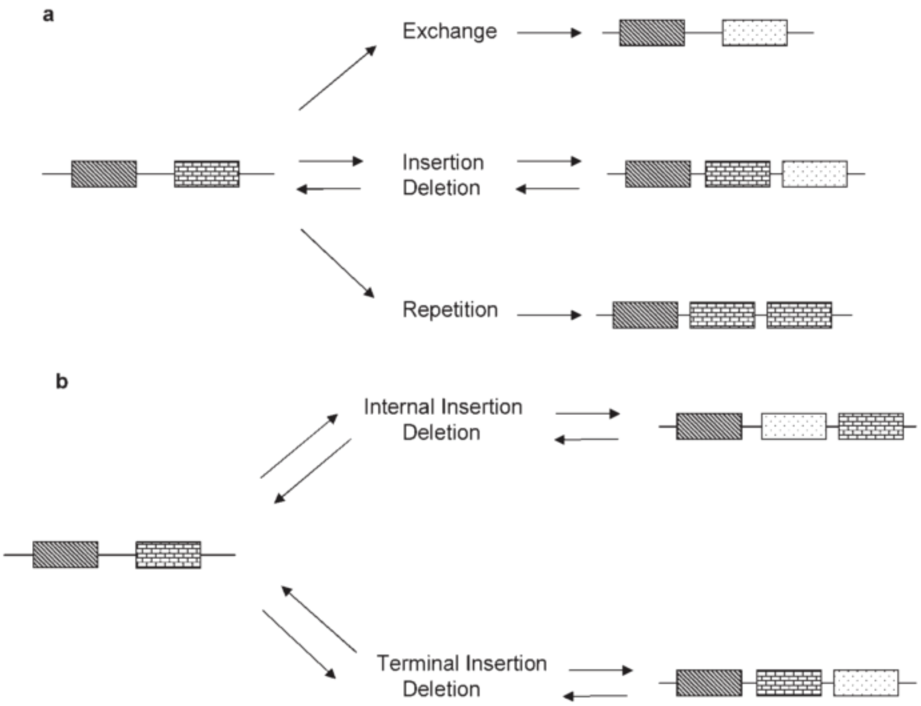


Figure 1.25: Figure 1 from Pasek *et al.* [2006] that illustrates domain architectures (as strings of domains linked by adjacencies) and the different events they can undergo.

Methods that model the adjacencies more directly include the one from Ma *et al.* [2006] who infer ancestral chromosomal regions by looking at the evolution of the relationships between adjacent conserved regions along chromosomes. Their method starts by the identification of such regions and their extant organizations (*i.e.*, the set of *adjacencies*[26]). It then reconstructs ancestral adjacencies following a parsimonious principle that minimizes changes in organization over time.

At another scale, Pasek *et al.* [2006]; Moore *et al.* [2013] look at the evolution of modular protein arrangements in eukaryotes. Thus, they consider protein domains families and link by an adjacency two protein domains if they are adjacent in a protein. Thus, a set of domains linked by adjacencies forms a protein, otherwise called *domain architecture* here. They focus on the quantification of the different events underlying domain variability between protein (intra and inter genome) and avoid potentially difficult cases of ancestral reconstruction by restricting their analysis to domain architecture that only differ by at most one event such as fusion (appari-

[26] used here to represent the relationship between two neighbours along a chromosome

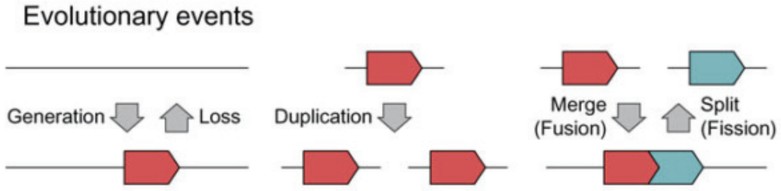


Figure 1.26: Figure 1D from Wu *et al.* [2012] representing the different events considered in their domain architecture evolution model.

tion of an adjacency between two previously unlinked domain architectures), fission (disappearance of an adjacency that cause a domain architecture to become domain architecture), internal or terminal domain addition or loss (see Figure 1.25).

Wu *et al.* [2012] proposes a more formal method to reconstruct ancestral domain architectures. From extant protein domains and their extant adjacencies, ancestral domains and ancestral adjacencies between domains are inferred in a parsimony framework that minimises a joint score taking into account events of domain gain (apparition of a new domain), domain duplication, domain loss (both similar to gene duplication and loss), merge/fusion (apparition of an adjacency between two previously unlinked proteins) and split/fission (disappearance of an adjacency that cause a protein to become two proteins) as shown in Figure 1.26. This model implies that a duplication and loss scenario (but not a reconciliation, as no domain trees are created) of each domain is implicitly inferred jointly with the ancestral adjacencies. This method is well adapted to eukaryotes (they applied it to *Drosophila* species here), in the absence of ILS or horizontal transfer.

DeCo [Bérard *et al.*, 2012] models the evolution of relationships between pairs of genes in a broader framework, mainly, but not limited to, revolving around physical neighbourhood relations. It considers *adjacencies* as the link between two genes and tries to infer an history of adjacencies that minimizes the number of adjacency gains (appearance of an adjacency) and adjacency breakages (disappearance of an adjacency), given a set of adjacencies between extant genes, the gene reconciliations and a species tree. This method stands aside from the ones I cited previously as it explicitly considers the genes reconciliations which, in turns, allows it to consider cases where an evolutionary event (such as a duplication) occurred in two adjacent genes at once, and thus in the adjacency between them as well (for instance an adjacency duplication); a visualization of these ideas can be seen in Figure 1.27.

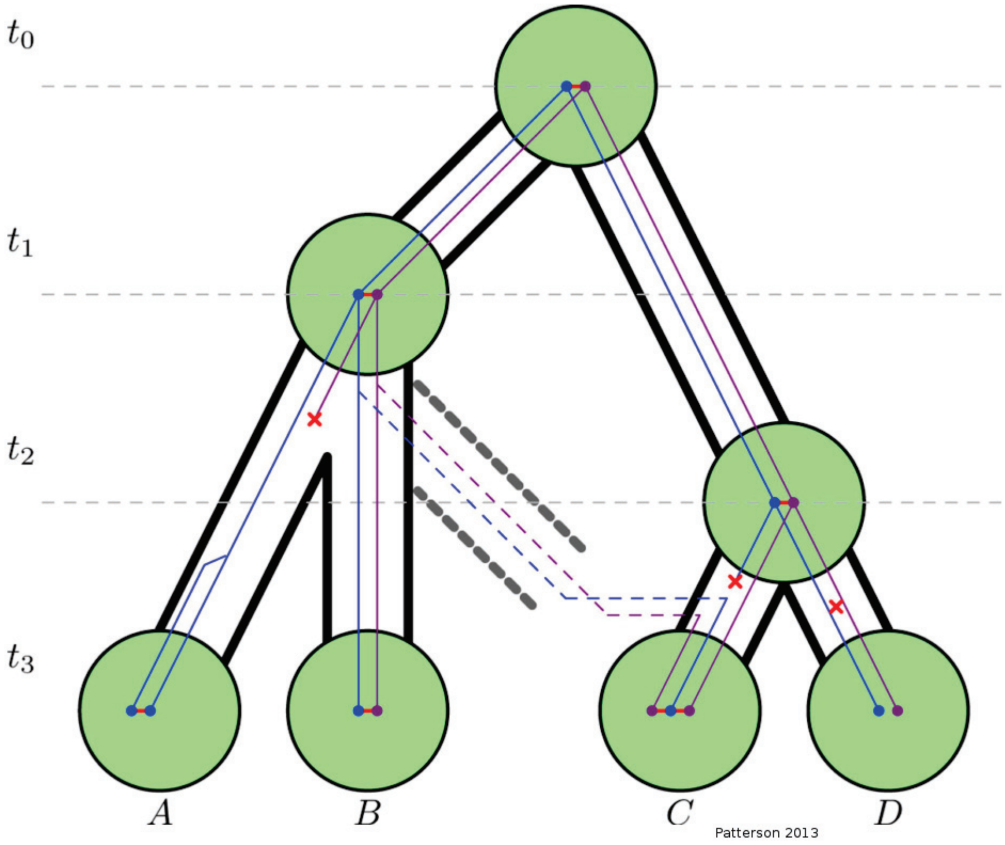


Figure 1.27: Figure 1 from [Patterson *et al.*, 2013] showing a species tree (bigger tree with green nodes), two reconciled gene trees (blue and purple) with duplications, losses and transfer (dashed branches), and adjacencies evolving between the genes (red lines). Notice that the adjacency is conserved along the transfer of the two genes.

Originally developed in a DL context, this method have been adapted to a DTL context [Patterson *et al.*, 2013] as well a several other extensions which I will detail in the following chapter. A probabilistic version of the DeCo algorithm has also been published, but it only allows events of Duplication and Loss [Semeria *et al.*, 2015] and also possesses other methodological and computational limits.

As I mentioned, gene to gene co-evolution appears to be an important factor in the determination of individual genes evolution. As such, the co-evolutionary signal between genes is a useful source of information for the annotation of genes. Methods have been designed to detect the co-evolution between genes (the *congruence* methods). Other methods have been developed to describe this co-evolution, in particular the history of this co-evolution. However, few methods have been proposed

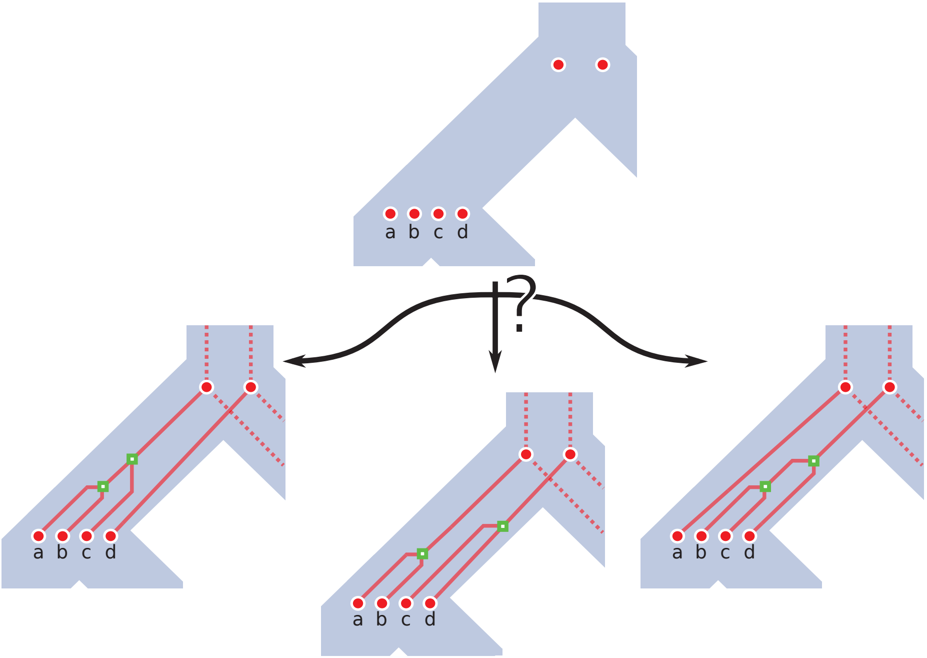


Figure 1.28: Profiles (top) correspond to different possible gene phylogenies/reconciliations (bottom). Here, just along a single branch of the tree, many different topologies (each with its own reconciliation) may correspond to the top profile (I only include 3 here).

to use this relationship as a source of information in the reconstruction of the gene phylogeny. Methods like Wu *et al.* [2012] offer a way to infer ancestral gene content, but this is only partially connected to the phylogeny problem (see Figure 1.28). Other methods use it in the form of the conservation of neighbourhood relationship between genomes (*synteny*) to aid homology search and discriminate genes that have diverged from a speciation (those genes are called *orthologs*) or a duplication (those genes are called *paralogs*) [Wapinski *et al.*, 2007] . Finally, Lafond *et al.* [2013] goes further and propose two methods that exploit information obtained from the study of extant and ancestral gene physical adjacencies along a chromosome (using DeCo for ancestral adjacency inference [Bérard *et al.*, 2012]) to correct a gene phylogeny and reconciliation. The first method uses knowledge, obtained via synteny, that some gene tree nodes should be speciations. The second use the idea that a chromosome is linear and so that a given gene should have exactly 2 (1 if it is a chromosome extremity) neighbours; thus the inference of more than 2 adjacencies for a gene is used as a marker that the tree should be changed. Both methods are illustrated in Figure 1.29.

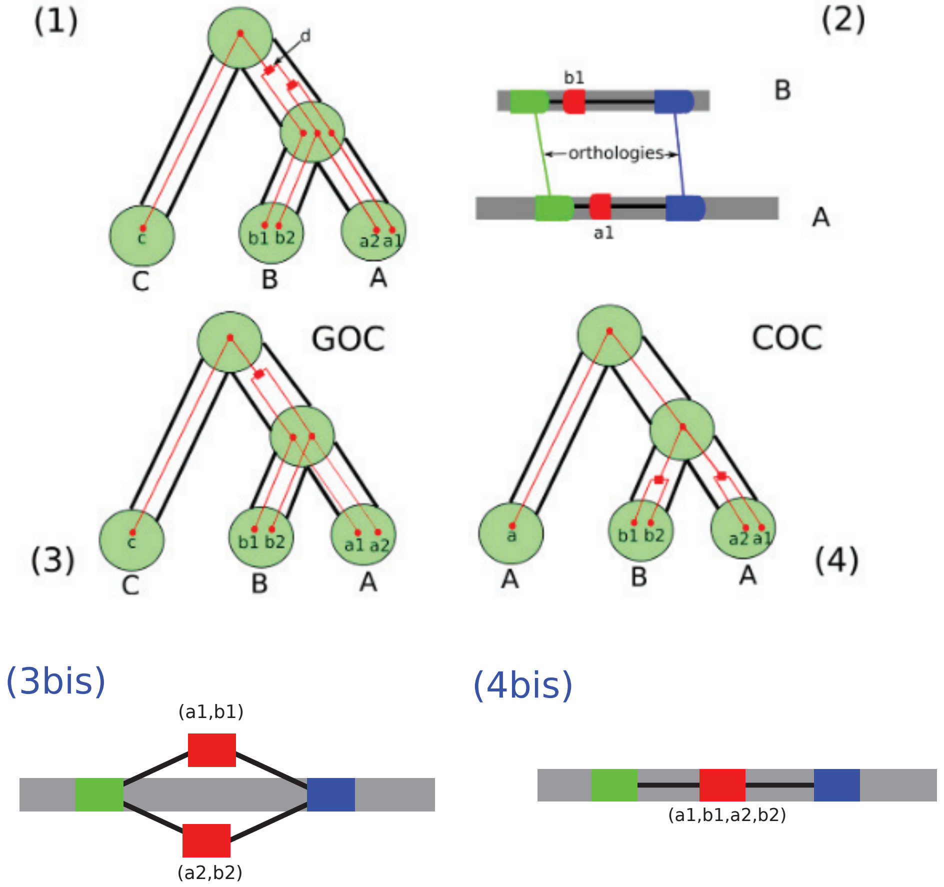


Figure 1.29: Figure 1 from Lafond *et al.* [2013] with added sub-figures 3bis and 4bis. **(1)** A gene tree (the “initial tree”) for the gene family c, $b1$,$b2$,$a1$,$a2$ is shown with small red nodes and single thin red edges. It is reconciled with the phylogeny of the three species A, B and C shown with large green nodes and hollow edges represented by a pair of parallel black lines. Duplication nodes of the reconciled gene tree are squared, while speciation nodes and leaves are dots. **(2)** The two neighbours of $b1$ on genome B and of $a1$ on genome A are inferred to be orthologous according to their lowest common ancestor in their respective gene trees (not shown). This is an argument for inferring orthology between $b1$ and $a1$, which is in contradiction with the information provided by the initial tree: their lowest common ancestor is a duplication, and thus they are inferred to be paralogous. **(3)** A solution that respects the inferred orthology between nodes, that is a gene tree of minimum Robinson-Foulds distance with the initial tree verifying the constraint of $b1$ and $a1$ being orthologous. **(3bis)** The ancestral adjacencies inferred with the reconciliation shown in (3). The green and blue genes have adjacencies both to the ancestor of $a1$ and $b1$ (noted $(a1, b1)$), and to the ancestor of $a2$ and $b2$ (noted $(a2, b2)$), which form a non-linear structure. **(4)** A solution that respects the linearity of ancestral adjacencies, that is a reconciled tree in which the clade $\{b1, b2, a1, a2\}$ of d in the initial tree is rather rooted by a speciation node in the corrected tree. This is an example where the optimal solutions to the two problems differ. **(4bis)** The ancestral adjacencies inferred with the reconciliation shown in (4), where the constraint of linearity is now respected.

1.2.4 Conclusion on the co-evolutions

In this section I presented three forms of co-evolution for the gene (represented in Figure 1.30):

- the co-evolution between the gene and its constituent nucleotides
- the co-evolution between the gene and its species
- the co-evolution between the gene and other genes

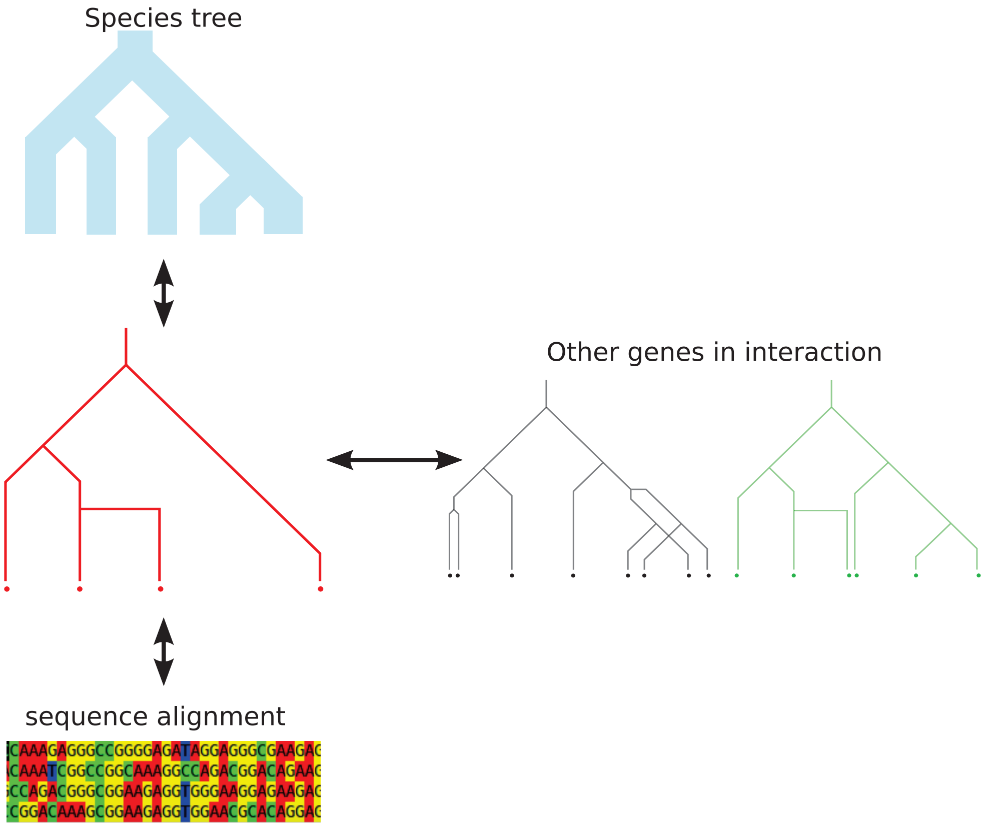


Figure 1.30: A gene family (represented by its gene tree, in red) co-evolves with three kind of entities. Its sequence alignment, the species tree, other gene families.

In the two first forms, we can speak about a *hierarchical* form of co-evolution, as it relates to the co-evolution between two nested objects (the nucleotides in the genes; the genes in the species). The third form of co-evolution can then be described as *non-hierarchical* as it corresponds to a relationship between two objects of the same nature (two genes). Note also that this representation separates the gene from its alignment (the nucleotides it is composed of), and assume a vision where it effectively is an entity that is defined by its relationships (including the ones with its own constituents).

Throughout this document, I generally refer to co-evolving *genes*, composed of *nucleotides* and evolving in *genomes/species*, but these ideas can be applied to different biological objects, and can thus be generalized. *Genes* can become a "character of interest", *genomes* become a "container" (of the character of interest) and *nucleotides* become *residue* (as the base atom of evolution). A last component of this generalized framework is the link between the different co-evolving characters of interest, which I call *adjacencies* (following the convention I used in the precedent section). To different evolutionary biology problems will correspond different definitions of the nature of the character of interest, container, residue and adjacencies, which I illustrate in Table 1.2 and through different examples throughout this document (the case of gene order evolution, as well as protein domain organization evolution, have already been evoked). Additionally, the nature of the character and adjacencies drives the hypotheses that we can make on the adjacencies evolutionary processes and the expectation we have toward the *adjacency graphs* (a graph where characters are nodes and adjacencies are edges between the characters). For instance, in the context of gene order we can expect that a given gene should not have more than two neighbours in a species (as chromosomes are linear or circular) and thus that there cannot be more than two adjacencies that links it to other genes. Such an assumption does not work in another context such as the study of protein interaction as there is no particular restriction on the number of interactions a protein can have.

However for clarity purpose (and without loss of generality) I will continue to use the terms of *genes*, *nucleotides* and *genomes* (or often *species*).

In a classical phylogeny analysis, it can appear that the different co-evolutionary relationships are not treated equally: individual gene trees are usually reconstructed from their nucleotide alignments alone , which means that they are reconstructed independently from the species history and the other genes they co-evolved with. In other terms, the first form of co-evolution is the only one that is taken into account for the reconstruction of the individual gene phylogeny.

Through homology search and the definition of genes, a decision is made that the nucleotides that are in the same gene are expected to have exactly the same tree (*i.e.*, we make the hypothesis that they co-evolve completely) and that the nucleotides

Evolutionary biology question	Character of interest	Residue	Container	Adjacency
chromosomal organization	non-recombining chromosomal regions	nucleotide	species	physical neighbourhood along a chromosome
gene order	gene	nucleotide / AA	species	physical neighbourhood along a chromosome
protein domain organization	protein domain	"	"	physical neighbourhood along a protein
gene fusion	gene	"	"	fusioned state
metabolism	enzyme	"	"	similar substract / product
Protein Interaction Network	protein	"	"	protein interaction
Ecosystems	species	genes	ecosystems	interacting species (predation/symbiose)
Microbiomes	strains	functions	hosts	interacting functions

Table 1.2: Different possible definitions of the described terms, implied by different problems.

from two different genes are not expected to show any form of congruence between their trees (*i.e.*, we make the hypothesis that they evolve independently: their co-evolution is null); see Figure 1.31 for an illustration of these hypotheses.

These two hypotheses are implicit to phylogeny inference procedures from an alignment. They are necessary as considering all the possible co-evolutionary partners of a gene when trying to build its phylogeny would render the task computationally intractable, even barring the idea that the phylogenies of these partners should be also be built using the one we are trying to infer, thus implying the need for some sort of simultaneous reconstruction of all gene trees together.

We should however remember that these hypotheses are made and that they do not reflect the biological reality where we expect a level of congruence between the trees of co-evolving genes, and between the tree of a gene and the one of the species it evolves in (as illustrated in Figure 1.32).

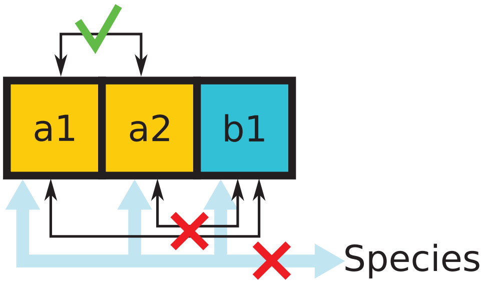


Figure 1.31: Three (neighbouring) nucleotides (a1,a2 and b1), coloured according to the gene they belong to (a1 and a2 are part of the same gene and b1 is in a different one). Co-evolutionary relationships between the nucleotides are shown with black double arrows. Co-evolutionary relationships between the nucleotides and their species are shown with blue double arrows. The co-evolutionary relationships are annotated with a green tick if they are part of the classical phylogeny, and with a red cross if they are not. Classical phylogeny forces the hypothesis that the different parts of the same gene are in total co-evolution, while there is no co-evolution with anything else (other part of other genes, or the species they evolve in)

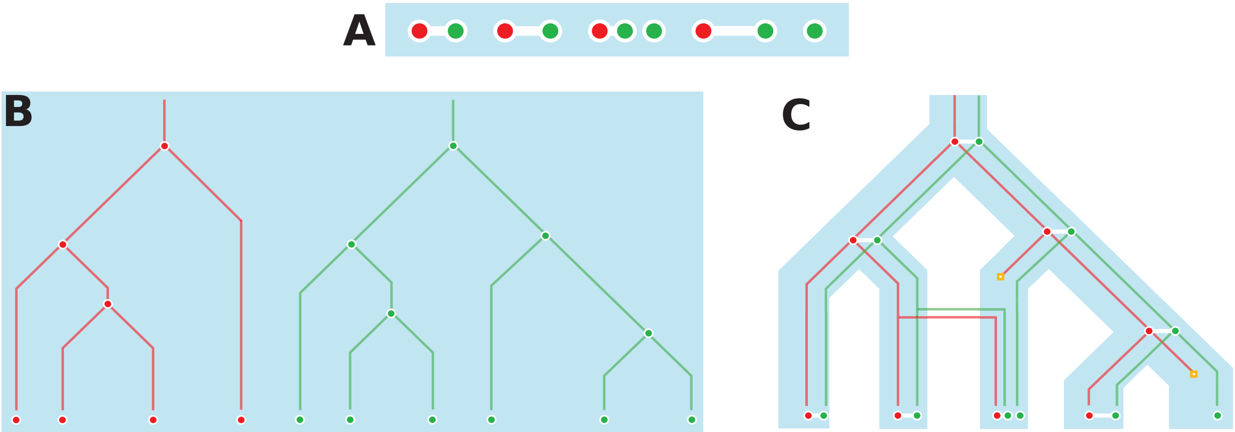


Figure 1.32: **A** observed genes for two gene families (the red and the green one), with adjacencies between them (indicating, for instance, a physical interaction). **B** Classical phylogeny: each gene family is reconstructed independently from the others, only based on sequence information. **C** A more integrated view where gene families evolve along the same species tree and sometimes share additional events due to their co-evolutionary relationship.

A considerable amount of effort has been spent on the optimization of methods using the first form of co-evolution to infer phylogenies (Minh *et al.* [2013]; Stamatakis [2014]; Nguyen *et al.* [2015] to only cite a few recent ones) while only a handful of methods use information from the second form of co-evolution to infer gene phylogenies (often jointly with the first form, or as an *a posteriori* correction/refinement; see the end of the reconciliation section for more details and citations) and even less use information from the third form of co-evolution (I have already mentioned the gene tree correction techniques of Lafond *et al.* [2013]).

However it is known that nucleotide-based gene tree inference is not without its issues and the resulting tree may be erroneous and/or suffer from a high level of uncertainty (*eg.*, have low support values) either because of artefacts from the methods themselves or because of a lack of information in the alignement. Gene tree correction techniques using information from other form of co-evolution (for instance, Wu *et al.* [2013]; Lafond *et al.* [2013]) have proven to be useful to get *better gene trees*, where a *better gene tree* is often defined as a tree whose support from the alignment is close to the support of the optimal tree inferred using the alignment alone (often called the *initial tree*) and that shows a better agreement with its species tree (*e.g.*, have a reconciliation with less events of duplication, loss or transfer) or with the trees of the other genes it interacts with (*e.g.*, it allows the inference of linear ancestral chromosomes).

To my knowledge, no method exists that takes into account information coming from the three forms of co-evolution simultaneously.

1.3 Work accomplished

The rest of this document is organized along two main chapters.

Chapter 2 details my work on adjacencies and their histories (where adjacencies represent the links between co-evolving genes) in a context where individual gene trees and reconciliations are considered fixed. In particular, the chapter presents two articles that I published. The first focuses on the inference of ancestral adjacencies given extant ones and reconciled gene trees. The second corresponds to an application of ancestral adjacency inference on a genome scale, highlights limitations arising from the use of fixed individual genes reconciliations and tree topologies that were inferred independently from each other and proposes ways to design solutions to the problems linked to these limitations.

Chapter 3 builds on the comments made in chapter 2 and describes a method that goes beyond the traditional phylogenetic hypothesis of independence between genes. It does so by describing a metric that accounts for all three described forms of co-evolution (co-evolution with the sequence, with the species, with other genes): a *global score*. This global score is designed to favour cases where an individual gene tree may be sub-optimal with regards to its alignment sequence alone (first form of co-evolution) but where this sub-optimality is compensated by a good fit to the species tree (assessed via the gene reconciliation) and a good fit to gene with whom it co-evolves (assessed via adjacencies). A method for the computation of the global score is proposed, as well as strategies to optimize it. Lastly, these methods are applied to mammalian and fungal data-sets where I show that the proposed method can indeed lead to lower values of the global score. In addition, I observe that instances with a lower global score exhibit longer chromosomal evolutionary events (loss, duplication or transfer) in terms of the number of genes they encompass. In the case of the mammalian data-set this is also coupled with an amelioration of the inferred ancestral genomes and more parsimonious adjacency histories (*i.e.*, ones that require less events of adjacency gains and breakages to be explained).

Chapter 2

Inference of adjacencies on fixed reconciliations and topologies

2.1 A software to infer adjacencies histories: DeCo*

2.1.1 The DeCo family of algorithms

The article presented in this section describes a software, called DeCoSTAR, whose aim is to reconstruct the evolutionary history of links between genes of the same species: *adjacencies*.

Adjacency histories form a good basis for the study of the co-evolutionary relationship between genes because an adjacency history displays events telling a conjoined story of the two genes, the evolutionary events they share and the ones they do not. Additionally, an adjacency history exists only when the two sides of the adjacency are present together in a species (at the same time).

Adjacencies can, for instance, model relationships between genes such as neighbourhood on a chromosome, or between domains along a protein. They can also symbolize the fusionned state of two genes (the absence of adjacency meaning that the two genes are transcribed independently)[1].

DeCoSTAR consists in a re-implementation, generalization and extension of a class of software that I mentioned in the introduction: the DeCo-like algorithms.

A probabilistic version of the DeCo algorithm have also been published [Semeria *et al.*, 2015] , however it is fairly distant from what I did from a methodological point

[1] These three use cases are demonstrated in the article.

of view and suffers from several limitations. Aside from being currently limited to events of duplication and loss, it cannot consider cases where more than one duplication occurred between two speciations and has to perform heavy computations to accounts for branch lengths. For these reasons the algorithm of Semeria *et al.* [2015] has not yet been implemented in DeCoSTAR. So more more precisely, DeCoSTAR includes the DeCo-like algorithms that use a parsimony framework.

The original DeCo publication of 2012 [Bérard *et al.*, 2012] proposes a model of evolution of adjacencies in a DL framework. Given a species tree, a set of gene trees and a set of adjacencies between extant genes (extant adjacencies), DeCo 1) computes the (DL) reconciliation of each gene tree 2) groups the adjacencies together by homology 3) computes the histories of homologous adjacencies.

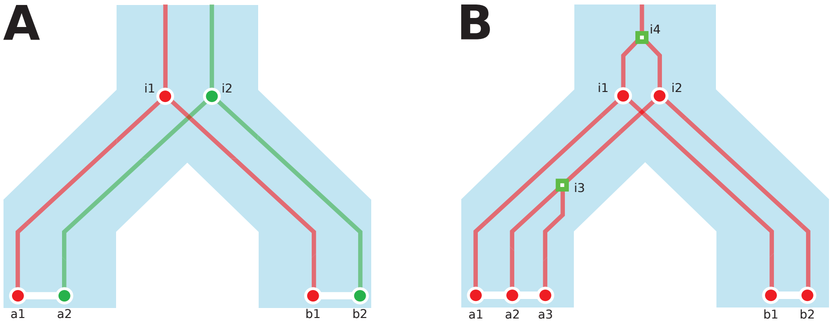


Figure 2.1: Extant adjacencies (white lines) between reconciled gene trees can be grouped by homology. **A** Adjacencies between different gene trees; the adjacency $a1 - a2$ (between gene $a1$ and $a2$) is homologous to the adjacency $b1 - b2$. **B** Adjacencies within the same gene tree; The adjacencies $a1 - a2$ and $b1 - b2$ are homologous, but they are not homologous to the adjacency $a2 - a3$.

This introduces the notion of *homology between adjacencies*. Two adjacencies $a1 - a2$ (linking gene $a1$ to gene $a2$) and $b1 - b2$ are homologous if $a1$ (respectively $a2$) is in the same gene tree as $b1$ (resp. $b2$) and there exists a common ancestor of $a1$ and $b1$ named $i1$ and a common ancestor of $a2$ and $b2$ named $i2$ such that $i1$ and $i2$ are in the same species (for instance, see Figure 2.1 **A**). Additionally, if $a1$, $a2$, $b1$ and $b2$ are all in the same gene tree, then the LCA of $a1$ and $a2$ must be the same node as the LCA of $b1$ an $b2$. This is illustrated in Figure 2.1 **B** where the adjacencies $a1 - a2$ and $b1 - b2$ respect this condition, but not the adjacencies $b1 - b2$ and $a2 - a3$ (the LCA of $a2 - a3$ is $i3$ and is different from the LCA of $b1 - b2$: $i4$). As any homology relationship, homology between adjacencies is transitive (if a is homologous to b and c, then b and c are homologous) and symmetric (if a is homologous to b, then b is homologous to a). I call a group of homologous adjacencies an *equivalence class*.

Defining homology between adjacencies justifies that we describe them as part of the same history where we precise the presence or absence of ancestral adjacencies (adjacencies between internal nodes of the gene tree) as well as events of adjacency gains (appearance of an adjacency) and adjacency breakages (disappearance of an adjacency).

The reconciliations and adjacency histories are computed following a parsimony principle that minimizes a score depending linearly on the number of events (gene duplication and loss for the reconciliations, adjacency gains and breakage for adjacency histories).

Maximum Parsimony

Maximum Parsimony refers to a framework where one wants to find the solution that implies the **minimum number of events**. This criterion is often used in the context where rather than focusing on minimizing a number of event, one minimizes a **score** that is a linear combination of the number of events. In this context, each type of event is assigned a **cost** (and minimizing the number of events becomes the particular case where each type of event has the same cost).

The definition of the costs associated with each event is of crucial importance as they will determine which solution will be favoured.

Value of these costs may for instance be fixed using knowledge about the modelled objects gleaned from the bibliography (as is the case in the DeCoSTAR article in the fusion/fission article) or by adapting ML estimation of event rates (going from rates to costs can be done using a $-log$ operation). Alternatively, several runs of the parsimonious inference, using each time different costs, can be done to either assess the robustness of the results or to look for a set of *good* costs (such as in the heuristic of [Scornavacca *et al.*, 2014]).

To use maximum parsimony can be viewed as making the choice that the most simple explanation is the best one (because you prefer solutions implying less costly events). While this principle is often used in science, it may not always apply very well to the modelling of evolution. In particular, multiple mutations, back mutations or convergent mutations typically will not be detected under parsimony assumption and thus lead to an underestimation of the number of events that occurred.

There is however an argument to be made that the design of parsimony-based algorithms may be easier than likelihood-based ones. Indeed for many problems, including sequence alignment, alignment-based phylogeny, reconciliation and adjacency histories inferences, a maximum parsimony algorithm was devised before probabilistic models and the methods to compute their likelihood were proposed.

Several algorithms have been published that extend the original algorithm.

- Patterson *et al.* [2013] added the possibility to compute adjacency histories in the presence of transfers, but did so at the cost of the reconciliation computation (this is mostly due to the fact that reconciling a gene tree and a species tree in the presence of HGT is more complicated than in its absence; additionally this gives the user the possibility to input the reconciliation of its choice, which was not previously possible).

- Chauve *et al.* [2015] introduced a method to transform the original DeCo formulas in order to be able to sample adjacency histories according to their score (meaning that an adjacency history with a better score than another will be sampled more often). This constitutes an important step in the development of the DeCo-like approaches as it allows to go beyond one of the major drawback of parsimony (which only yields the adjacency histories with the better score).

- Anselmetti *et al.* [2015] adapted DeCo to a context of genome assembly, where the adjacencies between certain extant genes are unknown and we want to infer them. The introduced modifications effectively allow the software to infer new extant adjacencies when they are supported by other, homologous, extant adjacencies.

Each of these extensions was done independently, meaning, for instance, that the possibility to sample adjacency histories according to their score or inferring new extant adjacencies was not possible in a DTL framework, DeCoSTAR remedy this and integrates all of these extensions in the same software[2]. Furthermore, I also extended the existing methods in many ways. Among those extensions is the possibility to account for adjacencies evolving between gene extremities (*e.g.*, genes start and stop) rather than between genes, the possibility to sample uniformly across parsimonious scenarios (rather than output an arbitrary one, as was the case before) and also various additional options related to the tree input (which may now be indifferently be rooted, unrooted, correspond to tree distribution or already reconciled gene trees). Also last but not least, the combination of certain options (in particular the addition lateral gene transfers from Patterson *et al.* [2013] and the Boltzmann-sampling from Chauve *et al.* [2015]) prompted a full re-write of the recurrence formulas (used to compute the adjacency histories cost) which were generalized.

As I would like to be able to jointly consider the information coming from all three forms of co-evolution (cf. the conclusion on co-evolution in the introduction) I integrated the adjacencies histories inference methods (third form of co-evolution) in the ecceTERA [Jacox *et al.*, 2016] package that is able to infer a gene tree / species tree reconciliation that jointly considers sequence and reconciliation information (first and second forms of co-evolution) thanks to the TERA algorithm [Scornavacca *et al.*, 2014], already mentioned in the introduction and further explained in a latter chapter. This integration is one of the main factors behind the flexibility of DeCoSTAR in terms of input.

2.1.2 DeCoSTAR: an integration of DeCo-like algorithms in ecceTERA

[2] It is named as it in reference to a regular expression of the names of the programs it regroups (DeCo, DeCoLT , art-DeCo, DeClone). This expression would be something like `*DeCl?o.*` , simplified `DeCo*`: DeCoSTAR

DeCoSTAR: Reconstructing the Ancestral Organization of Genes or Genomes Using Reconciled Phylogenies

Wandrille Duchemin[1,2,*], Yoann Anselmetti[2,3], Murray Patterson[2,4], Yann Ponty[5,6], Sèverine Bérard[3,7], Cedric Chauve[8], Celine Scornavacca[3], Vincent Daubin[2], and Eric Tannier[1,2]

[1]Inria Grenoble Rhône-Alpes, Montbonnot, France

[2]Université de Lyon, Université Lyon 1, CNRS, Laboratoire de Biométrie et Biologie Évolutive UMR5558, Villeurbanne, France

[3]Institut des Sciences de l'Évolution, Université de Montpellier, CNRS, IRD, EPHE, Montpellier, France

[4]Experimental Algorithmics Lab (AlgoLab), Dipartimento di Informatica, Sistemistica e Comunicazione (DISCo), Università degli Studi di Milano-Bicocca, Viale Sarca, Milano, Italy

[5]CNRS, Ecole Polytechnique, LIX UMR7161, Palaiseau, France

[6]Inria Saclay, EP AMIB, Palaiseau, France

[7]LIRMM, Université de Montpellier, CNRS, Montpellier, France

[8]Department of Mathematics, Simon Fraser University, Burnaby, British Columbia, Canada

*Corresponding author: E-mail: wandrille.duchemin@univ-lyon1.fr.

Accepted: April 7, 2017

Abstract

DeCoSTAR is a software that aims at reconstructing the organization of ancestral genes or genomes in the form of sets of neighborhood relations (adjacencies) between pairs of ancestral genes or gene domains. It can also improve the assembly of fragmented genomes by proposing evolutionary-induced adjacencies between scaffolding fragments. Ancestral genes or domains are deduced from reconciled phylogenetic trees under an evolutionary model that considers gains, losses, speciations, duplications, and transfers as possible events for gene evolution. Reconciliations are either given as input or computed with the ecceTERA package, into which DeCoSTAR is integrated. DeCoSTAR computes adjacency evolutionary scenarios using a scoring scheme based on a weighted sum of adjacency gains and breakages. Solutions, both optimal and near-optimal, are sampled according to the Boltzmann–Gibbs distribution centered around parsimonious solutions, and statistical supports on ancestral and extant adjacencies are provided. DeCoSTAR supports the features of previously contributed tools that reconstruct ancestral adjacencies, namely DeCo, DeCoLT, ART-DeCo, and DeClone. In a few minutes, DeCoSTAR can reconstruct the evolutionary history of domains inside genes, of gene fusion and fission events, or of gene order along chromosomes, for large data sets including dozens of whole genomes from all kingdoms of life. We illustrate the potential of DeCoSTAR with several applications: ancestral reconstruction of gene orders for *Anopheles* mosquito genomes, multidomain proteins in *Drosophila*, and gene fusion and fission detection in *Actinobacteria*.
Availability: http://pbil.univ-lyon1.fr/software/DeCoSTAR (Last accessed April 24, 2017).

Key words: gene order, software, reconciliation, protein domain, evolution, gene fusion/fission, rearrangements.

Introduction

Colocalization of genes along a chromosome, or combinations of domains within a gene are genomic features that evolve and can be gained or broken by rearrangements. We will use the term *gene* to designate an evolutionary unit (a gene or a domain or any smaller or larger module), and we call *adjacency* the link between two genes. An adjacency thus represents either the link between two contiguous genes on a chromosome, or the link between two domains of a protein, or may also represent the link between two genes fused into a single gene. The evolution of adjacencies is usually modeled differently for different scales (Pasek et al. 2006; Ma et al. 2006; Wu et al. 2013; Stolzer et al. 2015), complex gene histories are rarely handled in ancestral organization reconstruction, and models integrating fusions and fissions of genes are called for (Haggerty et al. 2014).

We describe a software, DeCoSTAR, which reconstructs putative ancestral states of adjacencies, for example, ancestral domain structures of a modular protein, as well as chromosome organizations of whole ancestral genomes, or fusion/fission histories or modular genes, when genes have complex histories made of gain, duplication, transfer, speciation, and loss events.

The input of DeCoSTAR consists in a species tree, a set of extant gene families—each in the form of one or several gene trees—and extant adjacencies between pairs of extant genes.

The gene trees and the species tree follow the *reconciliation* framework that is described by Jacox et al. (2016). (Reconciled gene trees are rooted gene trees whose nodes are associated to an evolutionary event, such as speciation, gene loss, gene duplication, or lateral gene transfer, and to a position in the species tree. Numerous methods exist to build reconciliations, see Åkerborg and Sennblad [2009], Bansal et al. [2012], Stolzer et al. [2012], and Szöllosi et al. [2015] for example.) The species tree may be dated or not, and gene families may be provided in the form of a gene tree sample, a single gene tree, or directly a fully reconciled gene tree. Reading direction (orientation) of genes on the chromosome may be given or not. Accordingly, ancestral genes are directed or not in ancestral organizations.

The output consists of adjacencies between ancestral genes along with evolutionary scenarios composed of *gains* and *breakages* of adjacencies. DeCoSTAR optimizes on a linear combination of the number of gains and breakages of adjacencies along the species tree. It can sample among optimal solutions, and thus give a statistical support to each inferred adjacency. It can also sample in the space of suboptimal solutions using a Boltzmann–Gibbs distribution centered on the optimal solutions. As an option, it is possible to propose, based on the adjacencies in other species, adjacencies that are not in the input between extant genes; these new adjacencies can be used to improve the assembly of extant genomes.

Note that input adjacencies depicting the linear organization of chromosomes do not guarantee the same linear organization in ancestral genomes. We provide in the distribution a linearization method (Manuch et al. 2012) to transform the output in a linear organization if needed.

An example of input and output for DeCoSTAR is depicted in figure 1 where the evolution of three gene families linked by some adjacencies is represented: The adjacencies follow the evolutionary path of the genes they link and undergo speciations (fig. 1A), are transferred (fig. 1B), disappear because of a gene loss (fig. 1C) or adjacency breakage (fig. 1D), and are gained (fig. 1E).

Features and Implementation

DeCoSTAR supersedes (with the exception of the ability of DeClone [Chauve et al. 2015] to compute the exact expectation of the frequency of a property of interest using a variant of the inside–outside algorithm) and combines all the features of DeCo (Bérard et al. 2012), DeCoLT (Patterson et al. 2013), DeClone (Chauve et al. 2015), and ART-DeCo (Anselmetti et al. 2015). The generalization of all these methods offers novel capabilities, including the Boltzman–Gibbs sampling of ancestral adjacencies in the presence of transfers from error-prone/partial genome assemblies. The integration with the software package ecceTERA dedicated to reconciliations (Jacox et al. 2016) adds novel features, such as the possibility of taking unrooted gene trees or undated species trees as input. As a novelty, it also fully handles gene orientations whenever available, and provides statistical supports of ancestral adjacencies by sampling among optimal solutions.

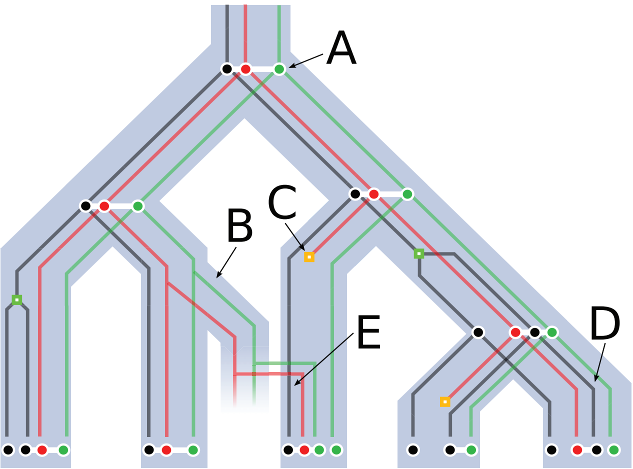


FIG. 1.—A species tree (light blue), three reconciled gene trees (black, red, and green) (losses are orange squares; duplications are green squares) and a set of extant and ancestral adjacencies linking genes (white). (*A*) An adjacency is inherited by both sister species after a speciation occurs. (*B*) An adjacency between the red and green gene is transferred, and so are both extremities of this adjacency. (*C*) The red gene undergoes gene loss and thus both adjacencies it was a part of disappear. (*D*) The adjacency between the red and black genes disappears due to an adjacency break on the branch leading to the leaf. (*E*) An adjacency is gained between the black gene and the newly acquired red gene.

DeCoSTAR is a C++ program requiring the Bio++ library (Gueguen et al. 2013) and the Boost library (BOOST 2003) to be installed. It is a command-line program whose various options and input can be specified on the command line or given in a parameter file. It handles newick format for trees and recPhyloXML (Gence 2016) format for trees and reconciliations.

A detailed documentation of DeCoSTAR options, input and output formats is available in the supplementary material (Supplementary Material online) and is included within the distributed version of the software.

Algorithm

Given a set of adjacencies between extant genes, DeCoSTAR partitions it into homologous families. Two adjacencies a_1a_2 and b_1b_2 are homologous if a_1 and b_1, respectively a_2 and b_2, have a common ancestor i_1, respectively i_2, such that i_1 and i_2 are in a different gene tree or, if they are in the same gene tree, one is not an ancestor of the other. This relation is transitive, yielding a partition of the full set of input adjacencies into families.

For each family of homologous adjacencies, a minimal cost adjacency history, that is, a history that minimizes the number of adjacency gains and adjacency breakages weighted by their respective costs, is computed. This is done in a dynamic programming matrix following a generalization of the propagation rules described in Patterson et al. (2013) (see table 1 and below where we introduce the notation we use).

Once the dynamic programming matrix of has been computed, backtracking on this matrix permits to produce an evolutionary history for the family of homologous adjacencies. This history takes the form of ancestral adjacencies (linking ancestral nodes of the gene trees) and the events they undergo. Events may occur to individual genes or to pairs of genes linked by an adjacency, in which case it is called a *coevent*. A coevent implies that the events from two different reconciled gene tree nodes are part of a single event spanning multiple genes.

DeCoSTAR allows multiple backtracks of the dynamic programming matrix in order to form a sample of adjacency histories, either within optimal solutions or according to a probability space defined by a Boltzmann–Gibbs distribution centered on the optimal solutions.

Each propagation rule is translated into a specific term in a dynamic programming equation for the reconstruction of ancestral states. The complete set of rules (19 rules, whose combinations cover all the cases encountered by the algorithm) implemented in DeCoSTAR is the result of a complete rewriting of a combination of rules taken in the previous softwares, aggregating them in more general rules. For comparison DeCoLT (Patterson et al. 2013), a less general algorithm, used a set of 23 rules.

For two genes a and b, we note $c_1(a,b)$ and $c_0(a,b)$ the cost of, respectively, having an adjacency and having no adjacency between a and b. We call a_1 and a_2 (respectively, b_1 and b_2) the children of a (respectively b) (NB: if a [respectively b] only has one child, then a_2 [respectively b_2] does not exist).

We denote by *Gain* the cost of a single adjacency gain. We denote by *Break* the cost of a single adjacency breakage. Two gene tree nodes a and b (from the same gene tree or not) are said to be *comparable* if they are in the same species, if they are in the same time slice when relevant, and if one is not an ancestor of the other. Otherwise they are said to be *incomparable*.

If the events at a and b (deduced from the gene tree/species tree reconciliations) occurred simultaneously (which is only possible if they are comparable), we call them *synchronous*. Otherwise we call them *asynchronous* and have to take into account if the event at a occurred before the one at b or the opposite.

The different formulas of the propagation rules are combinations of different cases where a and b are comparable, synchronous and how many children they have.

The different case formulas are presented in table 1. In the asynchronous cases, only the number of children of the events that happens first (a in the figure) matters.

An exception to these rules occurs in the specific case where a and b or their children are considered to be in an extinct or unsampled lineage of the species tree (Szöllosi et al. 2013). In these specific lineages event of adjacency breaks are not counted in the cost function.

If a and b both are leaves, the score associated to the presence of the adjacency relies on the adjacencies given as an input. If the scaffolding mode is used, then the formulas at the leaves follow the ones described in (Anselmetti et al. 2015), as described in table 2.

If Boltzmann sampling is used, then the formulas undergo the same changes described in Chauve et al. (2015). Namely, every occurrence of the + operator becomes a product, *min*() functions become sums, and any event cost *EventCost* becomes $e^{-\frac{EventCost}{T}}$, where T corresponds to a pseudo-temperature (the higher the temperature, higher the probability for nonparsimonious scenarios to be sampled). The costs between two leaves also follow a similar transformation.

Results

We tested DeCoSTAR on several biological data sets in order to demonstrate its versatility in various contexts. The first example shows a combination of options previously implemented separately: Boltzmann sampling on the adjacencies and the inference of new extant adjacencies in 18 mosquito genomes under an evolutionary model where only duplications and losses are allowed.

The two other data sets show the application of DeCoSTAR in a context different from gene order reconstruction: protein modular architecture evolution, shown on a set of drosophila genes in which we reconstruct ancestral adjacencies between protein domains, and a history of fusions/fissions between bacterial genes in the presence of transfers. Note that such applications where previously discussed (see, e.g., the conclusion of Patterson et al. [2013]), but had never been demonstrated.

Eighteen *Anopheles*

We selected 14,940 gene families in 18 mosquito species from Neafsey et al. (2015). Gene trees were constructed with RAxML (Stamatakis 2014) and corrected with ProfileNJ (Nouhati et al. 2016) (keeping all branches with a 100%

Table 1
Description of the Propagation Rules under Different Situations

	Synchronous	Asynchronous (a before b)
a has two children b has one child	$c_{0SYNCH}(a,b) = min($ $c_0(a_1,b_1) + c_0(a_2,b_1)$, $c_1(a_1,b_1) + c_0(a_2,b_1) + Gain,$ $c_0(a_1,b_1) + c_1(a_2,b_1) + Gain,$ $c_1(a_1,b_1) + c_1(a_2,b_1) + 2 * Gain)$ $c_{1SYNCH}(a,b) = min($ $c_0(a_1,b_1) + c_0(a_2,b_1) + Break,$ $c_1(a_1,b_1) + c_0(a_2,b_1),$ $c_0(a_1,b_1) + c_1(a_2,b_1),$ $c_1(a_1,b_1) + c_1(a_2,b_1) + Gain)$	$c_{0ASYNCH}(a,b) = min($ $c_0(a_1,b) + c_0(a_2,b),$ $c_1(a_1,b) + c_0(a_2,b) + Gain,$ $c_0(a_1,b) + c_1(a_2,b) + Gain,$ $c_1(a_1,b) + c_1(a_2,b) + Gain)$ $c_{1ASYNCH}(a,b) = min($ $c_0(a_1,b) + c_0(a_2,b) + Break,$ $c_1(a_1,b) + c_0(a_2,b),$ $c_0(a_1,b) + c_1(a_2,b),$ $c_1(a_1,b) + c_1(a_2,b) + Gain)$
a has one child b has one child	$c_{0SYNCH}(a,b) = min($ $c_0(a_1,b_1),$ $c_1(a_1,b_1) + Gain)$ $c_{1SYNCH}(a,b) = min($ $c_0(a_1,b_1) + Break,$ $c_1(a_1,b_1))$	$c_{0ASYNCH}(a,b) = min($ $c_0(a_1,b),$ $c_1(a_1,b) + Gain)$ $c_{1ASYNCH}(a,b) = min($ $c_0(a_1,b) + Break,$ $c_1(a_1,b))$
a has two children b has two children	$c_{0SYNCH}(a,b) = min($ $c_0(a_1,b_1) + c_0(a_2,b_1) + c_0(a_1,b_2) + c_0(a_2,b_2)$, $c_1(a_1,b_1) + c_0(a_2,b_1) + c_0(a_1,b_2) + c_0(a_2,b_2) + 1 * Gain$, $c_0(a_1,b_1) + c_1(a_2,b_1) + c_0(a_1,b_2) + c_0(a_2,b_2) + 1 * Gain$, $c_0(a_1,b_1) + c_0(a_2,b_1) + c_1(a_1,b_2) + c_0(a_2,b_2) + 1 * Gain$, $c_0(a_1,b_1) + c_0(a_2,b_1) + c_0(a_1,b_2) + c_1(a_2,b_2) + 1 * Gain$, $c_1(a_1,b_1) + c_1(a_2,b_1) + c_0(a_1,b_2) + c_0(a_2,b_2) + 2 * Gain$, $c_1(a_1,b_1) + c_0(a_2,b_1) + c_1(a_1,b_2) + c_0(a_2,b_2) + 2 * Gain$, $c_1(a_1,b_1) + c_0(a_2,b_1) + c_0(a_1,b_2) + c_1(a_2,b_2) + 2 * Gain$, $c_0(a_1,b_1) + c_1(a_2,b_1) + c_1(a_1,b_2) + c_0(a_2,b_2) + 2 * Gain$, $c_0(a_1,b_1) + c_1(a_2,b_1) + c_0(a_1,b_2) + c_1(a_2,b_2) + 2 * Gain$, $c_0(a_1,b_1) + c_0(a_2,b_1) + c_1(a_1,b_2) + c_1(a_2,b_2) + 2 * Gain$, $c_0(a_1,b_1) + c_1(a_2,b_1) + c_1(a_1,b_2) + c_1(a_2,b_2) + 3 * Gain$, $c_1(a_1,b_1) + c_0(a_2,b_1) + c_1(a_1,b_2) + c_1(a_2,b_2) + 3 * Gain$, $c_1(a_1,b_1) + c_1(a_2,b_1) + c_0(a_1,b_2) + c_1(a_2,b_2) + 3 * Gain$, $c_1(a_1,b_1) + c_1(a_2,b_1) + c_1(a_1,b_2) + c_0(a_2,b_2) + 3 * Gain$, $c_1(a_1,b_1) + c_1(a_2,b_1) + c_1(a_1,b_2) + c_1(a_2,b_2) + 4 * Gain$) $c_{1SYNCH}(a,b) = min($ $c_0(a_1,b_1) + c_0(a_2,b_1) + c_0(a_1,b_2) + c_0(a_2,b_2) + 2 * Break$, $c_1(a_1,b_1) + c_0(a_2,b_1) + c_0(a_1,b_2) + c_0(a_2,b_2) + 1 * Break$, $c_0(a_1,b_1) + c_1(a_2,b_1) + c_0(a_1,b_2) + c_0(a_2,b_2) + 1 * Break$, $c_0(a_1,b_1) + c_0(a_2,b_1) + c_1(a_1,b_2) + c_0(a_2,b_2) + 1 * Break$, $c_0(a_1,b_1) + c_0(a_2,b_1) + c_0(a_1,b_2) + c_1(a_2,b_2) + 1 * Break$, $c_1(a_1,b_1) + c_1(a_2,b_1) + c_0(a_1,b_2) + c_0(a_2,b_2) + 1 * Break + 1 * Gain$, $c_1(a_1,b_1) + c_0(a_2,b_1) + c_1(a_1,b_2) + c_0(a_2,b_2) + 1 * Break + 1 * Gain$, $c_1(a_1,b_1) + c_0(a_2,b_1) + c_0(a_1,b_2) + c_1(a_2,b_2)$, $c_0(a_1,b_1) + c_1(a_2,b_1) + c_1(a_1,b_2) + c_0(a_2,b_2)$, $c_0(a_1,b_1) + c_1(a_2,b_1) + c_0(a_1,b_2) + c_1(a_2,b_2) + 1 * Break + 1 * Gain$, $c_0(a_1,b_1) + c_0(a_2,b_1) + c_1(a_1,b_2) + c_1(a_2,b_2) + 1 * Break + 1 * Gain$, $c_0(a_1,b_1) + c_1(a_2,b_1) + c_1(a_1,b_2) + c_1(a_2,b_2) + 1 * Gain$, $c_1(a_1,b_1) + c_0(a_2,b_1) + c_1(a_1,b_2) + c_1(a_2,b_2) + 1 * Gain$, $c_1(a_1,b_1) + c_1(a_2,b_1) + c_0(a_1,b_2) + c_1(a_2,b_2) + 1 * Gain$, $c_1(a_1,b_1) + c_1(a_2,b_1) + c_1(a_1,b_2) + c_0(a_2,b_2) + 1 * Gain$, $c_1(a_1,b_1) + c_1(a_2,b_1) + c_1(a_1,b_2) + c_1(a_2,b_2) + 2 * Gain$)	In the case where a and b both are (comparable) **losses**: $c_1(a,b) = 0$ $c_0(a,b) = 0$ In the case where a and b are **incomparable**: $c_1(a,b) = \infty$ (no adjacency for incomparable genes) $c_0(a,b) = min($ $c_{0ASYNCH}(a,b)$, $c_{0ASYNCH}(b,a)$) (a is before b and b before a) In the case where a and b are **comparable**: $c_1(a,b) = min($ $c_{1ASYNCH}(a,b)$, $c_{1ASYNCH}(b,a)$, $c_{1SYNCH}(a,b)$) $c_0(a,b) = min($ $c_{0ASYNCH}(a,b)$, $c_{0ASYNCH}(b,a)$, $c_{0SYNCH}(a,b)$)

Table 2
Description of the Score between Leaves

General formulas : $c_1(a,b) = -\tau * log(p)$
$c_0(a,b) = -\tau * log(1-p)$

	Adjacency given in input		Adjacency absent from input
base mode	$p = 1$ τ does not matter $\Downarrow$ $c_1(a,b) = 0$ $c_0(a,b) = \infty$		$p = 0$ τ does not matter $\Downarrow$ $c_1(a,b) = \infty$ $c_0(a,b) = 0$
score given	$p =$ score given as input $\tau = log(\frac{1}{b})$ $b = 10\ 000$ by default	scaffolding mode	For each species: $p = F_{adj} * BP$ $\tau = \frac{Break}{SPI * log(\frac{1-BP}{BP})})$

$$BP = \frac{\#ctg - \#chr}{2 * \#ctg * (\#ctg - 1)}$$

$\#ctg$: number of contigs
$\#chr$: expected number of chromosomes

F_{adj} : 0, 1, 2 or 4

SPI : Scaffolding Propagation Index

NOTE.—The two parameters F_{adj} and *SPI* used in the scaffolding modes, respectively, account for the position of the genes in their contig and the repartition of poorly assembled genomes in the species tree; both parameters are described with more details in the supplementary material, Supplementary Material online. The *scaffolding mode* and *score given* option can be used simultaneously as they affect a different set of adjacencies: respectively, the adjacencies absent from the input and the adjacencies given in the input.

bootstrap support and correcting the others to minimize duplications and losses in a reconciliation with a species tree).

A sample of 100 solutions was generated according to a Boltzmann distribution with temperature 0.05. As the genomes are not fully assembled, we added the possibility of proposing extant adjacencies (the scaffolding mode). Combining these two options (sampling and extant adjacencies proposition) is a specificity of DeCoSTAR as they were hitherto only available separately.

This treatment provides a comprehensive history of duplications, losses and rearrangements of Anopheles, in addition to novel propositions for the scaffolding of extant genomes: 187,870 ancestral adjacencies and 16,193 new extant adjacencies were generated, all with a posterior probability which corresponds to their frequency in a sample. Figure 2 depicts the connectivity of genes with other genes in the same extant or ancestral species and thus gives an insight on the shape of extant and ancestral genomes in the input and output. In the input (see the black line), most genes have exactly two neighbors with adjacencies weighted 1, but some have one or zero neighbors because of incomplete assemblies. In the output, extant genomes are better scaffolded (less genes with zero or one neighbor, more with two) but ancestral genomes may show some conflict (genes with three neighbors or more) because adjacencies evolved independently in the model.

Fly Protein Domains

DeCoSTAR can also be applied to protein domain architecture. When doing so, gene trees become domain trees, evolving along a species tree. Proteins are not modeled explicitly but are rather formed by groups of domains linked together. Thus, the resolution slightly differs from a similar previous approach proposing to reconcile domain trees with gene trees (Stolzer et al. 2015). For example, the transfers of domains from one gene to another result in a sequence of adjacency gains and breakages, while they were modeled as singular events there. We exhibit an example of such an application on the protein domain families described in Wu et al. (2012). It features 22,867 protein domain families in nine fully sequenced fly genomes. Of these, we kept the 12,906 protein domain families that have at least one extant copy that is part of an extant multidomain protein. Protein domain families were aligned using MUSCLE (Edgar 2004) and their trees were inferred using RAxML (Stamatakis 2014) with the appropriate model (inferred using the RAxML perl helper script for finding the best protein substitution model). The adjacencies used as input reflect neighborhood relationship between domains of the same extant protein.

There are in average 5,278 proteins per extant species in the input data set with an average protein size of 2.030 domains. DeCoSTAR was used to infer ancestral adjacencies forming an average of 4,977 proteins per ancestral species, for an average protein size of 2.188 domains. As with the validation on the Anopheles species data set, some ancestral protein domains have been erroneously inferred with more than two neighbors, leading to the presence of some nonlinear proteins in the ancestral species. Nonlinear proteins should be seen as several linear proteins erroneously linked together. Their presence decreases the total number of proteins and

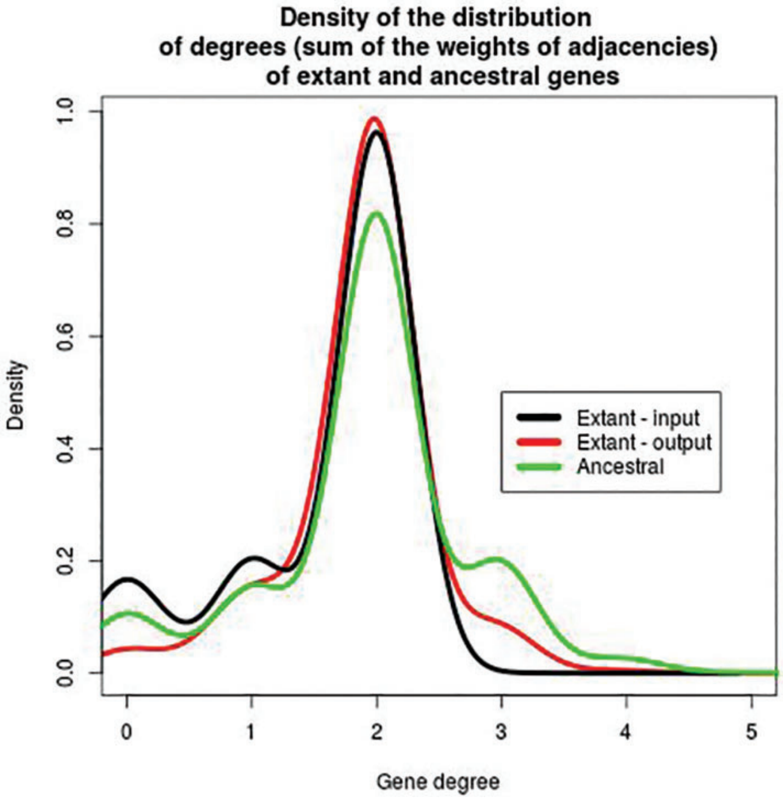


FIG. 2.—Density of the distribution of the degree of all genes inferred by DeCoSTAR on the 18 *Anopheles* data set. The degree of an extant or ancestral gene is the sum of the weights of all adjacencies containing this gene. For extant genomes in the input (black line), this value can only be 0, 1, or 2. For genomes in the output, extant (red line) or ancestral (green line), all values are possible because adjacencies have scores between 0 and 1, and a gene can belong to an arbitrary number of adjacencies. The difference between the black and red lines are due to the scaffolding: genes with 0 or 1 neighbor are linked to other genes as an output of DeCoSTAR. In ancestral genomes, some genes have degree three or slightly more.

increases the average number of protein domains per protein, which would explain the difference in average number of proteins and average protein size between extant and ancestral species.

A Fusion–Fission History in Actinobacteria

Adjacencies can be used to denote the fact that two genes are fused into one. To illustrate this, we use a set of three gene families from the HOGENOM database (Penel et al. 2009) that we, respectively, call *A*, *B*, and *C*. In all *Actinobacteria* present in HOGENOM, the *A* and *B* genes are always present together, but never with *C* genes. Furthermore, in a profile alignment, *A* and *B* both align on disjoint, consecutive regions of *C*, covering nearly 98% of its length. We use this signal as the marker that *A* and *B* genes fused in order to give *C* genes.

To reconstruct the history of this system, we manually cut each *C* gene into its parts that, respectively, aligned with *A* and *B*, added them to the alignment of the family with whom they aligned and put an adjacency between the newly formed gene so that we could account for the fact that they fused.

We used an option of DeCoSTAR that specifies that an adjacency at the root of its history should not be penalized by a gain, as we do not make any assumption about the ancestral fissioned or fusioned state (which is not the case for ancestral genome reconstruction for example, where an adjacency can always be considered as gained in some root branch of the phylogeny). Moreover, we set the event costs so that an adjacency break (corresponding to a fission event), costs four times as much as an adjacency gain (corresponding to a fusion event), following the results of (Kummerfeld and Teichmann 2005; by default, from the gene order context, an adjacency gain costs twice as much as an adjacency break).

The results obtained with DeCoSTAR are represented in figure 3. It exhibits three adjacency gains (represented by an upper *G* on the figure), which correspond to three independent fusion events between gene families *A* and *B*.

Conclusion

There exists an extensive set of bioinformatics tools aiming at reconstructing the history of an evolutionary unit, as a gene or

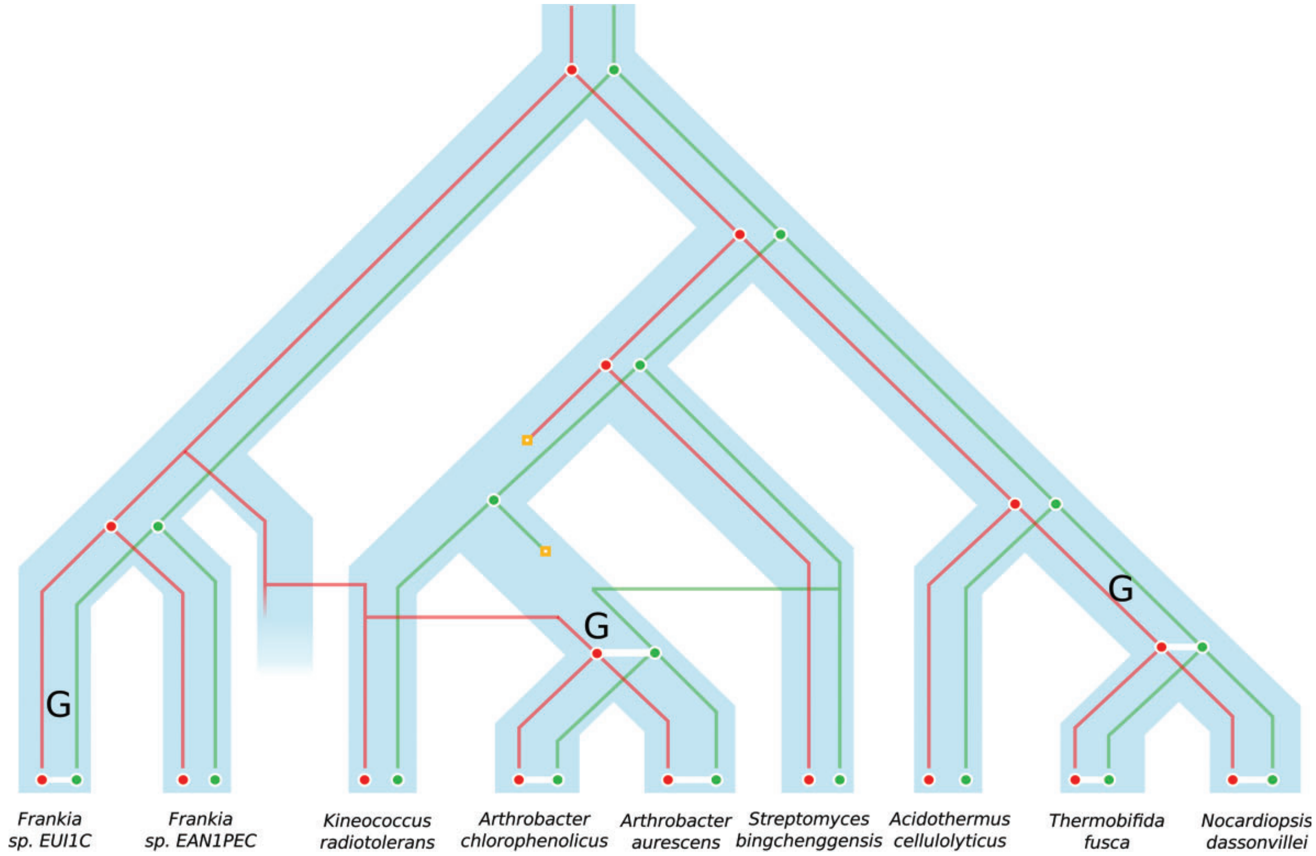

G
G
G
Frankia sp. EUI1C
Frankia sp. EAN1PEC
Kineococcus radiotolerans
Arthrobacter chlorophenolicus
Arthrobacter aurescens
Streptomyces bingchenggensis
Acidothermus cellulolyticus
Thermobifida fusca
Nocardiopsis dassonvillei

FIG. 3.—A schematic representation of the results obtained for the fusion–fission data set, following the schema described in figure 1, and with adjacency gain marked by an upper *G*. Family *A* and *B* are represented as reconciled gene trees, respectively, in red and green. The presence of an adjacency denotes the fusion of *A* and *B* to form the family *C*.

a domain or a gene concatenate. But they all make the assumptions that, inside a unit, all sites have the same history, and that two units are independent. The inter or intra unit organization is rarely modeled, with the effect of missing an evolutionary view on what the living is essentially made of: organization and interaction. Here, we propose to depict this interaction in the form of adjacencies between units, where the units can be genes, gene domains, or parts of genes having different histories like in the case of fusions or fissions. We present a software—DeCoSTAR—that generalizes several algorithms published by our group, is easy to install and to use, allows a wide range of genomic events such as duplications, transfers, losses, rearrangements, and can deal with poorly assembled genomes. We demonstrate the utility of this software on a diverse set of very large biological data sets where taking the interactions between units into account is crucial. We show that a single methodological framework can account for diverse situations which were previously approached separately by *ad-hoc* methods. Up to changes in propagation rules, the same principle can also be used to reconstruct ancestral states of any binary relationship, such as protein interaction, regulation, or coexpression.

Supplementary Material

Supplementary data are available at *Genome Biology and Evolution* online.

Acknowledgments

This work was funded by the Agence Nationale pour la Recherche, Ancestrome project ANR-10-BINF-01-01. This work was performed using the computing facilities of the CC LBBE/PRABI.

Literature Cited

Åkerborg Ö, Sennblad B. 2009. Simultaneous Bayesian gene tree reconstruction and reconciliation analysis. Proc Natl Acad Sci U S A. 106(14):5714–5719.

Anselmetti Y, et al. 2015. Ancestral gene synteny reconstruction improves extant species scaffolding. BMC Genomics 16(Suppl 10): S11.

Bansal MS, Alm EJ, Kellis M. 2012. Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss. Bioinformatics 28(12):283–291.

Bérard S, et al. 2012. Evolution of gene neighborhoods within reconciled phylogenies. Bioinformatics (Oxford, England) 28(18):i382–i388.

BOOST (2003). Boost C ++ libraries. Available from: http://www.boost.org/.

Chauve C, Ponty Y, Zanetti JPP. 2015. Evolution of genes neighborhood within reconciled phylogenies: an ensemble approach. Lecture Notes in Computer Science. Bioinformatics. 16(Suppl 19):S6.

Edgar RC. 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 32(5):1792–1797.

Gence G. (2016). recphyloxml. Available from: http://phylariane.univ-lyon1.fr/recphyloxml/.

Gueguen L, et al. 2013. Bio++: efficient extensible libraries and tools for computational molecular evolution. Mol Biol Evol. 30(8):1745–1750.

Haggerty L, et al. 2014. A pluralistic account of homology: adapting the models to the data. Mol Biol Evol. 31:501–516.

Jacox E, Chauve C, Szöllő si GJ, Ponty Y, Scornavacca C. 2016. ecceTERA: comprehensive gene tree-species tree reconciliation using parsimony. Bioinformatics 32:2056–2058.

Kummerfeld S, Teichmann S. 2005. Relative rates of gene fusion and fission in multi-domain proteins. Trends Genet. 21:35–30.

Ma J, et al. 2006. Reconstructing contiguous regions of an ancestral genome. Genome Res. 16:1557–1565.

Manuch J, Patterson M, Wittler R, Chauve C, Tannier E. 2012. Linearization of ancestral multichromosomal genomes. BMC Bioinformatics 13:S11.

Neafsey DE, et al. 2015. Highly evolvable malaria vectors: the genomes of 16 anopheles mosquitoes. Science 347(6217):1258522.

Nouhati E, et al. 2016. Efficient gene tree correction guided by species and synteny evolution. PLoS One 11(8):e0159559.

Pasek S, Risler J-L, Brézellec P. 2006. Gene fusion/fission is a major contributor to evolution of multi-domain bacterial proteins. Bioinformatics (Oxford, England) 22(12):1418–1423.

Patterson M, Szöllő si G, Daubin V, Tannier E. 2013. Lateral gene transfer, rearrangement, reconciliation. BMC Bioinformatics 14(Suppl 15):S4.

Penel S, et al. 2009. Databases of homologous gene families for comparative genomics. BMC Bioinformatics 10(Suppl 6):S3.

Stamatakis A. 2014. RAxML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies. Bioinformatics 30(9):1312–1313.

Stolzer M, et al. 2012. Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees. Bioinformatics 28(18):409–415.

Stolzer M, Siewert K, Lai H, Xu M, Durand D. 2015. Event inference in multidomain families with phylogenetic reconciliation. BMC Bioinformatics 16(Suppl 14):1–13.

Szöllosi G, Tannier E, Daubin V, Boussau B. 2015. The inference of gene trees with species trees. Syst Biol. 64:42–62.

Szöllosi GJ, Tannier E, Lartillot N, Daubin V. 2013. Lateral gene transfer from the dead. Syst Biol. 62(3):386–397.

Wu Y-C, Rasmussen MD, Kellis M. 2012. Evolution at the subgene level: domain rearrangements in the Drosophila phylogeny. Mol Biol Evol. 29(2):689–705.

Wu Y-C, Rasmussen MD, Bansal MS, Kellis M. 2013. TreeFix: statistically informed gene tree error correction using species trees. Syst Biol. 62(1):110–120.

Associate editor: Mary O'Connell

2.1.3 Discussion on DeCoSTAR

Adjacencies outside the gene order context

DeCo (and other DeCo-like algorithms) is inherently tuned for gene order problems, which is evident from some of its formulas, most notably the ones about duplication where only one duplicate inherits the adjacency *freely* (*i.e.*, without any *Gain* or *Break* cost), as shown in Figure 2.2 **A**. Other applications may find it better to consider that both duplicates automatically inherit all the adjacencies of their parent (this would be the case of protein interaction networks, where the duplicates, as they are identical at first, would interact with the same proteins), as shown in Figure 2.2 **B**, and extending the recurrence formulas to reflect this would constitute a useful future option for DeCoSTAR.

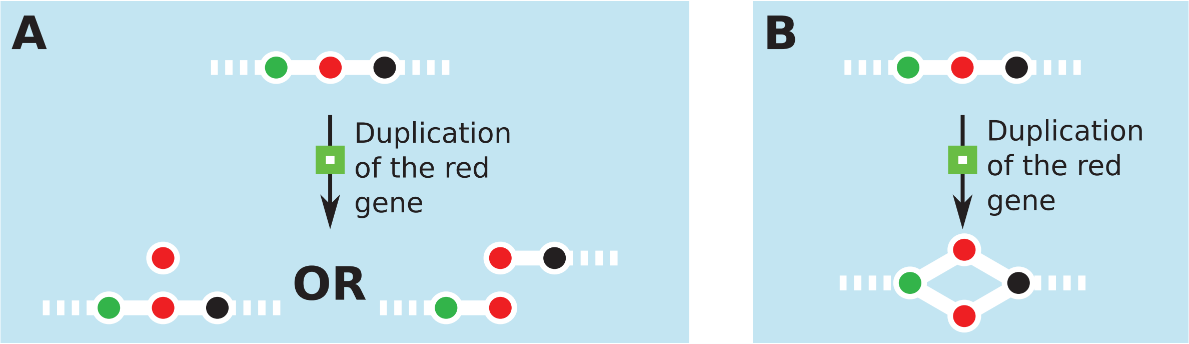


Figure 2.2: Different scenarios of adjacency transmission that do not imply any adjacency gains nor breakages following the red gene's duplication. **A** The adjacencies are transmitted to only one duplicate. **B** The adjacencies are transmitted to each duplicate. The transmission shown in **A** favours linearity, whereas the one in **B** does not.

DeCoSTAR in the context of the integration of co-evolution information

DeCoSTAR can be seen as a method that tries to align two reconciled gene (sub-)trees together and combines this information with extant adjacencies.

An adjacency history contains events such as adjacency gain, adjacency breakage, but also shared events of gene speciation, duplication, loss and transfer (or, to the contrary, events that occurred in one side of the adjacency and not the other).

From a co-evolution perspective, the presence of an adjacency can be seen as a sign of the existence of a certain amount of co-evolution between two genes. Under this hypothesis, adjacency gains thus become the start of the co-evolutionary relationship, adjacency breakages signal the end of it and shared or non-shared events establish a signal for the strength of the relationship between the genes.

From a more practical point of view, I now have at my disposal in a single software a joint topology and reconciliation inference method (TERA) and an adjacency history inference method (DeCo), which constitutes a step forward in the integration of the different forms of co-evolution as sources of information in a coherent framework.

However, it should be noted that this integration fundamentally lacks *intercommunication* between its different components. By this, I mean two things. First, I refer to the fact that TERA and DeCo are connected in a sequential manner: adjacencies (extant or otherwise) play no part in the inference of the topologies and reconciliations of gene families. Secondly, in DeCo (respectively TERA) adjacency histories (resp. reconciliations) are inferred completely independently between each equivalence class (resp. gene family).

This means that for each gene family, a choice of topology and reconciliation is made independently from the other genes with whom it co-evolves (viewed here as the genes it shares adjacencies with) and that any bias or error made in the topology and reconciliation phase will irremediably reflect on the adjacency history inference results without any chance to *go back* and change the genes individual histories given what we know about their adjacencies. It also means that, as each equivalence class has their histories reconstructed independently, it is not possible to enforce constraints that inherently rely on different equivalence classes, such as the constraint of linearity of ancestral chromosomes[3]. Indeed, DeCoSTAR explores the dependency between two genes (through adjacencies) but considers two dependencies as independent, which may lead to conflict (or incongruence) between their respective histories (similar to the conflict between individual gene trees).

These last points are illustrated in the next section where, through an example in the bacterium *Yersinia pestis*, I show how this lack of intercommunication leads to problems when inferring ancestral characters (here, ancestral chromosomes) but also how these problems may be exploited to pinpoint errors made during the inference of topology and reconciliation, and suggest how to correct them.

[3] For instance, if a gene family A has extant adjacencies with genes families B, C and D, we cannot a case where an ancestral gene of A shares an adjacency with ancestral genes of all three other families.

2.2 An application of adjacency history inference to reconstruct ancestral chromosomes

2.2.1 Context of the application

This section presents an application of the DeCo software in a phylogenetic analysis pipeline to reconstruct *ancestral* chromosomal sequences and validate them by comparing them to an *ancient* genome.

Here, we make the distinction between *ancestral* and *ancient* genomes. *Ancestral* genome refers precisely to the genome of the LCA of several extant individuals, and will usually (and this is the case here) be reconstructed from the extant data using phylogenetic methods. *Ancient* genome refers to the genome of an actual organism that lived in the past, which may or may not have left extant descendants, and whose remains have been sequenced. Both offer a window to the past but reconstructed ancestral genomes are, by nature, hard to validate as they rely on the adequateness of evolutionary models with what really happens while ancient genomes are often of bad quality (because the DNA present in the remains of the dead organism deteriorated). This is why we chose this *Yersinia pestis* example, for whom we have ample extant sequence data and an ancient genome that is both of good quality and is hypothesized to be phylogenetically close to an ancestor of extant strains of the bacterium[4].

The reconstruction of a whole ancestral genome at the scale of the nucleotide is, in itself, a task that was never achieved before and that opens new venues of study (in particular for intergenic regions). However, the most interesting part of this work, in the context of this document, is the reconstruction of the ancestral gene order which demonstrates some limitations of the current methods of inference which reconstruct gene trees and reconciliations independently from adjacencies histories, and also treats each gene family or equivalence class (*i.e.*, group of homologous adjacencies) separately, without intercommunication. Also of note is the suggestion on how the symptoms of these limitations could be used to provide corrections at the scale of individual gene families. More precisely, how the results of DeCo, here in the form of an ancestral adjacency graph (a graph where nodes are genes of a given ancestral species, and edges are adjacencies between these genes) upon

[4] Which is not a given, as an ancient organism may not have left any descent and it may have diverged quite a lot from its last common ancestor with an extant organism.

whom we want to apply constraints of linearity (as they represent chromosomes), could influence (and subsequently benefit from) the choice of tree topology and reconciliation.

It is worth noting that this work was done before the implementation of DeCoSTAR (evidenced by the absence of HGT in the analysis for instance). However I felt that, as it uses DeCo and its results, it would make sense to write about the present analysis only after the DeCo-like approaches we properly introduced.

2.2.2 Reconstruction of an ancestral *Yersinia pestis* chromosomes

BMC Genomics

RESEARCH Open Access

Reconstruction of an ancestral *Yersinia pestis* genome and comparison with an ancient sequence

Wandrille Duchemin[1], Vincent Daubin[1], Eric Tannier[1,2*]

From 13th Annual Research in Computational Molecular Biology (RECOMB) Satellite Workshop on Comparative Genomics
Frankfurt, Germany. 4-7 October 2015

Abstract

Background: We propose the computational reconstruction of a whole bacterial ancestral genome at the nucleotide scale, and its validation by a sequence of ancient DNA. This rare possibility is offered by an ancient sequence of the late middle ages plague agent. It has been hypothesized to be ancestral to extant *Yersinia pestis* strains based on the pattern of nucleotide substitutions. But the dynamics of indels, duplications, insertion sequences and rearrangements has impacted all genomes much more than the substitution process, which makes the ancestral reconstruction task challenging.

Results: We use a set of gene families from 13 *Yersinia* species, construct reconciled phylogenies for all of them, and determine gene orders in ancestral species. Gene trees integrate information from the sequence, the species tree and gene order. We reconstruct ancestral sequences for ancestral genic and intergenic regions, providing nearly a complete genome sequence for the ancestor, containing a chromosome and three plasmids.

Conclusion: The comparison of the ancestral and ancient sequences provides a unique opportunity to assess the quality of ancestral genome reconstruction methods. But the quality of the sequencing and assembly of the ancient sequence can also be questioned by this comparison.

Background

Extant species are derived from a process of evolution and diversification from species now disappeared. These species are called ancient in general and ancestral if they left a descendant. Ancestral genomic sequences can be estimated through computation from a set of extant sequences related by a phylogeny and a model of evolution [1], while ancient genomic sequences in general can be sequenced from the remains of dead organisms [2].

Ancestral genome reconstruction

Ancestral genome reconstruction can consist in predicting a gene content in ancestral species [3], and for each gene its sequence [1]. While originally used to study proteins or isolated genes, ancestral genome reconstructions are now robust at a scale larger than the gene, for fragments where no rearrangement have occurred [4]. Methods for inferring ancestral gene orders have also been explored [5-8]. Together, these methods open the way to the reconstruction of complete ancestral genomes, including their sequences.

Obtaining ancestral sequences can allow, through the study of physical properties of the reconstructed molecules, the inference of the paleoenvironnements in which these molecules evolved [9]. These methods also allow access to an oriented and ordered view of molecular events along the history of life. Moreover, they offer a better understanding of this history and can further our knowledge of the mechanisms linking organic sequences to their functions [10].

Despite this, ancestral sequence reconstruction suffers from several limits. Along with the study of molecular evolution, it relies on the validity of models and their

* Correspondence: eric.tannier@inria.fr
[1]Laboratoire de Biométrie et Biologie Évolutive, LBBE, UMR CNRS 5558, University of Lyon 1, 43 boulevard du 11 novembre 1918, 69622 Villeurbanne, France
Full list of author information is available at the end of the article

fundamental hypothesis. Furthermore, given that we are interested in a phenomenon often distant in time, it is at best difficult to obtain proofs validating proposed predictions. Thus, the validation of ancestral reconstruction methods is often limited to robustness tests, or simulations that themselves rely on the validity of the models of evolution [1].

Ancient genome sequencing

Ancient DNA sequences is another way to have an access to the past history of living organisms. Under certain conditions it is possible to obtain genetic material through the sequencing of the remains of an organism. Ancient DNA sequencing began in the middle of the 80s with the cloning and sequencing of fragments of mitochondrial DNA in a museum specimen of *Equus quagga*, an extinct equine species that disappeared in the XIXth century [11]. The advent of PCR methods [12] and high-throughput sequencing [13] followed by what is called third generation sequencing [14] allowed the sequencing of several extinct animals [15-17], ancient unicellular eukaryotes [18,19], bacteria [2,20,21], metagenome [22], or virome [23].

The ancient sequences disclose a new source of information concerning the evolution of lineages of interest. They have already been used, among other things, to understand the dynamic of extant populations of the genus *Homo* [24-26], or other animals [27], to correct and recalibrate phylogenies [17], or to better understand past pandemics [18,19,2,20,21].

However, along with the problems specific to sequencing technologies, ancient DNA sequencing is limited by the post-mortem chemical degradation of DNA molecules throughout time. Thus, like fossils, ancient sequences are scarce while, unlike them, limited to recent times.

Yersinia pestis

Classified among *Enterobacteriaceae, Yersinia pestis* is the bacterium thought to be responsible for the bubonic plague and the pneumonic plague. It diverged from the *Yersinia pseudotuberculosis* lineage, in part through the acquisition of two plasmids [28]. It has been demonstrated that strains of *Yersinia pestis* caused the black death of 1347-1353 AD that is thought to have killed between a third and half of the European population at that time and persisted in Europe until the middle of the XVIIIth century [29]. An ancient genome has been extracted and sequenced [2]. It was the first whole ancient bacterial genome. Based on a substitution pattern compared to extant *Yersinia* species, it has been hypothesized to take place on the extant species phylogeny in the vicinity of a known speciation node leading to two set of extant, sequenced and annotated strains of the bacterium (see Figure 1).

The existence of several sequenced and annotated extant genomes as well as the relatively short evolutionary time separating them make their ancestor a good candidate for an ancestral reconstruction including both sequence and gene organization along the chromosome and the plasmids. However despite the short evolutionary time, while substitutions are quite rare [2], there is a very active dynamics of rearrangements, insertion sequences propagation, duplications, copy number variation (see Figure 2), which makes the problem challenging.

The late-medieval ancient genome, likely close to that ancestor, offers a validation opportunity for the ancestral reconstruction method. We achieve here this reconstruction and perform the comparison.

Note that a sequence of the same genome was proposed recently by Rajaraman et al. [30], but was not issued from ancestral reconstruction. The contigs of the ancient genome were scaffolded with a method including the phylogeny of relatives, and some parts of the assembly could be corrected, but what we present here is not using at all the ancient sequence in the reconstruction phase, it is done only from independent extant data.

Methods

An overview of the method, including species tree construction, gene tree construction and reconciliation, gene order inference and gene tree corrections according to this gene order, and eventually genic and intergenic sequence prediction, is illustrated on Figure 3.

Data set

The data consists in 13 *Yersinia* annotated genomes (Figure 1) from which we extract 3772 homologous protein gene families containing at least two genes, using the HOGENOM database [31]. Of these, 1971 have exactly one copy per extant strain. This step corresponds to part A in Figure 3.

Species tree

Using Muscle [32] (default parameters), we aligned the 1971 families, concatenated the variable sites of all alignments and obtained a phylogenetic tree using PhyML [33] (100 bootstraps, otherwise default parameters) that we rooted by separating the *pestis* from the *pseudotuberculosis* clades, according to a consensus in the literature. In our tree the branch separating the two clades is well supported, as well as the branches surrounding the ancestor that we wish to reconstruct (see Figure 1). This step corresponds to part B in Figure 3.

Gene trees

All gene families sequences were then aligned using Prank [34] and one gene tree per family was computed using PhyML (100 bootstraps, otherwise default parameters). Because we are aligning recently diverged strains of the same organisms [35], the sequences often have not

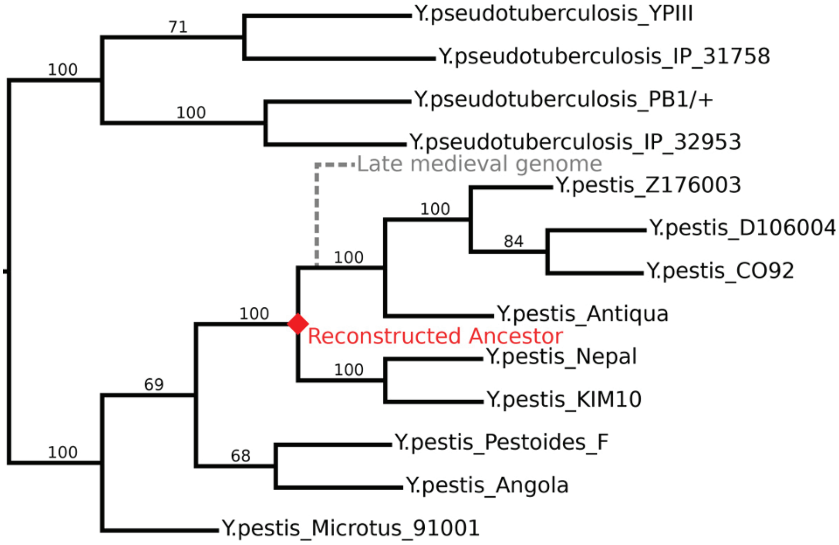


Figure 1 Yersinia pestis and pseudotuberculosis phylogeny. Tree obtained using a 1971 universal gene families concatenate. Bootstrap values are figured on the branches. For readability, the figured branch length is the inverse of the ten-logarithm of the real branch-length. The ancestral species of interest to us is figured as a red diamond. The late medieval ancient genome hypothetical position is figured in gray and dashed.

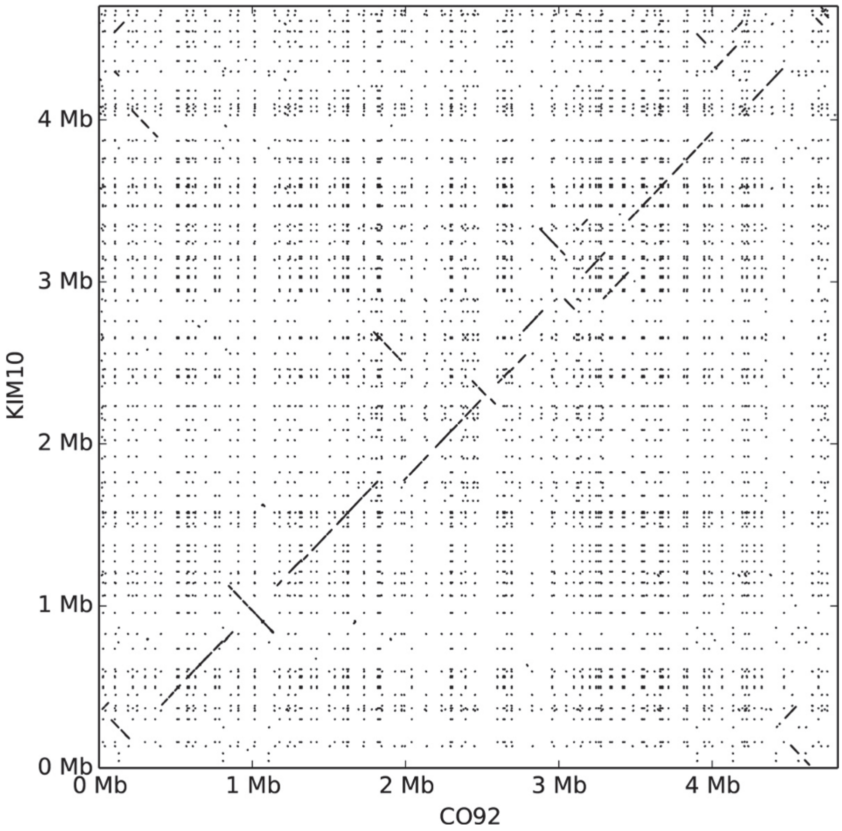


Figure 2 Dotplot between the sequence of two extant strains of Yersinia pestis: CO92 and KM10. Both strains are descendants of the ancestor we focus on. Data was obtained by aligning the sequence of strain KIM10 on the sequence of strain CO92 using megablast (default parameters, only hits with a length >$10^{2.5}$ were kept).

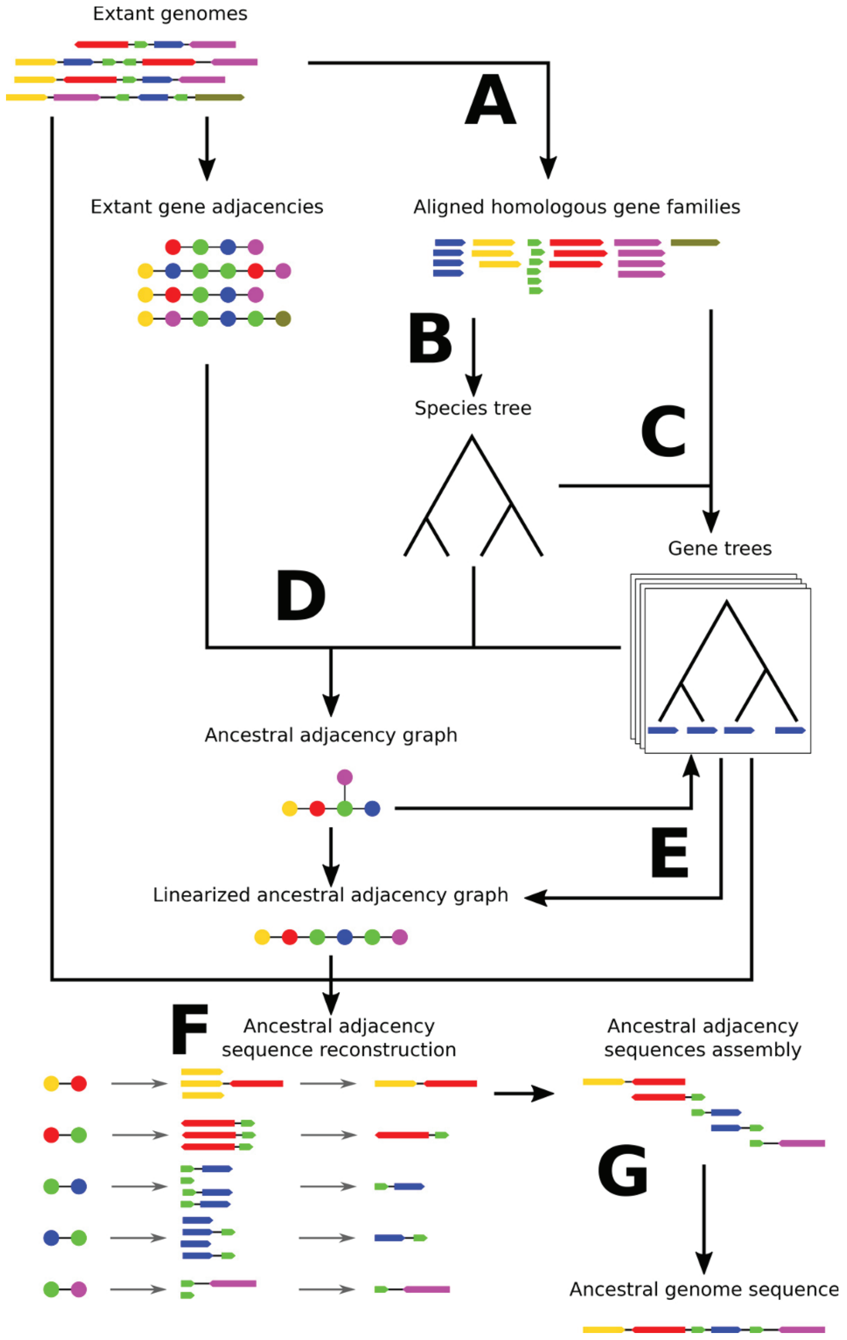

Extant genomes
A
Extant gene adjacencies
Aligned homologous gene families
B
Species tree
C
Gene trees
D
Ancestral adjacency graph
E
Linearized ancestral adjacency graph
F
Ancestral adjacency sequence reconstruction
Ancestral adjacency sequences assembly
G
Ancestral genome sequence

Figure 3 Protocol used to obtain the ancestral gene order and sequence of a Yersinia pestis ancestor. A) Extraction and filtering of gene families from extant genomes and alignment. B) Reconstruction of the species tree using a concatenate of the variant positions of 1971 universal gene families. C) ML reconstruction of gene trees followed by the collapse of any non-supported branch (bootstrap <99) and the resolution of the created polytomies using the species tree as a guide. D) Inference of ancestral gene adjacencies using DeCo. E) Detection and correction of wrongly inferred gene trees based on the ancestral adjacency graph linearity. F) Reconstruction of the ancestral sequence of each gene adjacency from their extant descendants. G) Alignment of the consecutive ancestral adjacency sequences to assemble the ancestral genome. Similar colors indicates homology. Dots represent a gene as a node in an adjacency graph while oriented segments represent a gene as a sequence.

diverged enough to allow an unambiguous tree reconstruction. So we collapsed all branches with a support lower than 99 and then used ProfileNJ [36] to solve the created polytomies. ProfileNJ reconstructs species tree branches instead of collapsed branches and chooses among several solutions with a Neighbor-Joining formula. Distances for the Neighbor-Joining part were computed with bppdist, a Bio++ suite software [37] (GTR + Γ(4) model).

ProfileNJ also roots the gene trees according to "Last Common Ancestor" reconciliation method, annotating internal nodes with duplications or speciations, and choosing a root minimizing the number of duplications.

Reconciled gene trees depict the history of the gene family, including all ancestral genes, uniquely defined by the reconciliation.

This step corresponds to part C in Figure 3.

Gene families filtering

From the 3772 gene families, some were discarded because they showed signal of a process that we do not handle well in our pipeline, gene transfer. Transfer was suspected when a branch in the reconciled gene tree would correspond to at least 4 independent losses in the species tree. We also removed the families with more than 5 genes in the black death ancestor, suspecting insertion sequences, which are poorly handled by the method. We also removed families containing genes fully included in other genes: as we model the evolution of gene orders, these would be difficult to handle. We eventually removed families when the reconciled gene tree did not contain a gene in the ancestor we want to reconstruct.

The final data set contained 3656 families. Note that when removing gene families from the study, we do not necessarily give up the reconstruction of parts of the ancestral sequence. We just define the removed parts as intergenic. As we also reconstruct intergenic sequences, this simply modifies the resolution at which we are able to detect rearrangements.

Extant gene order and adjacencies

Each gene is a segment of a chromosome or a plasmid and has a start and an end position on it. We identify these positions as the *extremities* of the gene. A start position may be greater than an end position: the order of the extremities defines the *orientation* of the gene. We model each genome by a graph, whose nodes are gene extremities of genes in that genome. We put an edge, called an *adjacency* between pairs of extremities of a same gene. Additionally if genes AA' and BB' are consecutive (A and A' are the extremities of the first gene, appearing in that order on the chromosome or plasmid, and B, B' are the extremities of the second gene), we put an adjacency between A' and B. So extant genomes are sets of disjoints cycles in a graph, modeling chromosomes and plasmids.

Gene extremities can be clustered into families, inherited from gene families, and also inherit the reconciled gene tree.

Ancestral gene order

Ancestral adjacencies between gene extremities were inferred using DeCo [7]. It models the evolution of an adjacency between two gene extremities following a parsimony principle, *i.e.* minimizing the number of gains and breakages of adjacencies, due to rearrangements. It takes as input the species tree, all gene trees, and extant adjacencies, and proposes a set of ancestral adjacencies between ancestral gene extremities defined by the reconciled gene trees. This step corresponds to part D in Figure 3.

DeCo assumes that adjacencies evolve independently. This means in particular that ancestral gene extremities can be involved in an arbitrary number of adjacencies. Ancestral gene extremities and adjacencies are not necessarily made of cycles as extant genomes, so we call this object an *adjacency graph*. Figure 4 shows the obtained adjacency graph at this step. While most of it shows a linear or circular structure, there are some gene extremities with too many adjacencies, others with not enough.

There can be several reasons for the adjacency graph not to be a collection of paths and cycles, as we would expect if the data and methods were perfect. Incorrect gene trees

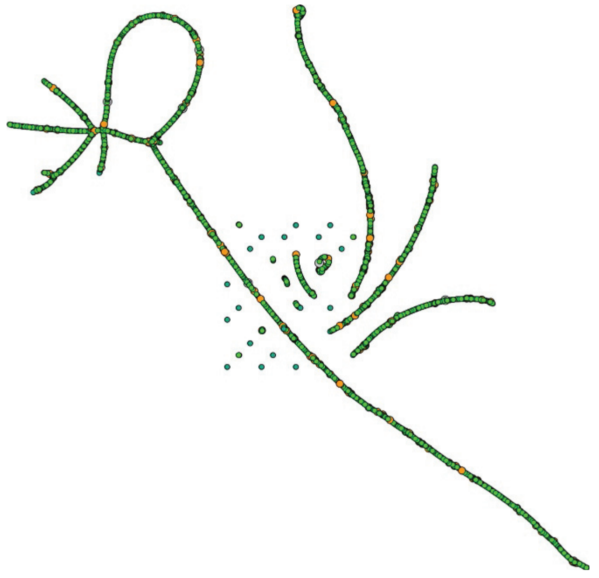


Figure 4 Ancestral adjacency graph obtained using DeCo on the set of 3656 gene families. Each node is colored according to its number of neighbors: green for two (ideal, linear case), turquoise for one (where one adjacency has been lost), orange for three and gray for four (when an error in the number of ancestral copies creates conflict in the ancestral gene order).

are probably the major source of such discrepancies, while others may come from uncertainties in adjacency history inference.

We transform the adjacency graph into a genome (*i.e.* an adjacency graph that is a collection of paths and cycles), first by correcting gene trees, by operations we call zipping and unzipping, then by removing a minimum number of adjacencies so that the remaining graph is a genome.

Correcting gene trees

This step corresponds to part E in Figure 3 and a more detailed picture is on Figure 5.

Unzipping

Each ancestral gene extremity of a gene g should have at most two adjacencies. If one has more than two, a first hypothesis can be that in the real ancestral genome, the gene g was duplicated in two copies, and each copy would carry some of the adjacencies of g.

If in one extant species, there are two homologous copies of the gene g, and their extremities share the homologs of the adjacencies attributed to an extremity of g, then we perform the *unzipping* operation.

It consists in making two genes out of g by modifying the gene tree T of the gene family containing g. Only the subtree rooted at g is changed, into a subtree rooted at a new duplication node with two descendants: g and a new gene g'. Then the two subtrees rooted at g and g' are reconstructed, first by assigning all leaves to g or g' according to their neighborhood; Then by constructing subtrees on these leaves using ProfileNJ. In the case where some leaves can't be assigned to either g or g' using their neighborhood (*i.e.* their extant neighbors are not descendant of any of the ancestral neighbors), then leaves are assigned to one of the two set of leaves according to their mean phylogenetic distances with them. Where there is a tie (for instance if all sequences are identical, all distances are null), the leaf is randomly assigned to one of the two leaf-set.

Figure 5A gives an example of an unzipping operation on the ancestral adjacency graph and on the gene tree.

If the unzipping procedure increases the number of adjacencies incident to a gene extremity of a gene h in the immediate neighborhood of g in the adjacency graph, then the unzipping procedure is applied to h as well, and then to its neighbors, until the region is linearized.

Zipping

Another possible reason for a gene g to be involved in more than two adjacencies is that two of these adjacencies gh and gh' concern two paralogs h and h' which in reality should form only one gene. In that case we perform a *zipping* operation, similar to the one described in [38].

Let h_d be the last common ancestor of h and h' in their gene tree. Suppose it is assigned to species s, whose descendants are s_1 and s_2. It is a duplication node, and we turn it into a speciation node by giving it two descendant nodes h_1 and h_2, and assigning its descendant leaves to either one of them, depending on whether they are genes from descendants of $s1$ or s_2. Then subtrees rooted at h_1 and h_2 are reconstructed using ProfileNJ.

Figure 5B gives an example of a zipping operation on the ancestral adjacency graph and on the gene tree.

Zipping produces a new ancestral gene h_d instead of two paralogues h and h'. We propagate the same operation to the neighbors of the ancestral gene h_d in the adjacency graph if they are themselves supernumerary paralogues.

Note that for zipping and unzipping, the propagation mechanism allows the treatment of several consecutive nodes, such that a large segmental duplication containing multiple genes can be dealt with as long as there exists a node to start the unzipping move (*e.g.* at one extremity of the segmental duplication).

Cutting

Zipping and unzipping are tested independently for each ancestral node with more than two neighbors. Each of them should decrease the number of gene extremities with more than two adjacencies. The operation that decreases it the most is kept.

If none of zipping and unzipping succeeds in removing all such supernumerary adjacencies (it is possible that none of the hypotheses applies), then we remove as few adjacencies as possible so that only gene extremities with at most two adjacencies remain. This is achieved using a maximum matching technique described in [39].

Ancestral sequence reconstruction

Ancestral sequences have to be reconstructed by pieces, because they need a multiple alignment free of rearrangements. The pieces have to be glued together, and in order to avoid between pieces border problems, pieces have to overlap. This is why we reconstruct an ancestral sequence for all pairs of genes which are connected by an adjacency. Then pairs are aligned together on their common gene, and merged.

We orient each adjacent gene pair with a first and a second gene, each gene should be once the first gene of a pair, and once the second in another pair. We use the gene tree of the first gene as a guide, to construct a multiple sequence alignment with the extant sequences that contain this adjacent pair (thus, the sequences contains both genes and the sequence between them when they are neighbors in an extant species, and only the first gene of the adjacency when they aren't), and the ancestral sequence using Prank [34].

Gene sequences at the ends of contigs are reconstructed alone using their own tree. In consequence each intergene sequence is reconstructed once and each gene sequence is reconstructed twice and at least once with its own tree. We assemble the obtained ancestral sequences

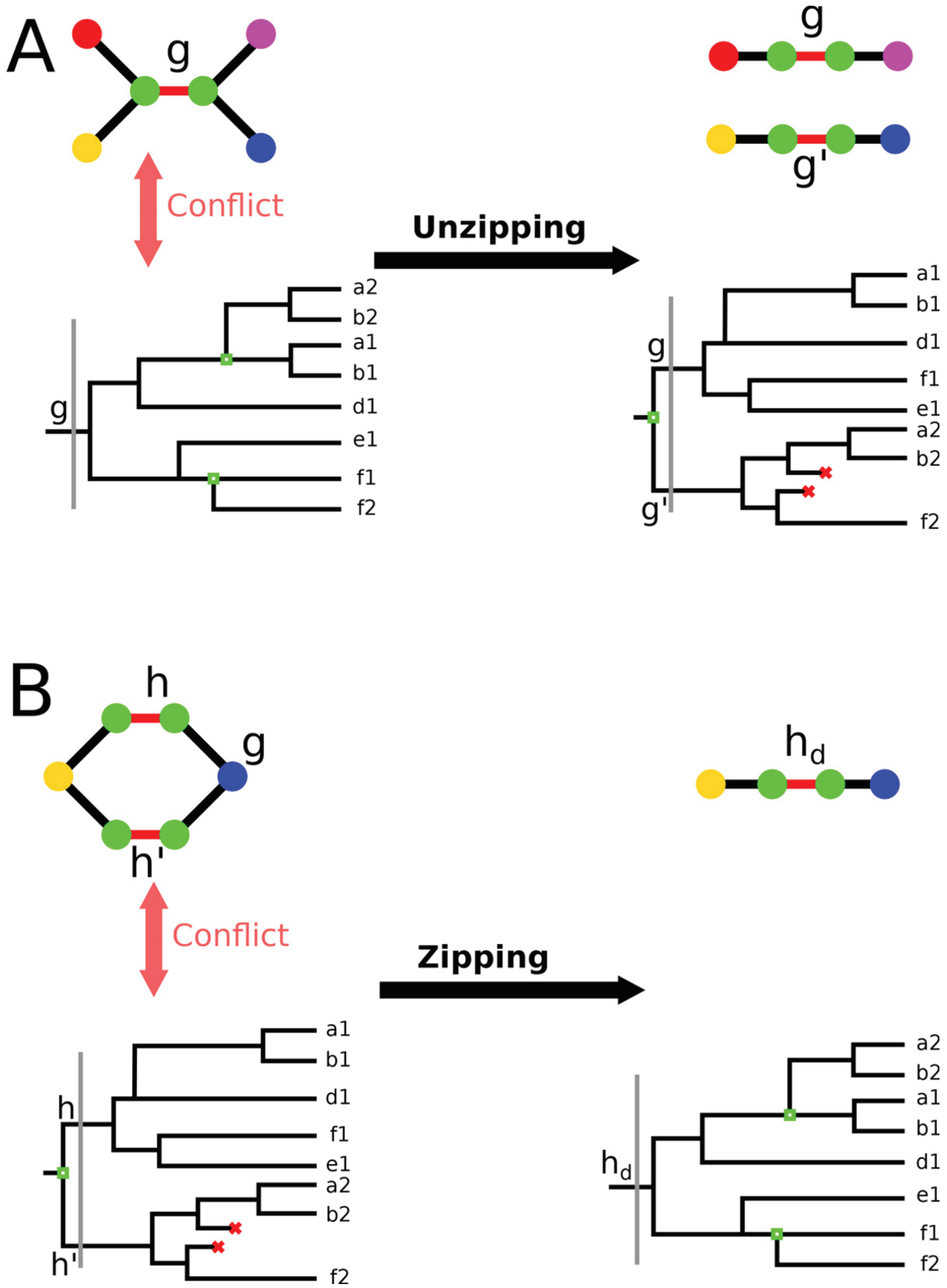


Figure 5 Illustration of the unzipping and zipping on gene trees and adjacency graphs. A) Prior to linearization (left of the black arrow), the gene *g* exists in one copy in the ancestor (vertical gray line on the tree) and two independent duplications occurs in its descendants (green hollow squares). In the ancestral adjacency graph above each of *g* extremities displays two neighbors. Unzipping (right of the black arrow) modifies the tree so that there are two ancestral copies *g* and *g′* each corresponding to a different path in the ancestral adjacency graph (losses in the tree are displayed as red crosses). B) Prior to linearization (left of the black arrow), two ancestral copies of the same gene *h* and *h′* exist in the ancestor (vertical gray line on the tree; losses in the tree are displayed as red crosses). In the ancestral adjacency graph above the extremities of *h* and *h′* each share a neighbor, forming a non-linear pattern. Zipping (right of the black arrow) modifies the tree so that there only one ancestral copy h_d followed by independent duplications in its descendants (green hollowed squares), forming one linear path in the ancestral adjacency graph.

by aligning (using Smith & Waterman's algorithm) the ones sharing a gene and then making the consensus sequence of that alignment, favoring the sequence reconstructed with the tree of the aligned gene.

For instance, consider the ancestral path *ABC* (where *A*,*B* and *C* are genes), we reconstruct the ancestral sequence of *A* using its own tree, *AB* using *A*'s tree, *BC* using *B*'s tree and *C* using its own tree. Afterward the ancestral sequence of *A* is aligned with the ancestral sequence *AB*, favoring the sequence of *A* when computing the consensus. Then the sequence *AB* is aligned with the sequence *BC*, favoring the sequence *BC* in the consensus (as both sequences align on gene *B* and *BC* used *B*'s tree for the reconstruction). Finally, the sequence *ABC* is aligned with the sequence *C*, favoring *C* in the consensus.

A graphical view of these steps are given in Figure 3, parts F and G.

Note that, as stated before, the ancestral sequence reconstruction needs a multiple alignment free of rearrangements. This means that the size of the recombination events that can be taken into account for ancestral sequences reconstruction depends on the density of the markers (here, the gene extremities of 3656 gene families) used in the ancestral order reconstruction step.

Results

The shape of the ancestral genome

We perform the whole process of ancestral gene order reconstruction for three data sets: the whole set of filtered families, the set of D free families, without duplication and the DL free families, without duplication nor loss.

Ancestral gene order is computed with the whole set, but it gives fragmented paths in the adjacency graph. The fragments are progressively assembled using the D free and DL free gene orders.

The ancestral gene order was reconstructed for the chromosome (3342 genes) and the three plasmids (pCD: 74 genes, pMT: 87 genes, pPCP: 5 genes). The plasmids pCD and pPCP were obtained as circular elements in the adjacency graph, while the plasmid pMT was represented by one linear fragment. The chromosome was obtained as three linear components. To join these components, we ran DeCo on their six extremities using a gradient of adjacency gain/loss costs ratio (from 1/10 to 10/1) and scored each potential adjacency by the number of times it was observed. We then applied a weighted maximum matching technique [40] to extract the best possible order between the fragments (only one optimal solution remained).

The ancestral gene order is different from all extant genomes. For example it is an intermediary between the two extant strains *CO92* and *KIM10*. Figure 6 B and C show the gene order comparison between the ancestral genome and two extant ones, while a comparison between the two extant ones is shown on Figure 6A. The isolated dots on the dotplots of Figure 6B and C are probably reconstruction errors. While they could be explained as small rearrangements, they probably are artifacts of the adjacency graph linearization method, like a leaf falsely associated to a subtree in an unzipping event for instance.

The ancestral sequences of the plasmids pCD, pMT and pPCP were entirely reconstructed, for a total of respectively 100.1 kb, 67.7 kb and 9.6 kb. Concerning the ancestral chromosome, a total of 4.7 Mb of ancestral sequence was reconstructed, which is close to the size of the extant chromosomes of *Yersinia pestis* strains (*e.g.* 4.7 Mb for the strain *Antiqua*). A lack of signal in extant genomes due to convergent rearrangements, prevented the reconstruction of four ancestral adjacencies. Because of these, the ancestral chromosome sequence is actually composed of four disjoint fragments (their sizes are respectively 3.44 Mb, 0.67 Mb, 0.40 Mb and 0.19 Mb).

The reconstructed ancestral sequences are avalaible in Additional file 1.

Comparison to the ancient genome

Using Megablast [41] we aligned the 2134 ancient *Yersinia pestis* contigs obtained by Bos et al.[2] (avalaible at http://paleogenomics.irmacs.sfu.ca/FPSAC/, last accessed 19 june 2015) against the obtained ancestral genome, including chromosome and plasmids.

We examine 2179 hits of length >$10^{2.5}$bp from 2087 contigs (see Additional file 2 for the bimodal distribution of hit lengths which justifies this threshold). The others are full of repeated elements, making the comparison difficult. As a consequence the examined hits all match to the chromosome and none to the plasmids.

Gene order

These hits show a quasi-total congruence between the organization of the ancient and ancestral sequence. Figure 7 represents the correspondence between the two in the form of a dotplot, where contigs of the ancient genome are concatenated according to the ancestral sequence. Three isolated dots deviate from the central line. Two of them concern large repeated regions, that is, the whole contigs match at several places. Only one seems to be a real discordance between the two genomes. Two contiguous regions of the contig hit on two different ancestral sequence fragments. This chimeric contig (number 8335 in [2]) had already been observed by Rajaraman et al. [30] in their scaffolding of the ancient genome. This stretches the proximity and the differences between the two approaches. Indeed, the latter, called FPSAC, takes as input the ancient contigs and the extant genomes, fragments the contigs according to their alignments to extant genomes, and orders fragments. Here we don't use at all

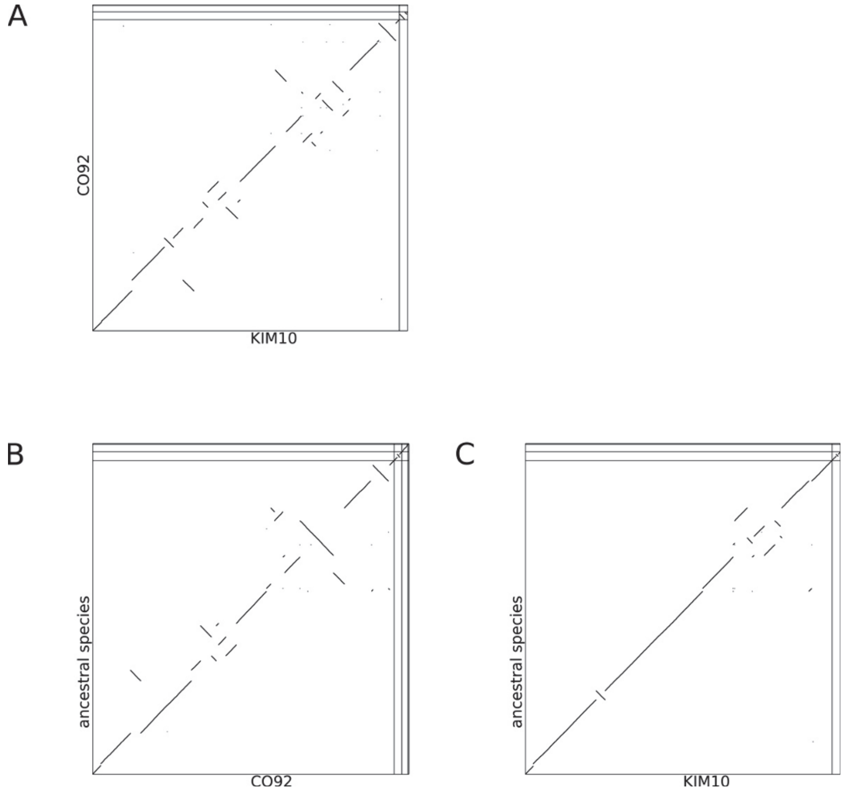


Figure 6 Dotplot between the ancestral genome and two extant strains of Yersinia pestis: CO92 and KIM10. Both strains are descendants of the ancestor we focus on. Data was obtained using the extant adjacency graphs of strains KIM10 and CO92 and concerns genes order. Vertical and horizontal lines separate the different molecules (here the chromosome and the plasmids). A) dotplot between the gene orders of the two extant strains KIM10 and CO92. B) dotplot between the gene orders of the ancestral genome and the extant strain CO92. C) dotplot between the gene orders of the ancestral genome and the extant strain KIM10.

the ancient contigs and start from extant genes. So we are independent of the extraction and assembly methodology for the ancient sequence, and we can compare to it. Moreover, all our sequences are computationally reconstructed, which was not the case of those obtained with FPSAC.

So at a large scale, there is only one difference which can be an assembly error in the ancient sequence or a derived mutation of the ancient bacteria, because the ancient configuration is not supported by extant genomes.

Sequences

At a finer scale, differences are more numerous. Approximately 81% of the 2084 contigs with a hit are exact matches to the ancestral genome. We examined some of the remaining and found that the differences could be explained by three kinds of error sources in the ancestral or ancient sequences:

- Lack of sufficient data for ancestral reconstruction: it is the case if only one of the two children which branches off the ancestor, in addition to an outgroup, support the presence of a sequence. In that case there is no comparison point to infer some bases, and some are inferred differently than in the ancient sequence.
- Lack of a good model of evolution at an intermediary scale, like duplication of small elements. They are here included in alignments and indel models, which do not account for repetitions.
- Assembly errors in the ancient sequence.

Consider for example the ancient contig number 497 where a mismatch occurs when aligned with the ancestral sequence. The mismatch is situated in an intergenic region of the ancestral genome that is present in one descendant of the reconstructed ancestor and two outgroup *Yersinia pestis* species. Consequently, the ancestral sequence was reconstructed using a tree where the node of interest was along a branch, missing a comparison point (*i.e.* another descendant) to choose between its descendant allele and the outgroup allele.

Consider also the ancient contig number 8849 which aligns with one mismatch to the reconstructed ancestor. At the position of the mismatch, all extant (group and

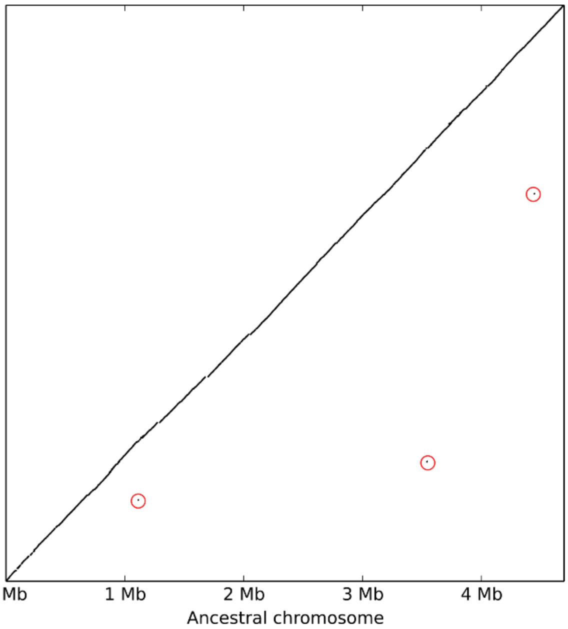


Figure 7 Dotplot between the late medieval Yersinia pestis genomes and the reconstructed ancestral sequence. The reconstructed chromosome was aligned to the 2134 ancient contigs, using megablast (default parameters, only hits with a length >$10^{2.5}$ were kept). Contigs are concatenated according to the reconstructed sequence, so the agreement is partly due to the fragmented nature of the ancient sequence. The contigs with hits departing from the diagonal are circled in red.

outgroup species) sequences bear the same allele and thus the reconstructed ancestral sequence bears it too. However, the ancient contig bears another allele at that position. If we consider the ancient contig as correct, then this difference would be an original mutation on the ancient strain. Such an hypothesis could be checked by mapping the ancient reads to their contigs in order to assess the validity of that specific allele. However, we note that the original study [2] that used read data to call SNPs did not detect any that were specific to the ancient strain.

There are also differences that are more structural in kind. For example 43 contigs show some structural differences with the ancestral genome. On 39 of them, the ancient contig displays two contiguous or slightly overlapping hits that are more distant on the ancestral genome (on 21 occasions, they are more than 300 bp apart in the ancestral sequence), as in Figure 8A. On 4 ancient contigs, contiguous regions are shown as overlapping in the ancestral genomes, as in Figure 8B.

Such discrepancies can sometimes be explained by errors in the ancient sequence, especially in regions where repetitions occur. For instance, the case illustrated on Figure 8A, is seen on the contig number 8335 obtained by Bos et al.[2] (which is also the chimeric contig but this discrepancy is independent). Around position 1860, that ancient contig displays one occurrence of a 20-mer. However, the reconstructed ancestral sequence has two consecutive occurrences of that 20-mer. This region is situated in an intergenic region, so it has been reconstructed by an alignment of an adjacency with its two flanking genes. The extant species (descendant of the reconstructed ancestor or not) which have this gene adjacency all display two occurrences (in favor of the ancestral reconstruction) at the exception of *Yersinia pestis* strain CO92, the *Yersinia pestis* reference genome which was used to map the ancient reads in [2]. While the fact that we did not use the raw reads obtained in [2] prevents us to draw any definitive conclusion, this appears to be an error in the ancient sequence assembly, caused by a derived mutation in the genome used as a reference.

Conversely, it happens that similar patterns are better explained by errors in the reconstructed ancestral sequence. Such a case occurs on the locus where the ancient contig number 5613 maps. The situation is also similar to Figure 8A. Two contiguous regions hit at a distance of 1315 bp on the reconstructed ancestral sequence. The sequence separating the two hits in the ancestor is only supported by one extant descendant (*Nepal* strain) and the other extant descendants match the ancient contig in only one long hit. This seems to be an error due to the absence of an evolutionary model allowing big insertions. Prank models indels but 1315 bp is not really an indel but is rather an insertion of what should perhaps have been an evolutionary unit. It seems that the indel model prefers losing several times such a long DNA segment rather than inserting it once in a terminal branch of the phylogeny. So we can expect a small number of such false additions in the ancestral sequence.

Discussion

A complete reconstruction of an ancestral genome at the nucleotide level requires to take into account evolutionary events at several scales: nucleotide substitutions, indels, duplications, losses, recombinations, transfers, transposable elements propagation, rearrangements. Each level is handled by dedicated bioinformatics tools which are rarely used together.

We associated here gene content/sequence/order tools in order to attempt the reconstruction of a whole ancestral bacterial genome, including a chromosome and three plasmids. We chose an organism from the *Yersinia pestis* clade because of a recently published ancient sequence. Despite being relatively recent at the evolutionary scale (650 years), the evolution at all levels, and in particular in genome structure and organization, makes the problem difficult. The difficulty can come from numerous events (rearrangements, insertion sequence dynamics), but also

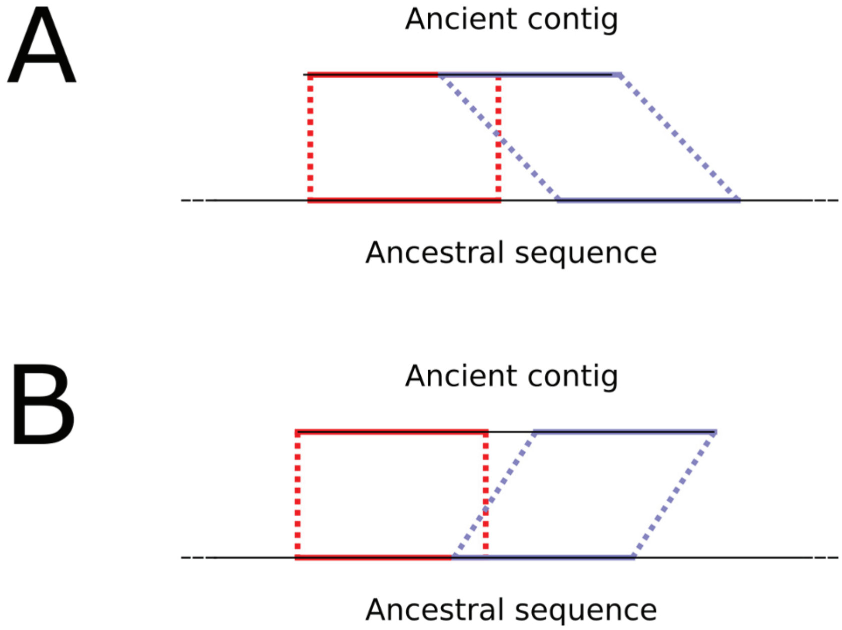


Figure 8 Different hit patterns for ancient contigs on the ancestral sequence. A) Contiguous or overlapping hits are more distant on the ancestor. B) Contiguous or distant hits are closer together (or overlapping) on the ancestor.

from scarce events (substitutions) that prevent reconstructing gene trees from sequences because of a lack of information.

We did not only assemble existing tools that handle evolution at different levels, but also report methodological novelties, like the zipping and unzipping processes to modify gene trees and linearize adjacency graphs. Using synteny information to construct gene trees is rarely achieved [36] and linearizing often only use cutting operations [39].

We cannot explicitly handle recombination events or gene transfers, duplications at levels different from the gene, and propagation of insertion sequences. Some tools exist to reconstruct gene content or order in the presence of transfers [3,42], but not equivalent to ProfileNJ [36], which we used because of a lack of signal from the sequences in many gene families. It has not been developed for transfers apparently for algorithmic purposes [43]. Transfers will probably limit the quality of the sequence, which at recombination points will be reconstructed with a wrong gene tree. We expect these limits to be rare, as we found only little evidence of gene evolution clearly discordant with the species tree.

Another limit of this method is that it handles evolution at three different scales: sequence, gene content, gene order, while evolution happens at a continuum of scales, some part of it we don't explicitly model. This is for example the case for small duplications: gene duplications are handled but if they are smaller than genes, duplications will be part of sequence evolution, where the models and alignements take indels into account but not duplications. This is also the case of insertion sequence propagation. If insertion sequences are annotated as genes, their dynamics is sometimes so fast that parsimony duplication/loss principles are not accounting for it, even within a very small amount of time. If insertion sequences are taken in intergenic regions, they will again be handled inside alignments and yield a small amount of false positives.

A small part of the sequence is not reconstructed because of convergent rearrangements which have wiped the traces of some intergenic sequences. These convergent rearrangements also introduce one ambiguity in the ancestral gene order. It is possible that it reflects an ancestral polymorphism which has differently been resolved in different lineages.

Polymorphism, and the absence of it in our ancestral genome, is another limitation of such an approach. The ancient population was probably composed of several variants, and the 650 years might not be sufficient to sort out all of it. So we are not sure that a single organism carried the genome we reconstruct, but it might be a consensus of several genomes.

Yet these limits concern probably a very small percentage of the sequence, which is largely reconstructed with a total match to the ancient sequence. Beyond the methodological challenge and the interesting comparison with an ancient genome, the goal of such a reconstruction is not to find an application in synthetic biology, but to understand the evolution of this dangerous pathogen. Substitutions, which apparently are only a minor part of the story, are often the only marker of evolution (for example in [2]) because of a better availability of performing tools.

Conclusions

In conclusion, we report here the reconstructed ancestral bacterial genome of an ancestral *Yersinia pestis*. The reconstruction is achieved using already published software and methods but also introduces methodological novelties, especially concerning ancestral adjacency graph linearization, leading to the obtention of larger reconstructed ancestral chromosome fragments.

The comparison of the reconstructed ancestral genome with an ancient sequence provides the opportunity to assess the quality of the reconstruction. It appears that while the reconstruction methods display some limits for events spanning more than a few nucleotides and smaller than a gene (for instance, a gene domain duplication), they yield good results concerning small (substitutions, short indels) and gene-scale events(for instance, gene duplications or rearrangements spanning at least a gene).

Additional material

Additional File 1: DucheminDaubinTannier2015 supplementary file 2.fas. Fasta file containing the ancestral sequences obtained for the chromosome and plasmids of the ancestral *Yersinia pestis* species.

Additional File 2: DucheminDaubinTannier2015 supplementary file 1. pdf. Histogram of the hit lengths (represented as its log10 here) when ancient contigs are aligned to the ancestral genome of *Yersinia pestis*.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

WD, VD and ET conceived the method, WD implemented and tested it. WD and ET wrote the article.

Acknowledgements

This work is funded by the Agence Nationale pour la Recherche, Ancestrome project ANR-10-BINF-01-01.
Publication charges for this work were funded by the Agence Nationale pour la Recherche, Ancestrome project ANR-10-BINF-01-01.
This article has been published as part of *BMC Genomics* Volume 16 Supplement 10, 2015: Proceedings of the 13th Annual Research in Computational Molecular Biology (RECOMB) Satellite Workshop on Comparative Genomics: Genomics. The full contents of the supplement are available online at http://www.biomedcentral.com/bmcgenomics/supplements/16/S10.

Authors' details

[1]Laboratoire de Biométrie et Biologie Évolutive, LBBE, UMR CNRS 5558, University of Lyon 1, 43 boulevard du 11 novembre 1918, 69622 Villeurbanne, France. [2]Institut National de Recherche en Informatique et en Automatique (INRIA) Grenoble Rhône-Alpes, 655 avenue de l'Europe, 38330 Montbonnot, France.

Published: 2 October 2015

References

1. Liberles DA: *Ancestral Sequence Reconstruction* Oxford University Press, Oxford, UK; 2007.
2. Bos KI, Schuenemann VJ, Golding GB, Burbano HA, Meyer M, Schmedes S, Wood J, Earn DJD, Herring DA: A draft genome of Yersinia pestis from victims of the Black Death. *Nature* 2011, **478(7370)**:506-510.
3. Szöllősi GJ, Boussau B, Abby SS, Tannier E, Daubin V: Phylogenetic modeling of lateral gene transfer reconstructs the pattern and relative timing of speciations. *Proceedings of the National Academy of Sciences of the United States of America* 2012, **109(43)**:17513-8.
4. Blanchette M, Green ED, Miller W, Haussler D: Reconstructing large regions of an ancestral mammalian genome in silico. *Genome research* 2004, **14**:2412-2423.
5. Sankoff D: Mechanisms of genome evolution: models and inference. *Bulletin of international statistical institute* 1989, **47**:461-475.
6. Ma J, Zhang L, Suh BB, Rany BJ, Burhans RC, Kent WJ, Blanchette M, Haussler D, Miller W: Reconstructing contiguous regions of an ancestral genome. *Genome Res* 2006, **16**:1557-1565.
7. Bérard S, Gallien C, Boussau B, Szöllősi GJ, Daubin V, Tannier E: Evolution of gene neighborhoods within reconciled phylogenies. *BMC Bioinformatics* 2012, **28(18)**:382-388.
8. Hu F, Lin Y, Tang J: MLGO: phylogeny reconstruction and ancestral inference from gene-order data. *BMC Bioinformatics* 2014, **15**:354-9.
9. Gaucher E, Thomson JM, Burgan MF, Benner S: Inferring the palaeoenvironment of ancient bacteria on the basis of resurrected proteins. *Nature* 2003, **425(6955)**:285-8.
10. Benner S, Caraco MD, Thomson JM, Gaucher Ea: Planetary biology-paleontological, geological, and molecular histories of life. *Science* 2002, **296(5569)**:864-8.
11. Higuchi R, Bowman B, Freiberger M, Ryder OA, Wilson AC: DNA sequences from the quagga, an extinct member of the horse family. *Nature* 1984, **312**:282-284.
12. Saiki RK, Scharf S, Faloona F, Mullis KB, Horn GT, Erlich H, Arnheim N: Enzymatic amplification of beta-globin genomic sequences and restriction site analysis for diagnosis of sickle cell anemia. *Nature* 1985, **230**:1350-1354.
13. Shendure J, Ji H: Next-generation DNA sequencing. *Nature biotechnology* 2008, **26(10)**:1135-45.
14. Harris TD, Buzby PR, Babcock H, Beer E, Bowers J, Braslavsky I, Causey M, Colonell J, Dimeo J, Efcavitch JW, Giladi E, Gill J, Healy J, Jarosz M, Lapen D, Moulton K, Quake SR, Steinmann K, Thayer E, Tyurina A, Ward R, Weiss H, Xie Z: Single-molecule DNA sequencing of a viral genome. *Science* 2008, **320(5872)**:106-9.
15. Miller W, Drautz DI, Ratan A, Pusey B, Qi J, Lesk AM, Tomsho LP, Packard MD, Zhao F, Sher A, Tikhonov A, Raney B, Patterson N, Lindblad-Toh K, Lander ES, Knight JR, Irzyk GP, Fredrikson KM, Harkins TT, Sheridan S, Pringle T, Schuster SC: Sequencing the nuclear genome of the extinct woolly mammoth. *Nature* 2008, **456(7220)**:387-90.
16. Orlando L, Ginolhac A, Raghavan M, Vilstrup J, Rasmussen M, Magnussen K, Steinmann KE, Kapranov P, Thompson JF, Zazula G, Froese D, Moltke I, Shapiro B, Hofreiter M, Al-rasheid KAS, Gilbert MTP, Willerslev E: True single-molecule DNA sequencing of a pleistocene horse bone. *Genome research* 2011, **21**:1705-1719.
17. Orlando L, Ginolhac A, Zhang G, Froese D, Albrechtsen A, Stiller M, Schubert M, Cappellini E, Petersen B, Moltke I, Johnson PLF, Fumagalli M, Vilstrup JT, Raghavan M, Korneliussen T, Malaspinas AS, Vogt J, Szklarczyk D, Kelstrup CD, Vinther J, Dolocan A, Stenderup J, Velazquez AMV, Cahill J, Rasmussen M, Wang X, Min J, Zazula GD, Seguin-Orlando A, Mortensen C, Magnussen K, Thompson JF, Weinstock J, Gregersen K, Rø ed KH, Eisenmann V, Rubin CJ, Miller DC, Antczak DF, Bertelsen MF, Brunak Sr, Al-Rasheid KaS, Ryder O, Andersson L, Mundy J, Krogh A, Gilbert MTP, Kjæ r K, Sicheritz-Ponten T, Jensen LJ, Olsen JV, Hofreiter M, Nielsen R, Shapiro B, Wang J, Willerslev E: Recalibrating Equus evolution using the genome sequence of an early Middle Pleistocene horse. *Nature* 2013, **499(7456)**:74-8.
18. Yoshida K, Schuenemann VJ, Cano LM, Pais M, Mishra B, Sharma R, Lanz C, Martin FN, Kamoun S, Krause J, Thines M, Weigel D, Burbano Ha: The rise and fall of the Phytophthora infestans lineage that triggered the Irish potato famine. *eLife* 2013, **2**:00731.
19. Martin MD, Cappellini E, Samaniego Ja, Zepeda ML, Campos PF, Seguin-Orlando A, Wales N, Orlando L, Ho SYW, Dietrich FS, Mieczkowski Pa,

Heitman J, Willerslev E, Krogh A, Ristaino JB, Gilbert MTP: **Reconstructing genome evolution in historic samples of the Irish potato famine pathogen.** *Nature communications* 2013, **4**:2172.

20. Wagner DM, Klunk J, Harbeck M, Devault A, Waglechner N, Sahl JW, Enk J, Birdsell DN, Kuch M, Lumibao C, Poinar D, Pearson T, Fourment M, Golding B, Riehm JM, Earn DJD, DeWitte S, Rouillard JM, Grupe G, Wiechmann I, Bliska JB, Keim PS, Scholz HC, Holmes EC, Poinar H: **Yersinia pestis and the Plague of Justinian 541-543 AD: a genomic analysis.** *The Lancet Infectious Diseases* 2014, **3099(13)**:1-8.
21. Mendum T, Schuenemann VJ, Roffey S, Taylor GM, Wu H, Singh P, Tucker K, Hinds J, Cole ST, Kierzek AM, Nieselt K, Krause J, Stewart GR: **Mycobacterium leprae genomes from a British medieval leprosy hospital: towards understanding an ancient epidemic.** *BMC genomics* 2014, **15(1)**:270.
22. D'Costa VM, King CE, Kalan L, Morar M, Sung WWL, Schwarz C, Froese D, Zazula G, Calmels F, Debruyne R, Golding GB, Poinar HN, Wright GD: **Antibiotic resistance is ancient.** *Nature* 2011, **477(7365)**:457-61.
23. Appelt S, Fancello L, Le Bailly M, Raoult D, Drancourt M, Desnues C: **Viruses in a 14th-century coprolite.** *Applied and environmental microbiology* 2014, (February).
24. Bramanti B, Thomas MG, Haak W, Unterlaender M, Jores P, Tambets K, Antanaitis-Jacobs I, Haidle MN, Jankauskas R, Kind CJ, Lueth F, Terberger T, Hiller J, Matsumura S, Forster P, Burger J: **Genetic discontinuity between local hunter-gatherers and central Europe's first farmers.** *Science* 2009, **326(5949)**:137-40.
25. Green RE, Krause J, Briggs AW, Maricic T, Stenzel U, Kircher M, Patterson N, Li H, Zhai W, Fritz MHY, Hansen NF, Durand EY, Malaspinas AS, Jensen JD, Marques-Bonet T, Alkan C, Prüfer K, Meyer M, Burbano Ha, Good JM, Schultz R, Aximu-Petri A, Butthof A, Höber B, Höffner B, Siegemund M, Weihmann A, Nusbaum C, Lander ES, Russ C, Novod N, Affourtit J, Egholm M, Verna C, Rudan P, Brajkovic D, Kucan Z, Gusic I, Doronichev VB, Golovanova LV, Lalueza-Fox C, de la Rasilla M, Fortea J, Rosas A, Schmitz RW, Johnson PLF, Eichler EE, Falush D, Birney E, Mullikin JC, Slatkin M, Nielsen R, Kelso J, Lachmann M, Reich D, Pääbo S: **A draft sequence of the Neandertal genome.** *Science* 2010, **328(5979)**:710-22.
26. Meyer M, Kircher M, Gansauge MT, Li H, Racimo F, Mallick S, Schraiber JG, Jay F, Prüfer K, de Filippo C, Sudmant PH, Alkan C, Fu Q, Do R, Rohland N, Tandon A, Siebauer M, Green RE, Bryc K, Briggs AW, Stenzel U, Dabney J, Shendure J, Kitzman J, Hammer MF, Shunkov MV, Derevianko AP, Patterson N, Andrés AM, Eichler EE, Slatkin M, Reich D, Kelso J, Pääbo S: **A high-coverage genome sequence from an archaic Denisovan individual.** *Science* 2012, **338(6104)**:222-6.
27. Stiller M, Baryshnikov G, Bocherens H, Grandal d'Anglade A, Hilpert B, Münzel SC, Pinhasi R, Rabeder G, Rosendahl W, Trinkaus E, Hofreiter M, Knapp M: **Withering away-25,000 years of genetic decline preceded cave bear extinction.** *Molecular biology and evolution* 2010, **27(5)**:975-8.
28. Achtman M, Zurth K, Morelli G, Torrea G, Guiyoule A, Carniel E: **Yersinia pestis, the cause of plague, is a recently emerged clone of Yersinia pseudotuberculosis.** *Proceedings of the National Academy of Sciences of the United States of America* 1999, **96(24)**:14043-8.
29. Haensch S, Bianucci R, Signoli M, Rajerison M, Schultz M, Kacki S, Vermunt M, Weston Da, Hurst D, Achtman M, Carniel E, Bramanti B: **Distinct clones of Yersinia pestis caused the black death.** *PLoS pathogens* 2010, **6(10)**:1001134.
30. Rajaraman A, Tannier E, Chauve C: **FPSAC: fast phylogenetic scaffolding of ancient contigs.** *Bioinformatics* 2013, **29(23)**:2987-2994.
31. Penel S, Arigon AM, Dufayard JF, Sertier AS, Daubin V, Duret L, Gouy M, Perrière G: **Databases of homologous gene families for comparative genomics.** *BMC Bioinformatics* 2009, **10(Suppl 6)**:3.
32. Edgar RC: **MUSCLE: multiple sequence alignment with high accuracy and high throughput.** *Nucleic acids research* 2004, **32(5)**:1792-7.
33. Guindon S, Dufayard JF, Lefort V, Anisimova M, Hordijk W, Gascuel O: **New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0.** *Systematic biology* 2010, **59(3)**:307-21.
34. Löytynoja A, Goldman N: **An algorithm for progressive multiple alignment of sequences with insertions.** *Proceedings of the National Academy of Sciences of the United States of America* 2005, **102(30)**:10557-62.
35. Morelli G, Song Y, Mazzoni CJ, Eppinger M, Roumagnac P, Wagner DM, Feldkamp M, Kusecek B, Vogler AJ, Li Y, Cui Y, Thomson NR, Jombart T, Leblois R, Lichtner P, Rahalison L, Petersen JM, Balloux F, Keim P, Wirth T, Ravel J, Yang R, Carniel E, Achtman M: **Yersinia pestis genome sequencing identifies patterns of global phylogenetic diversity.** *Nature genetics* 2010, **42(12)**:1140-3.
36. Nouhati E, Semeria M, Lafond M, Seguin J, Boussau B, Guéguen L, El-Mabrouk N, Tannier E: *Efficient gene tree correction guided by species and synteny evolution* 2015 [https://hal.archives-ouvertes.fr/hal-01162963].
37. Gueguen L, Gaillard S, Boussau B, Gouy M, Groussin M, Rochette NC, Bigot T, Fournier D, Pouyet F, Cahais V, Bernard A, Scornavacca C, Nabholz B, Haudry A, Dachary L, Galtier N, Belkhir K, Dutheil JY: **Bio++: Efficient Extensible Libraries and Tools for Computational Molecular Evolution.** *Molecular Biology and Evolution* 2013, **30(8)**:1745-1750.
38. Chauve C, El-Mabrouk N, Guéguen L, Semeria M, Tannier E: **Duplication rearrangement and reconciliation: A follow-up 13 years later.** In *Models and Algorithms for Genome Evolution Computational Molecular Biology.* Springer, London;Chauve, T. El-Mabrouk 2013:47-62.
39. Maňuch J, Patterson M, Wittler R, Chauve C, Tannier E: **Linearization of ancestral multichromosomal genomes.** *BMC Bioinformatics* 2012, **13(Suppl 19)**:11.
40. Edmonds J: **Paths trees, and flowers.** *Canad J Math* 1965, **17**:449-467.
41. Morgulis A, Coulouris G, Raytselis Y, Madden TL, Agarwala R, Schäffer AA: **Database indexing for production MegaBLAST searches.** *Bioinformatics (Oxford, England)* 2008, **24(16)**:1757-64.
42. Patterson M, Szöllősi G, Daubin V, Tannier E: **Lateral gene transfer, rearrangement, reconciliation.** *BMC Bioinformatics* 2013, **14(Suppl 15)**:4.
43. Kordi M, Bansal MS: **On the complexity of duplication-transfer-loss reconciliation with non-binary gene trees.** *LNCS* 2015, **9096**:187-198.

doi:10.1186/1471-2164-16-S10-S9
Cite this article as: Duchemin *et al.*: Reconstruction of an ancestral *Yersinia pestis* genome and comparison with an ancient sequence. *BMC Genomics* 2015 **16**(Suppl 10):S9.

2.2.3 The need to correct for independent inferences when looking at genome wide properties

In this article, I demonstrate that the inference of ancestral adjacencies (*via* DeCo) using reconciliations and tree topologies obtained independently (from each other, and from the adjacencies) leads to erroneous results (non-linear ancestral gene order / chromosome).

The proposed moves (*zipping* and *unzipping*) try to correct genes tree topologies and reconciliations based on a constraint on the number of copies of a given gene in an ancestral genome inferred from adjacencies information.

The method proposed here only accounts for gene duplications and losses and is limited to the reconstruction of a unique ancestral species, both in terms of results (*i.e.*, it only linearises one ancestral genome) and in terms of the constraints it takes into account. However it inscribes itself in the broader context of being able to make information from the third form of co-evolution (gene to gene co-evolution, using here adjacencies as a proxy) in the reconstruction of the individual gene history.

Another interesting idea is that *zipping* and *unzipping* are moves that were not applied to a lone gene family: they were applied to sets of contiguous gene families along the genome. This is in keeping with the idea that adjacent genes will be affected together by larger structural mutations (including events of duplication , transfer or loss) and that by inferring each gene history independently we:

1. produce reconciliations that when taken independently are good but when taken together are bad.

2. overestimate the number of duplication, transfer and loss events because some are longer than a gene.

The first point refers to the idea that a given reconciliation satisfies (by definition) the criterion that has been used to infer it, but that this criterion does not account for other gene families with whom the gene co-evolves. As a consequence, and given the uncertainty associated with the inference process (at all levels of the phylogenetic pipeline an error could arise that would affect the final result), there is no guarantee that this reconciliation will still be satisfying when we evaluate it with a criterion accounting for multiple genes (for instance, the implication of this reconciliation in terms of ancestral adjacency graph linearity, as was the case here).

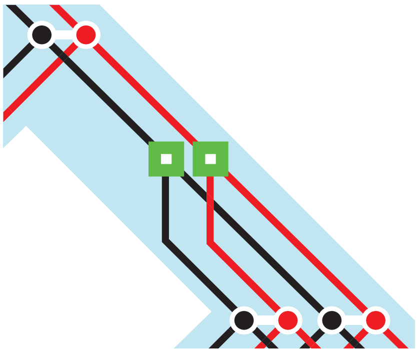


Figure 2.3: Details of two reconciliations (of the red gene and the black gene) with adjacencies at speciation nodes (in white). The parents of each duplication shares an adjacency, and the children of the duplications form couples that share an adjacency, which suggest that these duplications were actually one single duplication that encompassed both genes.

The second point means that to know the total number of times a given event (for instance, a duplication) occurred one cannot just count the number of times this event occurs across all reconciliations but should look at the neighbours of duplicated genes to detect larger events (duplications of several genes together). This is illustrated in Figure 2.3 where considering the red and black genes independently one would count 2 duplication events when the pattern of adjacencies suggests that there was only 1. Note also that as we overestimate the number of events, we are also lead to underestimate their sizes, in terms of number of genes they encompass (which is indirectly related to their size in number of nucleotides).

These two points are connected: as reconciliations are produced independently from each other, they are less likely to share events and this may result in erroneous estimations of the number of evolutionary events that occurred. Consider, for instance, Figure 2.4. The reconciliation of the black gene independently from any other leads to a reconciliation without events (Figure 2.4 **A**). In 2.4 **B**, when considering this reconciliation with the neighbours of the black gene , each seems to undergo an independent duplication and so 2 duplication events are counted. But the alternative reconciliation for the black gene of Figure 2.4 **C** (where a subsequent loss made the traces of a duplication disappear) shows, combined with the patterns of adjacencies, the possibility that only 1 duplication event occurred, encompassing the red, black and green gene at once.

The case shown in Figure 2.4 also raises a question: when considering not only

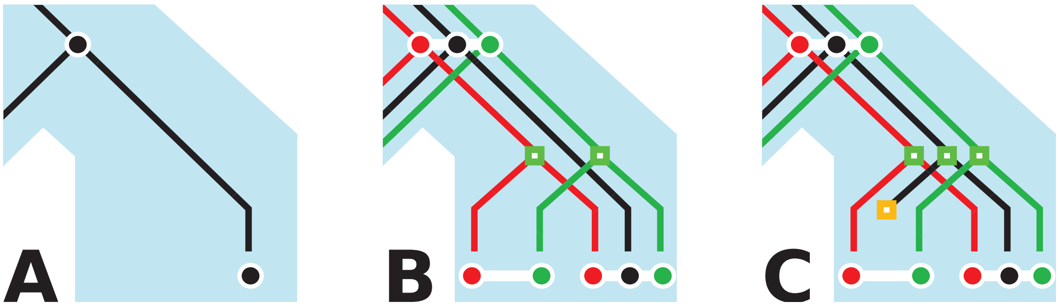


Figure 2.4: **A** The black gene is reconciled independently, without event of duplication or loss along the shown branch. **B** The black gene is considered with its neighbours, the red and the green gene. **C** Alternative reconciliation for the black gene, with an added duplication and a subsequent loss.

one gene, but several linked together by adjacencies, which case do we consider better than the other? Do we prefer a scenario with two independent duplications (Figure 2.4 **B**) or one with a big duplication (encompassing three genes) and a loss (Figure 2.4 **C**)?

It is to answer such a question that I went on to develop a metric that aims to account for both individual gene histories and the history of their relationships and which is the topic of the next chapter.

Chapter 3

Integration of topology, reconciliation and adjacencies for better gene histories

3.1 A global score

3.1.1 Motivation for the definition of a global score

Consider a gene family whose alignment analysis yielded two, equally likely, topologies[1] as shown in the Figure 3.1. I will henceforward refer to this family as the red gene family. I would like to be able to discriminate between these two topologies and to do this I will add information coming from other forms of co-evolution (as the source of the first form of co-evolution, the alignment, led to uncertainties).

Considering first the second form of co-evolution, I consider a DTL parsimony framework, where the costs are:

[1] I consider here only equiprobable possibilities for the sake of the exposition. The same reasoning can (and is, in a latter section) be extended to more complex cases of topology incertitude albeit with significantly more complexity.

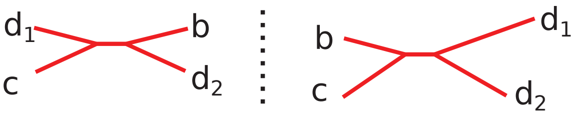


Figure 3.1: Two equally possible topologies for the red gene family.

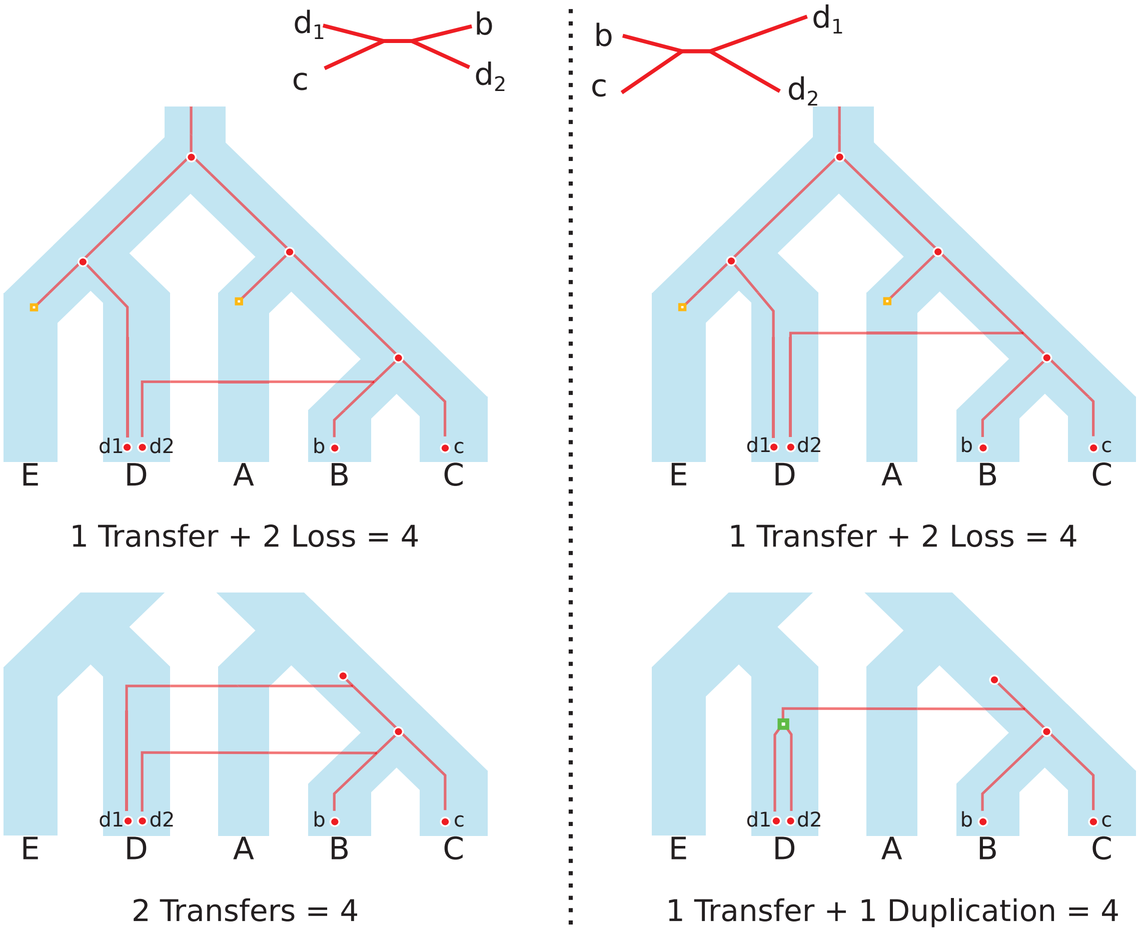


Figure 3.2: For each topology, several equally parsimonious reconciliations are possible.

$$2\ duplication = 2\ loss = 1\ transfer = 2$$

and where the leaves b, c , d_1 and d_2 are respectively associated with the species B, C , D and D. However reconciling yields equiparsimonious reconciliations for both topologies, as shown in Figure 3.2 (only a few of these equally parsimonious reconciliations are shown here), so that I am still unable to determine which topology to favour.

As the first and second form of evolution did not yield sufficient information to choose a topology (and reconciliation), I will evaluate the equiprobable topologies and equiparsimonious reconciliations through the third form of co-evolution, using the framework of evolving adjacencies that was described in the previous chapter.

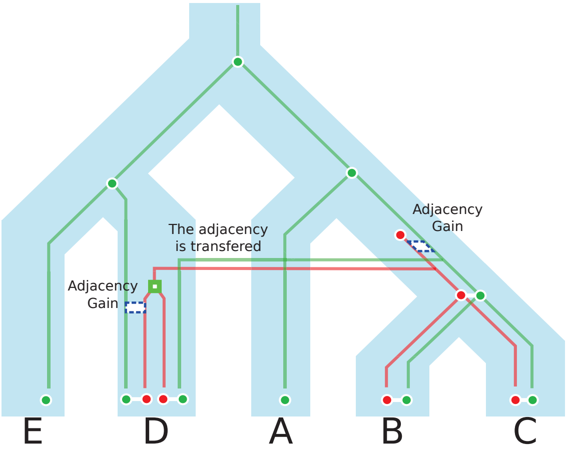


Figure 3.3: Reconciliations of the green gene family and red gene family along with an adjacency history explaining their extant adjacencies. This adjacency history implies two adjacency gains and also that an event of transfer was shared between the families.

Consider the green gene family, which has some leaves that are linked by adjacencies to leaves of the red family as shown in Figure 3.3. To each different reconciliation of the red gene family corresponds a different adjacency history, with differences in the number of adjacency gains and breakages (see Figure 3.4). Furthermore, some of these adjacency histories may imply co-events (*i.e.*, events that occur to the two genes simultaneously) and we thus may want to account for the idea that only 1 event occurred when it was counted twice (each time independently in the red and green family) (this is the case in the two adjacencies history on the right in Figure 3.4).

Considering that:

$$1\ adjacency\ Gain = 2\ adjacency\ Breakage = 2\ transfer = 4$$

And that I account for co-events by counting them as a single event (*e.g.*, $1\ transfer = 1\ co - transfer = 2$), I compute a score composed of the reconciliations costs and the adjacency histories costs for each solution (shown in Figure 3.5). In Figure 3.5, the upper right case shows the lowest score (and is the only one with a score that low), to the correspond topology as the better one for the red gene (the one grouping the genes in species **D**).

This simple example shows how the introduction of information from the third

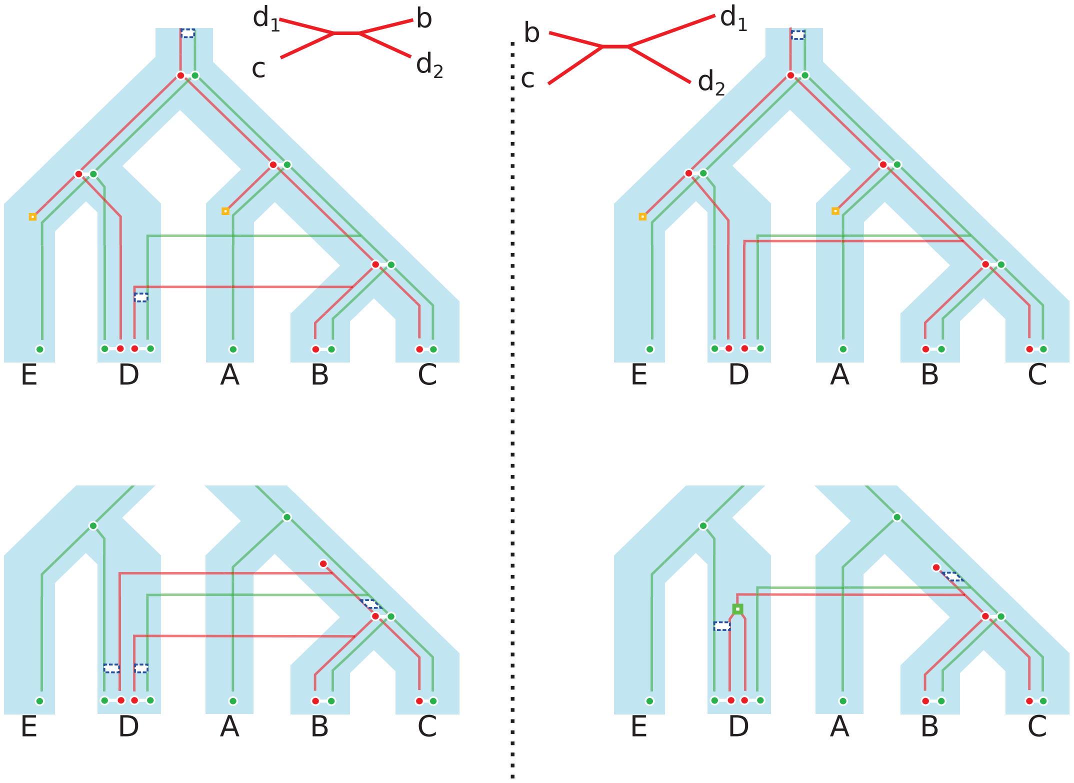


Figure 3.4: To each reconciliation of the red gene family corresponds a different adjacency history.

form of co-evolution (in the form of adjacencies and co-events) can help when the first and second forms of co-evolution produce many equivalent optima. Further than that, Figure 3.5 shows numbers that include score of reconciliations and hint at the idea that the addition of adjacencies may lead us to chose a reconciliation which, when taken in isolation, may not be optimal [2].

Here I just evaluated the red gene family and considered the green family fixed. However it stands to reason that the green family can suffer from the same uncertainties and would benefit from information coming from adjacencies with its neighbours. Taking this into account means that I cannot evaluate and reconstruct the history of each gene family independently, I have to consider them together.

This is why, in order to evaluate together the fit of a set of gene families topologies, reconciliations and adjacencies histories to information coming from the first,

[2] This idea was also touched upon in the last chapter, when for instance *zipping* events would add additional duplications to a reconciliation.

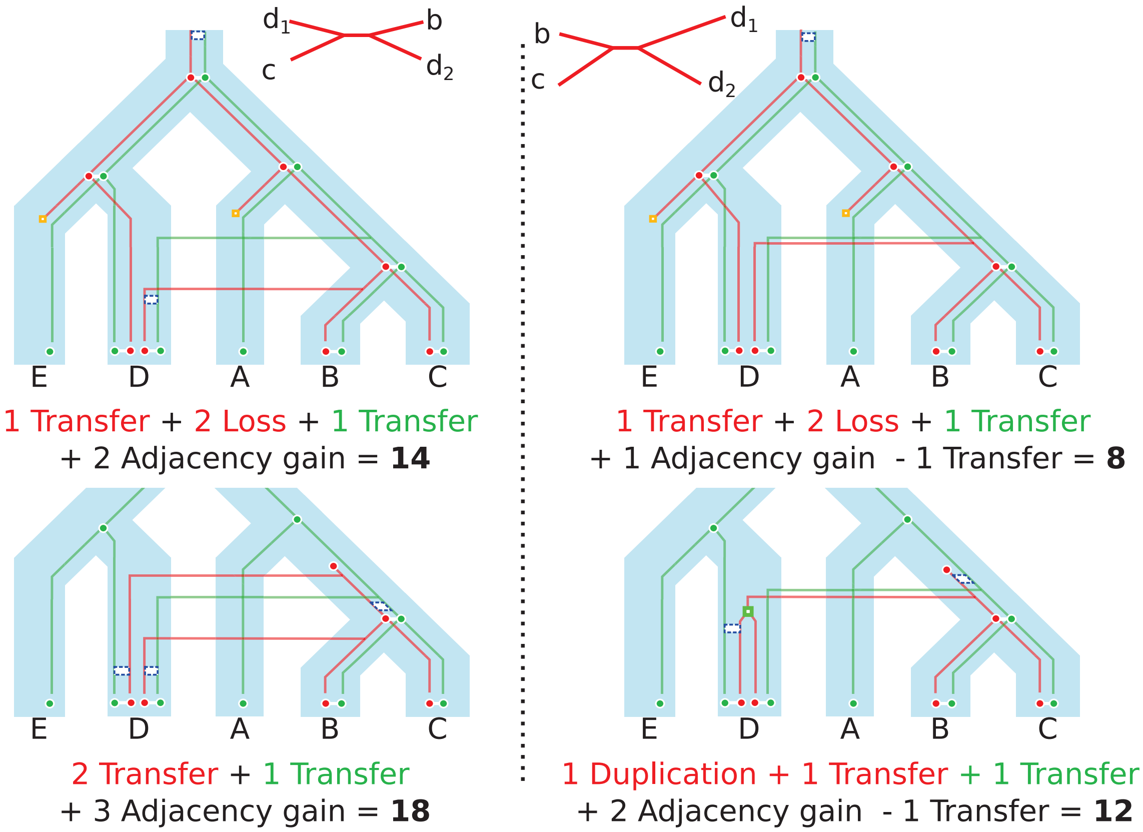

d1
b
c
d2
E
D
A
B
C
1 Transfer + 2 Loss + 1 Transfer
+ 2 Adjacency gain = 14
1 Transfer + 2 Loss + 1 Transfer
+ 1 Adjacency gain - 1 Transfer = 8
2 Transfer + 1 Transfer
+ 3 Adjacency gain = 18
1 Duplication + 1 Transfer + 1 Transfer
+ 2 Adjacency gain - 1 Transfer = 12

Figure 3.5: A score composed of the reconciliation costs (in the color of their respective gene family) and the adjacencies history can be computed for each reconciliation of the red gene. The `-1 Tansfer` accounts for co-transfers of the red and green gene (to compensate for the fact that we counted a transfer in each gene family, we deduce 1 so as to count only 1 co-transfer).

second and third forms of co-evolution, I devise a score which I call *global* because it takes into account all the considered gene families at once and all forms of co-evolution at once.

Schematically, I write the global score as being composed of four parts:

$$\text{Global Score} = \textit{topology} + \textit{reconciliation} + \textit{adjacency} + co - event$$

The *topology* part of the global score evaluates the fit of each gene family with its alignment: this corresponds to information from the first form of co-evolution. The *reconciliation* part of the global score considers the fit to the species tree: this is the second form of co-evolution.

Finally the $adjacency$ and $co-event$ parts of the score represent information coming from the third form of co-evolution: they evaluate the congruence between the reconciliations of gene families that share adjacencies (*i.e.*, are supposed to co-evolve).

3.1.2 Topology

This part of the global score represents the information on the gene family that is coming from the genes sequence alignment.

Rather than estimate sequence support for a given gene tree topology directly from an alignment, I prefer to use a tree distribution that reflects this alignment, such as an *a posteriori* tree distribution obtained via a Bayesian inference method.

This tree distribution is used in the form of a *Conditional Clade Probability distribution (CCP distribution)*. CCP distributions allow to compact a tree distribution in such a way that it is easy to obtain an accurate approximation of the likelihood of a given tree topology according to the original tree distribution [Höhna and Drummond, 2012], which is what interests me.

The choice of using CCP distributions was further motivated by its use in the joint inference of optimal topology and reconciliations, which will be further explained in the section about the reconciliation part of the score.

I will now describe CCP distributions, how to build them from a given tree distribution, and how to use them to approximate the likelihood of a tree.

Building a CCP distribution

As its name entails, a CCP distribution is a distribution of *conditional clade probabilities* [Höhna and Drummond, 2012], which are computed from a tree distribution sample (for instance, an *a posteriori* sample obtained with Bayesian inference method), which I will refer to as $\mathcal{G}$. In this sample, each (unrooted) tree can be described in terms of the clades it contains. In particular the clades shared (or not) between the different trees can be used to compute posterior clade probabilities, which were already introduced in a section of the introduction: they represent the frequencies at which given clades C occur among the trees of $\mathcal{G}$.

In an unrooted bifurcating tree (which is the sort of trees that we consider here), each node that is not a leaf corresponds to a tripartition of the leaves of the tree into three clades. I call π such a tripartition, and $\pi[1]$, $\pi[2]$ and $\pi[3]$ each of the three clades formed by the tripartition (I give them indexes here for the sake of the explanation, but the clades defined by a tripartition are not actually ordered). Furthermore, π is a *split* of C if $\pi[1] \cup \pi[2] = C$. Note that this implies that $\pi[3]$ is the complementary clade of C (in practice, I only note the two first clades formed by the tripartition, as the third one is always the complementary clade of their union).

The *conditional clade probability* of π given C, written $P_{CCP}(\pi|C)$, is the ratio of the number of times we observe the tripartition π in $\mathcal{G}$ divided by the number of times we observe the clade C in $\mathcal{G}$. In other words, it is the observed frequency at which clade C is split according to π (*i.e.*, it is split into $\pi[1]$ and $\pi[2]$). Note that, given a clade C, the sum of the conditional clade probabilities of the set all of its possible splits is equal to 1.

The CCP distribution corresponds to the set of all conditional clade probabilities, for each clade and each tripartition present in $\mathcal{G}$.

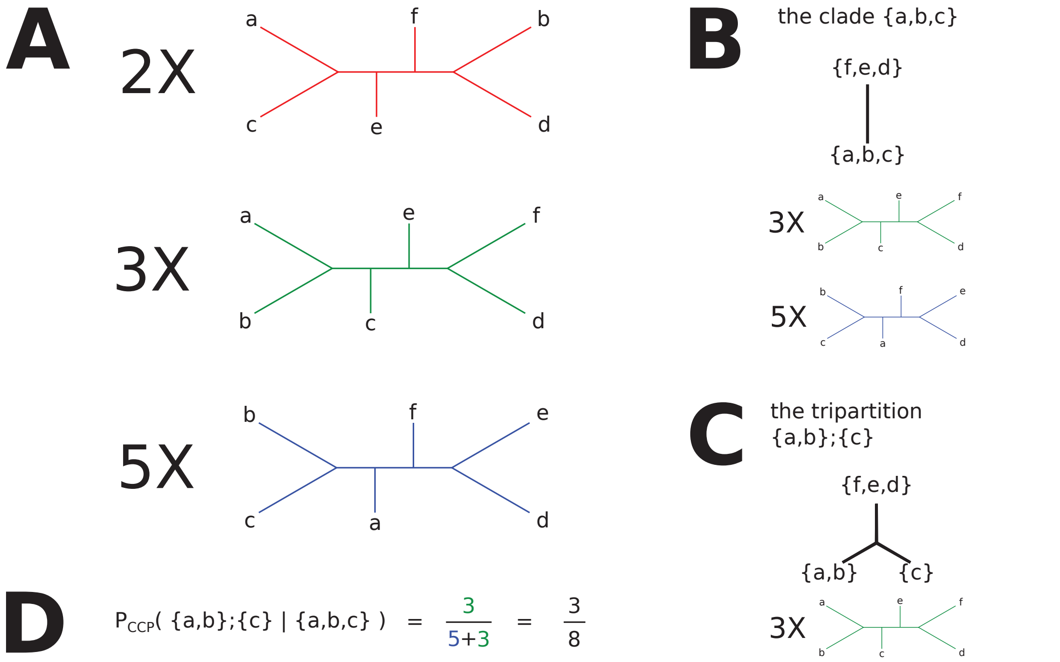


Figure 3.6: **A** A tree sample composed of three different topologies (in red, green and blue). Each topology is represented a certain number of time (respectively, 2, 3 and 5 times). **B** A visual representation of the clade $\{a, b, c\}$ which is represented in the green and the blue topologies (so 8 times in total). **C** A visual representation of the tripartition $\{a, b\}; \{c\}$ which is represented in the green topologies (so 3 times in total). **D** The computation of the conditional clade probability of the split $\{a, b\}; \{c\}$

.

For instance, consider Figure 3.6. Figure 3.6 **A** shows a tree distribution in which is constituted of 10 trees that are split among 3 different topologies. The clade presented in Figure 3.6 **B** is seen 8 times in the 10 trees of the sample (for a posterior clade probability of 0.8). The tripartition $\{a, b\}; \{c\}$ (meaning: $\pi[1] = \{a, b\}$, $\pi[2] = \{c\}$) corresponds to a split of clade $\{a, b, c\}$ (note again that $\pi[3] = \{d, e, f\}$ is implicit at it is the complementary of the union of the two others), as illustrated in Figure 3.6 **C**.

This particular split of $\{a, b, c\}$ is observed a total of 3 times in the 10 trees of the sample of Figure 3.6 **A**. Figure 3.6 **D** follows this definition and shows the computation of the CCP of split $\{a, b\}; \{c\}$ given clade $\{a, b, c\}$ (Note that given the split, the clade is simply the union of the two elements of the split so that C could not be mentioned).

Computing the likelihood of a tree according to a CCP distribution

Given a CCP distribution, estimating the likelihood of an unrooted tree (whose leaves are labelled in the same way as the leaves of the tree sample that was used to build the CCP distribution) is quite straightforward: it is the product of the CCPs of all the clades and splits encountered during a traversal of this this tree.

Computing its likelihood starts by choosing a bipartition in the tree, and then traversing each of the two subtree defined by this bipartition to make the list of the different clades and splits it is composed of to finally retrieve the associated CCPs and multiply them together.

Due to the manner in which CCP distribution are constructed, the chosen initial bipartition does not change the computed likelihood. In practice, it is useful to choose a bipartition that separates a single leaf from the rest of the tree because it means that only one subtree (the one with all leaves but one) needs to be traversed.

Consider for instance the black tree presented in Figure 3.7 **A**. Figure 3.7 **B** details the computation of its likelihood where the initial bifurcations is the one that separates `f` from the rest of the tree.

Note that the black topology of Figure 3.7 **A** does not correspond to any of the topologies presented in Figure 3.6 **A** but that we were still able to compute a likelihood for it. Indeed, an important property of CCP distributions, called *amalgamation* [David and Alm, 2011], is to be able to combine clades in such a way that tree that were not in the original tree sample can be accounted for. However, should

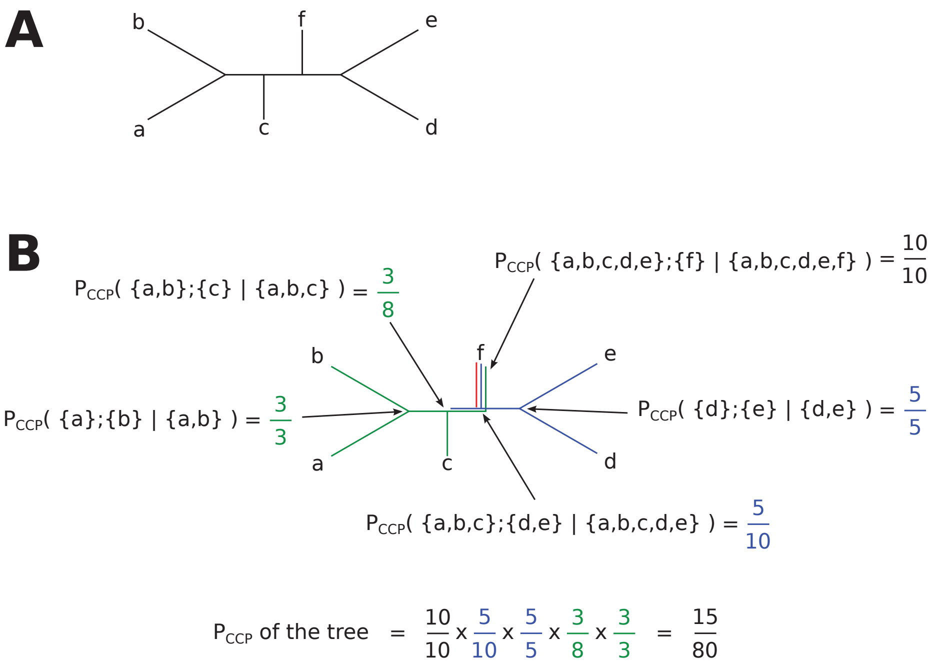

A
b
f
e
a
c
d
B
PCCP({a,b};{c} | {a,b,c}) = 3/8
PCCP({a,b,c,d,e};{f} | {a,b,c,d,e,f}) = 10/10
b
f
e
PCCP({a};{b} | {a,b}) = 3/3
PCCP({d};{e} | {d,e}) = 5/5
a
c
d
PCCP({a,b,c};{d,e} | {a,b,c,d,e}) = 5/10
PCCP of the tree = 10/10 x 5/10 x 5/5 x 3/8 x 3/3 = 15/80

Figure 3.7: **A** a black topology whose likelihood we want to estimate. **B** the computation of the black topology's likelihood using the CCP distribution presented in Figure 3.6. Starting at the bifurcation separating leaf `f` from the rest of the tree, the colors represent which topologies of the CCP distribution support the black topology.

a tree contains a clade (or a split) not present in the CCP distribution, then its likelihood according to said CCP distribution shall be 0 (or, alternatively, suffer an arbitrary penalty for each clade of the tree that are absent from the CCP). Henceforward, I note the likelihood of tree T according to a CCP distribution $P_{CCP}(T)$.

Back to the score

Following definitions of [Scornavacca *et al.*, 2014] concerning the part of their score that was topology related, I define the contribution of the topology T of a given gene family to the global score as:

$$w_{topology} \times -log(\frac{P_{CCP}(T)}{P_{CCP}(T_{max})})$$

where T_{max} is the tree with the maximum likelihood according to the CCP distribution and $w_{topology}$ is the weighting factor of the topology part of the global score.

Considering all N gene families, the overall topology part of the score is:

$$\textit{topology} \quad = w_{topology} \times \sum_{i=0}^{N} \Big(- log(\frac{P_{CCP}(T_i)}{P_{CCP}(T_{i\ max})}) \Big)$$

Where T_i refers to the tree of the i-th gene family (and similarly with $T_{i\ max}$).

3.1.3 Reconciliation

This part of the global score represents the information on the gene family that is coming from the second form of co-evolution: co-evolution with the species tree. In this section, I describe the actual formula I use in the global score to account for reconciliations and then I discuss two algorithms: the first actually minimizes the reconciliation part of the score, while the second jointly minimizes the topology and reconciliation part of the score and was the inspiration for these parts.

The reconciliation part of the score

In the global score, the part corresponding to individual reconciliations is fairly straightforward: it corresponds to the weighted sum of all the individual events present across all reconciliations. Indeed, it is written as:

$$\textit{reconciliation} \quad = w_{reconciliation} \times \big(c_{dup}.n_{dup} + c_{loss}.n_{loss} + c_{tr}.n_{tr} \big)$$

where $w_{reconciliation}$ is the weight of the reconciliation part of the global score, c_x and n_x are respectively the cost of a single event x and the number of times an event x is seen across the reconciliations of all gene families. x can take three values: *dup*, *tr* and *loss* corresponding respectively to a gene duplication, a gene transfer (technically a transfer reception) and a gene loss.

Most parsimonious reconciliation of gene tree / species tree in a duplication, loss and transfer context: Doyon *et al.* [2010]

If computing the contribution of a single reconciled gene tree to the global score is quite trivial, obtaining said reconciliation is not, especially in a DTL context. What follows is a discussion on the algorithm presented in Doyon *et al.* [2010], as it forms the basis for the TERA [Scornavacca *et al.*, 2014] algorithm that is implemented in the same package as DeCoSTAR.

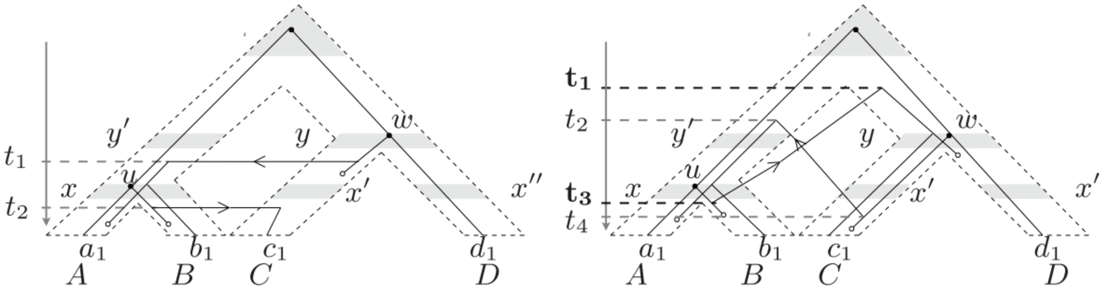


Figure 3.8: Figure 1 from [Doyon *et al.*, 2010]: "Two scenarios for a gene tree G (plain lines) along a species tree S (tubes), where the symbol ○ represents loss. **(Left)** A time consistent scenario. **(Right)** A scenario that is not time consistent: the transfer from the donor at t_3 (resp. t_4) to a receiver at t_1 (resp. t_2) implies that u predates (resp. follows) w".

This algorithm builds the most parsimonious reconciliation of a gene tree with an ultrametric species tree in a DTL context using a dynamic programming approach, much like DeCoSTAR. This algorithm mainly defines a matrix whose rows and columns respectively correspond to nodes of the gene and species tree, and a case of the matrix with row u and column x contains the cost of reconciling the subtree of the gene tree rooted at node u such that u is in species x. This approach allows efficient computations as it makes the hypothesis that the reconciliation of a given lineage does not depend on the reconciliation of its sister lineages.

However this hypothesis is linked with several limitations of these methods, including the impossibility to include gene conversion, horizontal tranfers with homologous recombination or the possibility to infer scenarios that are not *time consistent*.

Time consistency refers to the production of reconciliations that contains contradiction in the order of the nodes of the species tree that they imply (remember that a transfer is indicative of the co-existence in time of the donor and receiver species and thus implies a temporal constraint on the species tree nodes), as illustrated in Figure 3.8.

The method proposed in Doyon *et al.* [2010] bypasses the time consistency by requiring an ultrametric species tree (a species where branch lengths correspond to time) as it contains the (supposed) real order between the different speciations. More precisely, the method use a *subdivided species tree* where the different nodes of the species tree have been associated with a *time slice* corresponding to their index in the list of the nodes ordered by height in the tree (and where all the leaves are

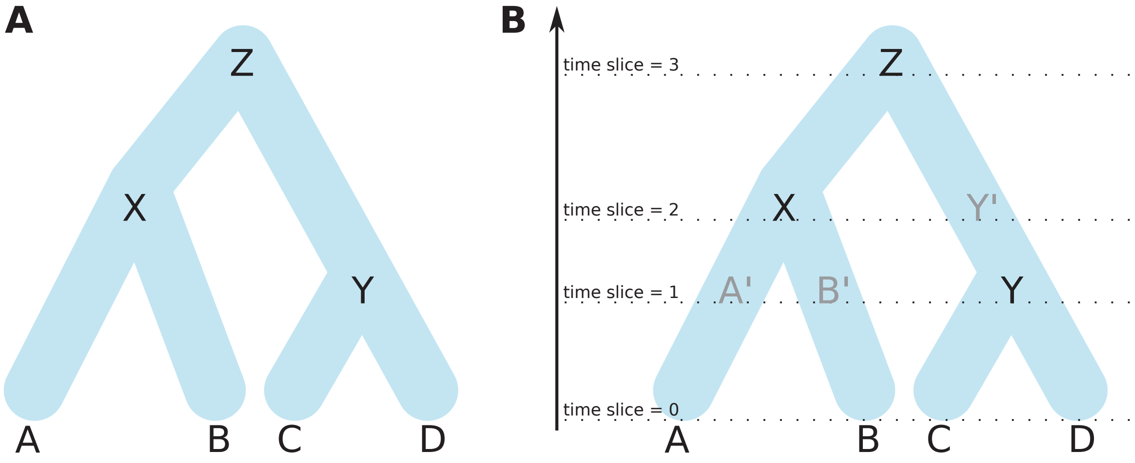


Figure 3.9: **A** Ultrametric species tree with internal nodes (`X`,`Y` and `Z`). **B** The same ultrametric species tree, subdivided into time slices. The added artificial nodes are represented in gray (`A'`, `B'` and `Y'`).

considered to correspond to the same time slice: 0) and additional nodes (called *artificial nodes*, because they only have 1 child and were not part of the original tree) are created along some branches to ensure that the time slice of a node and its parent differ by exactly 1. An example of such a subdivided species tree is shown in Figure 3.9 where the time slices associated to the different nodes correspond to the dashed line they are found on. This representation also symbolizes the idea that nodes associated to the same time slice are considered simultaneous (for instance in the figure, `A'`, `B'` and `Y` are supposed to occur at the same time).

Only allowing transfers between species tree nodes that are associated to the same time-slice forces the gene lineages to follow the defined order of speciations in the species tree and thus ensures the time consistency of reconciliations.

The original algorithm presented in Doyon *et al.* [2010] only treats transfer between two species present in the species tree. However, as mentioned earlier, transfers may come from species absent in the species tree (because they are extinct or that their extant descendants are not part of the species tree), which I shall refer to as *dead* species, for clarity purposes. Arguably, it could be said that it is more likely that they come from such a dead lineage rather than a sampled one [Szöllősi *et al.*, 2013b] (because, at any point in time, it is supposed that there are considerably more species absent from the species tree than species that are present in it).

The original algorithm can be easily adapted to consider such *lateral gene transfer from the dead* (to refer to the 2013 article) by adding a special, subdivided, branch

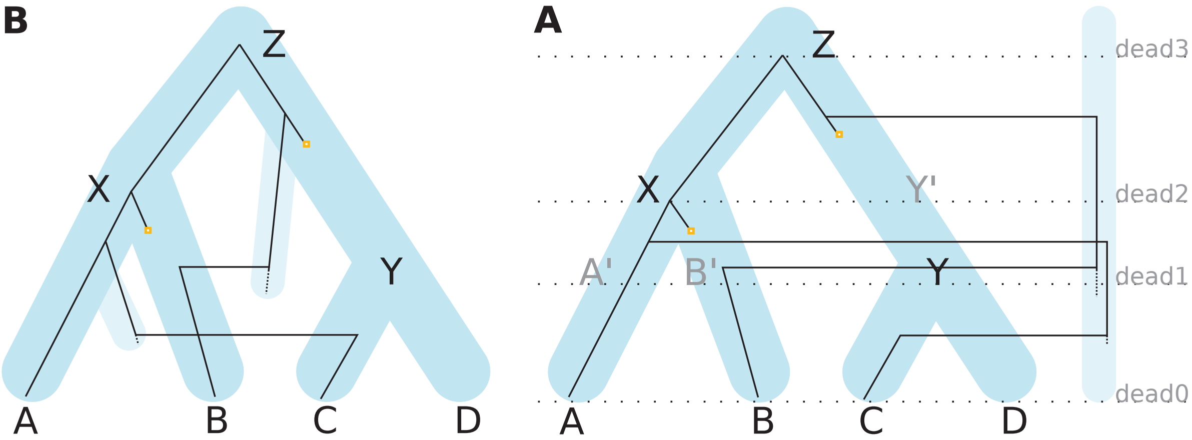


Figure 3.10: **A** Reconciliation of a black gene tree in the species tree with transfers from extinct/unsampled lineages (lighter blue). **B** The same reconciliation, representing all dead lineages with a single `dead` branch (lighter blue), subdivided to comply with the time slices.

alongside the species tree which represent the dead lineages (see Figure 3.10). This branch is special as it can receive transfer freely (as a transfer toward this branch is actually a speciation toward a dead lineage), and losses in it are free (as they are considered to arise from the extinction of the species the gene evolves in). Additionally, an event where a gene lineage bifurcates (*i.e.*, divides into two) inside the dead species can be considered (not represented in the figure). Such an event could result from a duplication of the gene in a dead species, a transfer from a dead species to another, or a speciation of dead species. In a parsimonious context a speciation would be preferred (as they usually cost nothing) however I generally refer to such an event as a *Bifurcation Out* (Bifurcation Outside the species tree) to underline the idea that we are usually unable to distinguish between the different scenarios.

Joint optimization of topology and reconciliation: TERA

As mentioned earlier, the TERA algorithm [Scornavacca *et al.*, 2014] offers a mean to compute the reconciliation that jointly minimizes the discord with the species tree and the alignement. It does so using the idea developed by Szöllősi *et al.* [2013a] (called the ALE approach) which proposes formulas for the joint likelihood of a DTL reconciliation and its topology given a species tree, a model of reconciliation and a gene CCP distribution (which represents the information of the gene alignment).

By coupling the manner in which ALE explores the set of possible topologies (through the CCP distribution) with the algorithm of Doyon *et al.* [2010], Scornavacca *et al.* [2014] describe an algorithm that reconciles a gene CCP distribution with a species tree in a parsimonious DTL framework.

Much of the original algorithm remains identical, however in the cost matrix (and corresponding structures), where rows corresponded to nodes in the gene tree, rows now correspond to clades of the CCP distribution. Furthermore, whenever a bifurcation is implied (for instance, when computing scores for events of speciation, duplication or transfer) all possible splits of the clade are tested and weighted according to their CCP (actually, the opposite of the logarithm of the CCP, normalized).

Note also that the original algorithm requires an ultrametric (subdivided) species tree. TERA's algorithm can make use of such a tree, but also authorizes a non-ultrametric species tree, at the cost of sometimes producing time-inconsistent reconciliations.

The score that is defined and optimized by the TERA algorithm, when summed across all gene families, actually corresponds to the topology and reconciliation part of my global score (although I use an additional parameter to weight the reconciliation part of the score).

3.1.4 Adjacency

This part corresponds to information from the third form of co-evolution. It follows the idea that adjacencies can be used to represent the link between genes, and that the histories of adjacencies, in particular the number of gains and breakages of adjacencies, can be taken as a measure of congruence between two reconciled gene trees in the subtrees where they are linked by adjacencies (*i.e.*, the part where they are supposed to co-evolve). Thus, I define the adjacency part of the global score as:

$$adjacency \ = w_{adjacency} \times \left(c_{Gain}.n_{Gain} + c_{Break}.n_{Break}\right)$$

where c_{Gain} (respectively, c_{Break}) represents the cost of a single adjacency gain (resp. breakage), n_{Gain} (resp. n_{Break}) represents the number of adjacency gains (resp. breakages) across all adjacency histories (that is, all adjacency equivalence classes) and $w_{adjacency}$ represents a scaling factor for the global score.

Using the DeCo algorithm already introduced in the previous chapter I am able to reconstruct adjacencies histories given reconciled gene trees and extant adjacencies, such that they minimize a linear combination of gains and breakages. In other words, considering fixed extant adjacencies, I am able to associate a score of adjacency gains and breakages to the reconciliations of the genes that are parts of adjacencies, which amounts to an evaluation of reconciliations (and topologies) according to a gene-to-gene co-evolution criterion.

3.1.5 Co-event

A co-event is an event, such as a duplication or a loss for instance, that encompasses several genes at once. While this idea can be understood as to represent segmental events (*e.g.*, the duplication of a chromosome fragment containing several genes), it may also take on the sense of two separate events whose fitness is interdependent (*e.g.*, the transfer of a protein is more likely to be retained by evolution if a co-evolutionary partner that is necessary to its function is also transferred). In any case, both visions come down to the idea that a co-event (*e.g.*, 1 duplication of 3 genes) will have a different probability to be observed than the probability of observing independently an event of the same nature in each individual genes it encompasses (*e.g.*, 3 duplications of 1 gene each) (in the first case because its probability to occur is different, in the other because its fitness is different).

However, as each reconciliation is computed independently, we can be lead to see some events as independents when they may actually be part of the same co-event, an idea that was discussed in the last section of the previous chapter, and that can be seen also in the reconciliation part of the global score, which counts events across all reconciliations, implicitly making the hypothesis of independence of events.

The co-event part of the score is there is to correct biases implied by this hypothesis and introduce yet more information from the third form of co-evolution in the global score. The first step to define this part of the score is the detection of co-events.

Detecting co-events

In my framework, I define a co-event using adjacencies, and in particular the adjacencies history computed using DeCoSTAR. Indeed, the formulas of DeCoSTAR explicitly define cases which implies that two evolutionary events (*i.e.*, nodes in reconciled gene trees) occurred simultaneously (In Table 1 of the DeCoSTAR article, such cases are referred to as c_{1SYNCH}, for synchronous). If these nodes bear events of the same nature, then the synchronous formulas correspond to the case where the nodes are actually part of the same co-event.

From this definition, I can actually define a co-event of type e as a graph where each node corresponds to a node bearing event e in a reconciliation and edges link together nodes backtracked using a c_{1SYNCH} formula by DeCoSTAR (note that this implies that all the nodes of a given co-event are in the same species, and time slice when applicable). Henceforward, I refer to such a graph as a co-event graph.

The co-event part of the global score

The cost of a co-event of a given type can be written in numerous way, for instance considering the number of genes it contains. I will consider here that 1 event cost the same independent of the number of genes it contains (this means that a co-duplication of 3 gene costs the same as a duplication of 1 gene). So the contribution to the global score of a single co-event whose event is e and which contains n genes is:

$$w_{reconciliation} \times (c_e - c_e.n)$$

which can be re-written as:

$$w_{reconciliation} \times (-c_e.(n-1))$$

where $w_{reconciliation}$ is the reconciliation a global score scaling factor[3], c_e is the cost of a single e event and e takes the value *dup*, *loss* or *tr*, representing respectively a duplication, a loss or a transfer.

This formula considers that, in the reconciliation part of the global score, every individual event that is part of the co-event was already counted independently, hence the negative term that *reimburses* the cost of these events.

[3] The scaling factor applied here is that of the reconciliation because co-events are considered to be of the same nature as single events, despite the fact that they represent a different form of co-evolution.

With this definition, the formula of the co-event part of the global score can be written as:

$$co - event \ = $$
$$w_{reconciliation} \times \Big(\sum_{i=0}^{D} \big(- c_{dup}.(n^i_{dup} - 1) \big) + \sum_{i=0}^{L} \big(- c_{loss}.(n^i_{loss} - 1) \big) + \sum_{i=0}^{T} \big(- c_{tr}.(n^i_{tr} - 1) \big) \Big)$$

where D (respectively, L, T) is number of co-duplication (resp. co-loss, co-transfer) events, c_e and e are as before, and n^i_e is number of e events that the i-th co-event of type e encompasses.

3.1.6 Explicit formulation of the global score

Given all the definitions above, the global score for a set of gene family topologies, reconciliations and adjacency histories given CCP distributions for each gene family, a species tree, extant adjacencies, costs for each events and scaling factor of the different parts is:

$$w_{topology} \times \sum_{i=0}^{N} \Big(- log(\frac{P_{CCP}(T_i)}{P_{CCP}(T_{i\ max})}) \Big) +$$

$$w_{reconciliation} \times \big(c_{dup}.n_{dup} + c_{loss}.n_{loss} + c_{tr}.n_{tr} \big) +$$

$$w_{adjacency} \times \big(c_{Gain}.n_{Gain} + c_{Break}.n_{Break} \big) +$$

$$w_{reconciliation} \times \Big(\sum_{i=0}^{D} \big(- c_{dup}.(n^i_{dup} - 1) \big) + \sum_{i=0}^{L} \big(- c_{loss}.(n^i_{loss} - 1) \big) + \sum_{i=0}^{T} \big(- c_{tr}.(n^i_{tr} - 1) \big) \Big)$$

I call such a set of gene families topology, reconciliations and adjacency histories a *solution*. The global score allows us to consider that a solution is better than another when its associated score is lower. The different part of the score posit that a solution can be better than another even if one part of it is worse (for instance, there is more events in the reconciliations), provided the other parts of the score compensate for this difference (for instance, all new events become part of co-events and the topologies show more agreement with the CCP distribution).

My objective now becomes to be able to find a solution that minimizes the global score and the next section describes the approach that I chose to do so.

3.2 Optimizing the score

Finding a solution that optimizes the global score can safely be assumed not to be trivial. For once, an exhaustive search is not feasible in practice. To convince oneself of this, one can consider the size of the space of solutions to explore. For each family, there is multiple possible topologies to consider (up to the number possible number of topologies for the number of leaves of this family), and for each such topology, there exists several reconciliations (potentially an infinite number of them, although this would imply events such as a duplication directly followed by a loss). It follows that testing each possible combinations of topology and reconciliation across all families quickly becomes intractable. Furthermore, one should consider that for a given set of reconciliations, there can be more than one set of adjacency histories and co-events, adding yet more dimensions to an already big space.

On the other hand, simply considering the problem of reconstructing the reconciliation (with fixed topologies) and adjacencies history for two gene families jointly is a hard problem, for which solutions have been searched without success so far[4], although no formal proof of NP-hardess of the general problem, nor any of its particular cases, is known. In light of this a heuristic approach seems fitting for the optimization of the global score (which integrates more gene families and additional criterions: topology and co-events) on a potentially large number of gene families.

It is however worth noting that given the size of the solution space, I can presume that in practice the global optimal solution will never be found. What matters to

[4] Personal communications with Vincent Berry, Céline Scornavacca, Sèverine Bérard and Eric Tannier.

me is that I get new gene trees and reconciliation that are better that the initial ones in a biological sense, using metrics included in the global score or not (*e.g.*, the inference of linear ancestral chromosomes, such as I did in the article about *Yersinia pestis* ancestral reconstruction). In that sense there actually is no guarantee that the solution that minimizes the global score yields the best gene trees, but I suspect that solutions associated with a lower global score will contain on average better gene trees.

3.2.1 A Gibbs sampling-like approach

Consider a solution: a set of topologies, reconciliations, adjacencies histories and co-events. This solution corresponds to a given score and I will here refer to them as the current solution and current score.

I choose the gene family to be the basis of a *local move* in the solution space. Thus, a local move consists in the modification of the topology and/or reconciliation of a single gene family in the solution. This modification affects the histories of the adjacencies this gene family is an extremity of, and consequently the co-events are impacted. In short, the local move can affect each part of the global score and the solution *post local move* (which I will call the new solution) corresponds to a score (the new score) different from the current score. Then, I chose whether or not I accept the new solution based on the difference between the new and current score. If the new solution is accepted, then it replaces the current solution. Otherwise the current solution stays the same.

A new score that is lower than the current score means that the new solution is better than the current solution (*i.e.*, the local move leads to a decrease in the global score); the converse can also be held to be true. However, as I expect that there exists local optima (*i.e.*, solutions which do not minimize the global score but that are better than all the solutions they can reach in one local move) among the solutions, I use an approach where it is possible to accept a new solution even though it worsens the score (*i.e.*, increases it).

If the new score is lower than the current score, then I automatically accept the new solution. Otherwise I define the probability to accept the new solution as:

$$e^{\frac{S_{current}-S_{new}}{Temperature}}$$

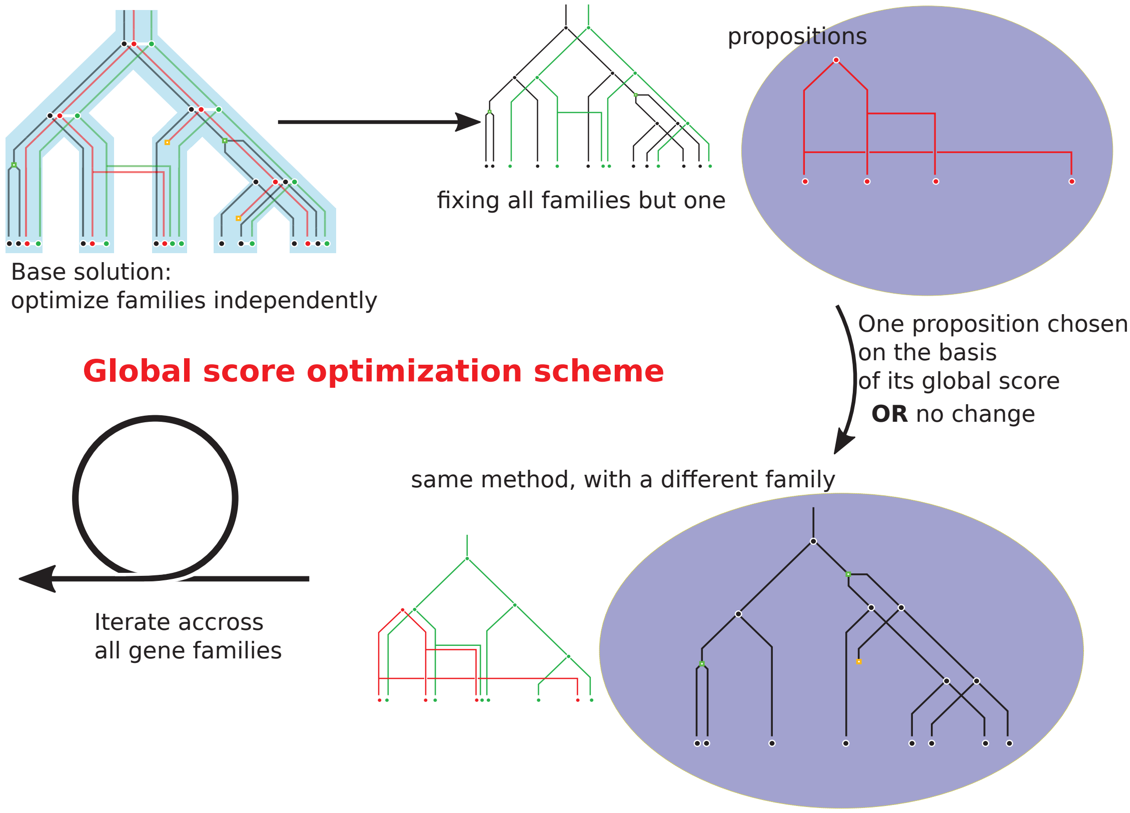


Figure 3.11: The proposed method for the optimization of the global score.

where $S_{current}$ is the current score, S_{new} is the new score and $Temperature$ is a scaling factor. $Temperature$ is named as it is because given a fixed difference between the current and new score, a higher value of $Temperature$ will result in a higher probability to accept the new (worse) solution, like a physical temperature which heightens the probability of high energy states.

Whether or not the local move was accepted, we can then try another local move in another gene family and, local move by local move, explore the space of solutions in a manner biased toward solutions with lower scores. This strategy reminds of Gibbs sampling as it aims to explore a large multivariate space by changing the value of one variable at a time (where variables correspond to gene families here).

The proposed local move operates at the scale of one gene family; during the exploration of the space of solutions, I prefer to use the term *rounds*, where a round corresponds to an iteration of local moves across all the gene families.

Figure 3.11 sums up the chosen optimization strategy.

3.2.2 Implementation

In practice, the different elements for the computation global score and the proposition and evaluation of moves were implemented using the version of the ecceTERA [Jacox *et al.*, 2016] which contains DeCoSTAR as a backbone.

Given a current solution, once a new topology / reconciliation is proposed the adjacency histories that need to be changed are computed using a version of DeCoSTAR modified to favour adjacency histories containing co-events. This adjacencies histories are then parsed to count the number of adjacency gains and breakages and detect co-events.

The modification of DeCoSTAR, already described in Jean [2013], amounts to explicitly recognize during the computation of the cost matrix the cases implying a co-event, tag them in order to report them during backtrack and subtract the (properly scaled) cost of a single event of loss, duplication or transfer (depending on the case) directly to their cost. When the cost of a co-event is considered equal to the cost of a single event of the same nature and when co-event are supposed linear, this modification effectively reflects the final cost of co-events across all gene families and I thus jointly optimize the adjacency and co-event part of the score (given fixed reconciliations). Otherwise this just represents a heuristic that favours the formation of co-events, and may even cause an overestimation of them, which is why I programmed this modification as an option which can be toggled off whenever the described conditions are not met[5].

As for the local move, the proposition of a new topology and reconciliation for a gene family, I propose several methods to do it in the next section.

3.3 Proposing new topologies and reconciliations for a gene family

The chosen method for the optimization of the global score is inspired by Gibbs sampling. Gibbs sampling consists in choosing a new solution for the variable parameter according to its probability distribution conditioned on the value of all the other parameters. In my case, that would come down to be able to sample a gene

[5] Remember however that DeCoSTAR's costs formulas correspond to the gene order problem (where a linear solution is expected) anyway.

family topology and reconciliation given the species tree, all the other gene families reconciliations, and adjacencies (or, in other words, directly according to the global score).

While there exists no formal proof that this would constitute a hard problem (in the sense of the algorithmic complexity theory), I have already mentioned that several groups have attempted, since 2012 and the original DeCo publication ([Bérard *et al.*, 2012]), to solve the (smaller) problem of jointly optimizing the reconciliation and adjacency scores (which are only two of the four components of the global score) for a couple of gene family with no success (the usual dynamic programming used for optimizing each part of the score does not generalize), which leaves little hope about the more general problem.

Therefore I have to rely on approximations to propose a new solution for a single family. In the next sections I propose several methods to propose new solutions. I start with a method that is blind to the global score (it does undirected moves) and then describe methods that gradually add information from different parts of the global score to direct them.

3.3.1 Sampling uniform random trees

A very basic way to propose a new topology is just choose one at random among all the possible topologies with a number of leaves equal to that of the gene family considered[6].

Even if it is chosen randomly, the new topology must be evaluated by the CCP distribution of the gene family in order to evaluate the topology.This can be done using the procedure I discussed in a previous section, with special attention toward the possibility to apply an arbitrary "penalty" multiplier when encountering a clade absent from the CCP if one wants to be able to explore solutions whose topologies are absent from the original CCP distributions[7] (this may be desirable if, for instance, the original CCP distributions were made from very few trees).

There already exists numerous methods to choose a random topology and I will here describe one that is based on clades and splits, mainly because I will use

[6] See the part about phylogenetic tree jargon for a definition of the number of possible topologies given a fixed number of leaves.

[7] Indeed, not using this possibility comes down to assign a likelihood of 0 to the tree, which translates into a topology part of the score of infinite value, and thus a solution that may not be selected.

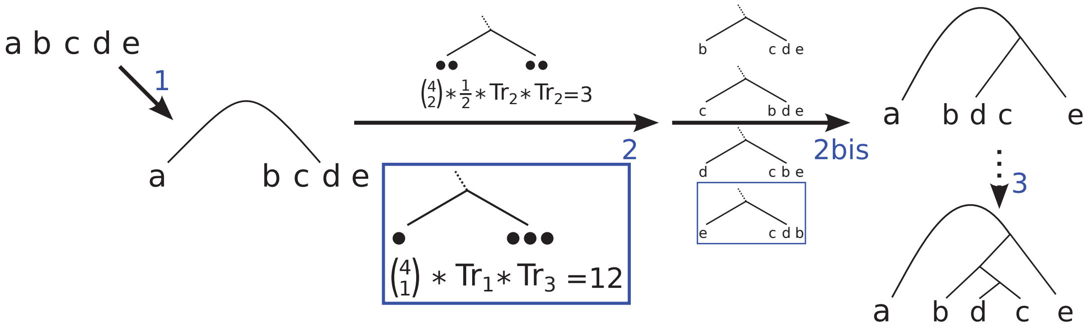

a b c d e
1
a
b c d e
$\binom{4}{2} * \frac{1}{2} * \mathrm{Tr}_2 * \mathrm{Tr}_2 = 3$
2
$\binom{4}{1} * \mathrm{Tr}_1 * \mathrm{Tr}_3 = 12$
b
c d e
c
b d e
d
c b e
e
c d b
2bis
a
b d c
e
3
a
b
d
c
e

Figure 3.12: Procedure to generate a random topology through clades splits. **1.** The root clade is split by separating a single leaf. **2.** A size smallest child clade, l, is chosen randomly according the possible number of topologies they can generate (the different possibilities are shown such that their sizes reflect their probabilities; the chosen split is framed in blue). **2bis.** A split of the given size (here, one) is generated by randomly selecting leaves. **3.** The process is repeated until the tree is fully bifurcating.

this method in the next section in order to extend the gene family's original CCP distribution (which is also based on clades and splits).

This method boils down to a procedure that splits a given clade in two children clades so that any possible topology is equiprobable, and which is illustrated in Figure 3.12.

Procedure to randomly split a clade

Consider a clade C, that is not the root clade, comprised of n leaves. Exactly Tr_n rooted topologies can be generated from C, where Tr_x the number of **rooted** topologies with x leaves[8].

The first step is to choose l, the size of the smallest child clade of C when it is split (the other child clade being of size $n - l$).

Each possible split whose smallest child is of size l can generate $Tr_l * Tr_{n-l}$ subtrees (that is, the number of rooted topologies formed by the first child clade multiplied by the number of rooted topologies formed by the second child clade).

It follows that given l, $Tr_l * Tr_{n-l} * L$ subtrees may be generated, where L corresponds to the number of splits where the smallest child is of size l and $L = \binom{n}{l}$, except in the special case where $l = \frac{n}{2}$, in which case $L = \binom{n}{l}/2$ because of

[8] As defined in the part about phylogenetic tree jargon.

symmetry[9].

Consequently, the probability to split C such that the smallest resulting clade is of size l is $Tr_l * Tr_{n-l} * L/Tr_n$. I use this probability to chose l.

Once l is chosen, it is easy to randomly select l leaves among the n of C to generate the desired split (see Figure 3.13.2bis).

With this procedure, each (rooted) subtree is given an equal chance to appear. By applying it recursively, a random (unrooted) topology is generated (the first split, which starts the recursion, is a split whose smallest child consists of only 1 as these splits occur in all topologies).

On the efficiency of random topologies

This method is able to explore fully the possible space of tree topologies for a given family, it is *a priori* not biased toward any of the source of information (as it does not take any of them into consideration).

However, and as a consequence, it can also be expected that the generated tree topologies do not fit very well with the alignment data, imply costly reconciliations and show little congruence with their evolutionary partners. In short, I expect that random topologies will often lead to a worse new score and shall therefore rarely be kept, thereby not decreasing the global score by much per round of optimization.

If it seems natural that directed moves (moves that takes into account some of the global score parts for instance) will be more efficient (*i.e.*, lead to a higher rate of score decrease per round), however it may not necessarily the case[10]. At worse these "blind" local moves may serve as a good baseline of comparison for different, directed, methods of new solution propositions.

3.3.2 Sampling according to gene sequences

A way to improve the likelihood that a proposed topology yields a good score (and thus is selected) is to add some information from the first from of co-evolution to the topology choosing process.

[9] $\binom{n}{k}$ is the number of subsets of size k among n, also described as the number of way to choose k elements among n.

[10] And many software rely on such undirected moves, such as PhyML [Guindon *et al.*, 2010] or TreeFix [Wu *et al.*, 2013].

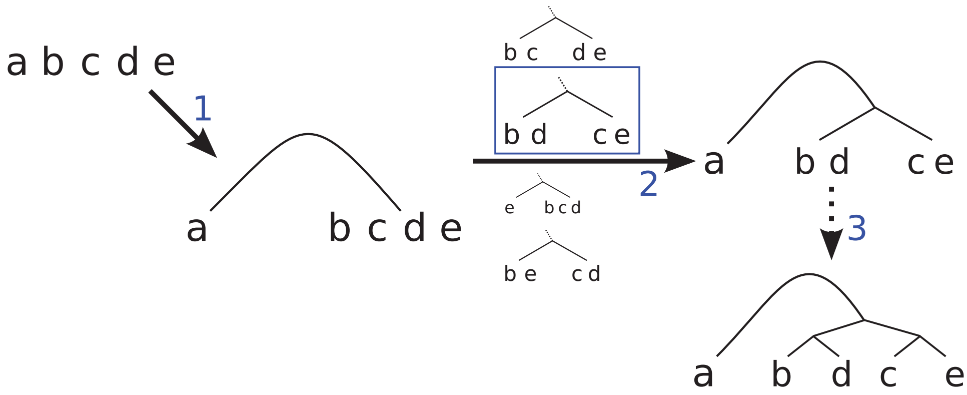


Figure 3.13: Procedure to sample in a CCP distribution. **1.** The root clade is split by separating a single leaf. **2.** A split of the clade is chosen randomly, according to its conditional clade probability (the different possibilities are shown such that their sizes reflect their probabilities; the chosen split is framed in blue). **3.** The process is repeated until the tree is fully bifurcating.

In my framework the first form of co-evolution is represented by the CCP distributions, so sampling a topology according to the first form of co-evolution comes down to sampling a topology in the CCP distribution of the concerned gene family. I give below the procedure to do so.

Sampling a topology according a CCP distribution

Consider a CCP distribution[11]. Remember that a clade C that is not composed of a single leave can be separated in two according to a split π and the *conditional probability* of this event is written $P_{CCP}(\pi|C)$. $P_{CCP}(\pi|C)$ is the ratio of the number of times we observe the split π in $\mathcal{G}$ divided by the number of time we observe the clade C in $\mathcal{G}$.

Starting from the clade that contains all the leaves of the families, I do a between one leaf and the clade containing all the other ones (see Figure 3.13.1), which I will now refer to as C [12].

Next, I consider all possible splits π for the clade C, and choose one randomly, with a probability $P_{CCP}(\pi|C)$ (see Figure 3.13.2). Each of the two clades generated trough the split π is recursively split (using the same process), until a fully

[11] cf. The topology chapter for a detailed description of CCP distributions

[12] Here, the choice of the leaf does not matter because in all trees, for each leaf there exists a bipartition that separates the leaf from all the other clades. So a tree sampled in the CCP distribution will have the split that we chose anyway, and the CCP distribution are build in such a way that the order in which we traverse them does not affect CCP-based computations.

bifurcating tree is generated (see Figure 3.13.3).

Such a simple procedure allows the generation of any possible amalgamable tree represented in the CCP distribution, with a probability proportional to its likelihood under this distribution.

Beyond the set of amalgamable tree topologies

The method mentioned above is able to generate a number of topologies, but usually not all of them. It is however possible that the *good topologies* (with respect to the global score) may not be represented in the CCP distribution, because of insufficient sampling and/or a lack of phylogenetic signal in the original alignment.

This phenomenon will tend to increase as the number of leaves grows (because the number of possible topologies grows exponentially with the number of leaves).

To overcome this limitation, I propose to apply the procedure described above, with the addition that any clade will have a probability α to be split according to the procedure for random topologies (as in Figure 3.12) instead of using the observed CCP distribution.

Adding this allows to randomly generate all possible topologies with a given number of leaves, but favouring the clades and splits observed in the original CCP distribution, thus trying to reap the advantages of each of the two aforementioned procedure (namely, reaching all possible topologies and biasing oneself towards topologies with a good score).

Discussion

By looking at many topologies, each with its own support, each implying its own reconciliation, adjacency histories and co-events, the space of solution can be explored, I propose here to sample topologies in a manner that is biased only by the sequence information.

Sampling tree topologies, unbiased by the criterion of reconciliation (contrary to TERA that is biased by both reconciliation and topology) allows for the exploration of scenarios that may not be parsimonious with respect to each individual criterion but possibly with respect to the global score, which is what I want.

The addition of a probability to chose random clades on top of the pre-existing CCP distribution one allows me to overcome its limitation: its reliance on a tree sample which may be incomplete for numerous reasons (lack of information in the sequence, lack of computational power to produce a representative sample). This addition offers a mean to combine sequence information while not limiting the exploration of the space of solution too much. Furthermore, it may remove some of the bias toward sequence information in the proposition of solutions if one so wishes, albeit I expect that this will also heighten the probability that a new proposition is rejected, thereby lengthening the time to convergence toward an optimum.

In any case, proposing new topologies chosen at random can miss good solutions. Indeed the reconciliations of the chosen topologies are still parsimonious (given the sampled topologies, becasuse they are obtained using TERA), while the global optimal certainly implies non-parsimonious reconciliations.

3.3.3 Sampling according to gene sequences and reconciliations

As the global score that I want to optimize has several components, taking more than one into account when proposing new solutions could lead to a faster global score improvement.

In the last section, I described how to sample solutions according to sequence information only (though the CCP distribution). In this section, I will describe an algorithm to sample solutions according to both sequence and reconciliation information (first and second form of co-evolution).

TERA [Scornavacca *et al.*, 2014] already presents an algorithm for joint optimisation of sequence and reconciliation based scores. However TERA computes most parsimonious scenarios. As I mentioned in the article about *Yersinia pestis* and earlier, solutions that minimize the global score are not expected to minimize the topology and reconciliation part of the score in the presence of co-evolution between genes.

To overcome the limitations brought around by parsimony, I applied to the TERA algorithm a transformation similar to the one described in Chauve *et al.* [2015] in the context of adjacency histories computation and that I familiarized myself with when I implemented it in DeCoSTAR. This transformation changes a dynamic programming algorithm (as is the case of DeCo, or TERA) for parsimonious scenario inference into one that is able to sample scenarios, with a relative probability inversely proportional to their cost (meaning that a given scenario will have a higher chance to be sampled than one with a higher cost). It does so by redefining costs as probabilities, mainly through an algebraic change from sums to products and from $min()$ operations to sums.

In this sense, the new algorithm I propose here is an intermediary between maximum parsimony and probabilistic models because it allows the exploration of a probabilised solution space that is centred around parsimony (as the parsimonious solutions have a higher probability to be sampled).

However, there already exists a software, ALE [Szöllősi *et al.*, 2013a], that considers jointly topologies and reconciliations in a probabilistic framework, and that could be ideal for this part. But because the global score framework is already a parsimony-based one, I think that it is pertinent to continue using my parsimony-derived sampling method[13].

Technically speaking, we consider reconciliations between a gene CCP distribution and a rooted binary ultrametric species tree, subdivided as described in the description of the algorithm in Doyon *et al.* [2010][14] and supplemented with a dead lineage.

Given a single reconciliation R, I do not consider its cost $\mathbb{C}$ (*i.e.*, the sum of the costs of each of its individual components), but rather its *Boltzmann factor*: $\mathcal{B}(R) = e^{-\frac{\mathbb{C}}{T}}$, where T is a temperature that determines how easy it is to sample non-parsimonious solutions.

For a given gene clade u and a given species x (*i.e.*, a node in the species tree), let $\mathcal{R}(u, x)$ be the set of all reconciliations of the subtree rooted at clade u and with the root in species x.

I define the $partition function$ of u and x as:

$$\mathcal{Z}(u, x) = \sum_{R \in \mathcal{R}(u,x)} \mathcal{B}(R)$$

That is, the sum of all the Boltzmann factors of the reconciliations in $\mathcal{R}(u, x)$. When the species is actually a node of the dead lineage, I use the symbol α instead

[13] Another, important, reason is that the algorithm that I describe here serves as the basis for another one that includes more parameters and that I describe in the next section. As I mentioned previously, an advantage of parsimony-based algorithm is that they are often simpler to design (and thus extend) than likelihood-based methods.

[14] Here, I only describe the algorithm for an ultrametric species tree. I have designed the recurrence formulas such that in the case where the species tree is not ultrametric, the main algorithm stays the same with the exception that all nodes are considered to have the same *time slice.* As with the maximum parsimony algorithm, the absence of an ultrametric species tree can lead to the generation of *time-inconsistent* scenarios.

of x.

A similar transformation is done for single event costs:

- the cost of a single duplication $\mathbb{D}$, $\delta = e^{-\frac{\mathbb{D}}{T}}$
- the cost of a single transfer $\mathbb{T}$, $\tau = e^{-\frac{\mathbb{T}}{T}}$
- the cost of a single loss $\mathbb{L}$, $\lambda = e^{-\frac{\mathbb{L}}{T}}$

Furthermore, I define $P_{CCP}(\pi|u)^{w_S}$ as the topology participation of split π of clade u to the partition function, with the normalised weight of sequence information: w_S (this definition is derived from the expression of the same contribution in [Scornavacca *et al.*, 2014]).

What follows are the formulas used to compute the partition function $\mathcal{Z}(u, x)$.

Partition function formulas

If the species is not a dead lineage of S:

$\mathcal{Z}(u, x) = \mathcal{Z}_{NoSoL}(u, x) + \mathcal{Z}_{speciationOutLoss}(u, x)$

Otherwise, when the species is in a dead lineage:

$$\mathcal{Z}(u, \alpha) = \sum_{\pi \in \Pi_u} \left(P_{CCP}(\pi|u)^{w_S} * \mathcal{Z}_{bifurcationOut}(\pi, \alpha) \right) + \mathcal{Z}_{Neutral}(u, \alpha) + \mathcal{Z}_{tansferBack}(u, \alpha)$$

Where Π_u is the set of all possible splits of u, $\mathcal{Z}_{event}(u, x)$ describes the part of the partition function that corresponds to u undergoing the (non-splitting) event *event* in species x and $\mathcal{Z}_{event}(\pi, x)$ describes the part of the partition function that corresponds to a split of u according to π in species x with the event *event*.

$\mathcal{Z}_{NoSoL}(u, x)$ represents the solutions for the reconciliation of u and x, excluding events of speciation to an dead/unsampled lineage and loss in the current lineage and is written as:

$$\mathcal{Z}_{NoSoL}(u, x) = \sum_{\pi \in \Pi_u} \left(P_{CCP}(\pi|u)^{w_S} * \left(\mathcal{Z}_{speciation}(\pi, x) + \mathcal{Z}_{duplication}(\pi, x) + \mathcal{Z}_{speciationOut}(\pi, x) \right) \right) + \mathcal{Z}_{neutral}(u, x) + \mathcal{Z}_{speciationLoss}(u, x) + \mathcal{Z}_{leaf}(u, x)$$

The formulas for the different terms of these equations are shown in Table 3.1, where x_1 and x_2 are the children of x (when applicable); $\pi[1]$ and $\pi[2]$ are the clades

	split of the gene tree	no split of the gene tree
duplication and loss	$\mathcal{Z}_{speciation}(\pi, x) =$ $\big(\mathcal{Z}(\pi[1], x_1) * \mathcal{Z}(\pi[2], x_2)\big)+$ $\big(\mathcal{Z}(\pi[2], x_1) * \mathcal{Z}(\pi[1], x_2)\big)$ $\mathcal{Z}_{duplication}(\pi, x) =$ $\big(\mathcal{Z}(\pi[1], x) * \mathcal{Z}(\pi[2], x) * \delta\big)$	$\mathcal{Z}_{neutral}(u, x) =$ $\mathcal{Z}(u, x_1)$ $\mathcal{Z}_{speciationLoss}(u, x) =$ $\big(\mathcal{Z}(u, x_1) * \lambda\big)+$ $\big(\mathcal{Z}(u, x_2) * \lambda\big)$ $\mathcal{Z}_{leaf}(u, x) =$ 1 , if the leaf of u maps to x 0, otherwise
transfer	$\mathcal{Z}_{speciationOut}(\pi, x) =$ $\big(\mathcal{Z}(\pi[1], x) * \mathcal{Z}(\pi[2], \alpha_x)\big)+$ $\big(\mathcal{Z}(\pi[2], x) * \mathcal{Z}(\pi[1], \alpha_x)\big)$	$\mathcal{Z}_{speciationOutLoss}(u, x) =$ $\big(\mathcal{Z}(u, \alpha_x) * \lambda - \mathcal{Z}_{NoSoL}(u, x) * \tau\big)$
dead lineage	$\mathcal{Z}_{bifurcationOut}(\pi, \alpha) =$ $\Big(\mathcal{Z}(\pi[1], \alpha) * \mathcal{Z}(\pi[2], \alpha)\Big)$	$\mathcal{Z}_{tansferBack}(u, \alpha) =$ $\sum_{x \in S(ts_\alpha)} \Big(\mathcal{Z}_{NoSoL}(u, x) * \tau\Big)$

Table 3.1: Description of the formulas to reconcile a gene clade u, or a gene split π, with a species x or a dead lineage α and with different events.

formed by the split π; α_x is the node of the dead lineage with the same time slice as x and $S(ts_\alpha)$ is the set of species with the same time slice as α.

Events of speciation and speciation and loss ($\mathcal{Z}_{speciation}(\pi, x)$ and $\mathcal{Z}_{speciationLoss}(u, x)$) can only be computed when the species x has two children (meaning that it is not a leaf or an artificial node). Conversely, simple vertical transmission (without any further event) ($\mathcal{Z}_{neutral}(u, x)$) is computed if x is an artificial node. Also, leaf events ($\mathcal{Z}_{leaf}(u, x)$) are only computed if u is a *leaf clade* and x is a leaf.

In the case of the speciation toward a dead lineage and loss in the current lineage ($\mathcal{Z}_{speciationOutLoss}(u, x)$) the first term of the formula describes the speciation and loss toward the dead lineage, the second, negative term, is an adjustment that avoids a direct transfer back of clade u in species x.

Note that if I kept the original form it takes in TERA, the equation for $\mathcal{Z}_{tansferBack}(u, \alpha)$ would both include and be included in itself, thus creating a circularity that invalidates the Boltzmann-Gibbs sampling scheme. This problem is not present in a maximum parsimony framework as such potentially circular cases would have always implied additional transfers and losses than their non circular counterpart, thus ensuring they would never end up in the solutions. However in my context all the scenarios that I specify through the recurrence equations can actually be sampled, hence the need to modify the equation so that it still means the same thing (computation of the cost of having this clade undergo a transfer reception event), but without circularity.

Algorithmic complexity issues related to transfer inference

In TERA, computing the cost of a transfer is done by identifying, at each clade and each time slice, the *best receiving species*: the species with minimal cost. Then, given a clade and a time slice, all parsimonious transfers are toward this *best receiving species* (except when the transfer originates from this species; in this case, the transfer is toward the second best receiving species). This strategy allows the algorithm to find the most parsimonious reconciliation in time and space complexity of $O(|S'| \cdot |G|)$ (where $|G|$ is the size of the gene CCP distribution and $|S'|$ is the size of the subdivided species tree).

However in the case of the Boltzmann-Gibbs sampling of reconciliations, all (parsimonious and non-parsimonious) solutions have to be listed, which means that all possible transfer recipients have to be enumerated. A naive translation of the rules of TERA in a Boltzmann-Gibbs sampling scheme would thus yield a $O(|S'|^2 \cdot |G|)$ complexity algorithm, both in time and space.

I solve this increase in complexity, by forcing all the transfers to go through a dead lineage. This change, when applied as described in the formulas above, does not modify the space of solutions. It can be seen in the formulas for $\mathcal{Z}_{speciationOut}(\pi, x)$ and $\mathcal{Z}_{speciationOutLoss}(u, x)$, which only make reference to $Z(u, \alpha)$, but more importantly in the formula for $\mathcal{Z}_{tansferBack}(u, \alpha)$, which iterates over all species in a given time slice. As this iteration is done only once per time slice, the complexity is $O(2|S'| \cdot |G|) \sim O(|S'| \cdot |G|)$, as in the original TERA algorithm.

Sampling reconciled gene trees

Once the partition function has been recursively computed for each couple of clade and species, it can be stochastically backtracked to sample reconciled gene trees.

In the event where the species tree was subdivided, then all the produced reconciled gene tree are valid. However if the species tree was not subdivided, then it is possible to sample time-inconsistent solutions when transfers are allowed. Such time-inconsistent solutions are detected during the backtracking procedure and discarded. This simple procedure manages to produce the expected distribution of time-consistent reconciled gene trees, but it also creates an additional time overhead when sampling as there may be several backtracking attempt to produce a single valid reconciled tree[15]. For practicality purposes, the user is free to choose (through options in the implemented software) if time-inconsistent scenarios should be discarded or not, and can even specify the maximum number of attempts authorized to get the desired number of time-consistent reconciliations (above this number time-inconsistent reconciliations will be kept, so that the output will contain the desired number of reconciliations) in order to limit the computation overhead.

The described algorithm has been implemented and tested as an object oriented python program.

3.3.4 Sampling according to adjacencies and co-events

In the previous sections I described ways to approximate the sampling of a gene family topology and reconciliation according to the global score. Instead of sampling according to the four parts of the global score at once, I only consider a fraction of them. Namely I sample according to the sequence information (only the *topology* part of the score) and I sample according to the sequence and reconciliation information (*topology* and *reconciliation* parts of the score).

I will examine in this section the possibility to sample according to yet another component of the score.

Rather that sampling according to the *topology*, *reconciliation*, *adjacency* and $co-event$ parts together (a problem that I consider intractable), I will consider a score that I note:

$$topology + reconciliation + co - event'$$

[15] Note that in the case of most parsimonious reconciliation, all optimal reconciliations may be time inconsistent. However here, as we are not limited to most parsimonious solutions, it is always possible (albeit potentially improbable) to sample a time-consistent scenario. A simple proof of this is that the space of solutions includes reconciliations without any transfers (such a reconciliation is always time-consistent).

where *topology* and *reconciliation* correspond to their homonymous part of the global score, but defined at the scale of a single gene family and $co-event'$ corresponds to an approximation of the $co-event$ part of the global score (again, at the scale of this gene family).

The computation of the $co-event$ part of the global score is dependent on adjacencies and spans multiple gene families (*i.e.*, more than two). In contrast, $co-event'$ corresponds to associations between reconciliation events (*i.e.*, duplication, transfer and loss events) of two gene families, independent from the adjacency histories. I respectively call these two gene families, the *target* gene family and the *guide* gene family (the target is the one we are doing inference on, and the guide is fixed).

An association may form between an event of the target gene family (target event) and an event of the guide gene family (guide event) if:

- they are in the same species (and the same time slice if this applies).
- they describe the same type of event (duplication, loss or transfer reception).
- the guide event has not already formed an association with a descendant or an ancestor of the target event.
- the target event has not already formed an association with another guide event.

These conditions are also illustrated for duplications in Figure 3.14. They mimics the way in which co-events are formed in the global score, which the associations are here to approximate.

In the score that I define, an association costs:

- $\mathbb{A}_{dup}$, if the association occurs between duplications.
- $\mathbb{A}_{tr}$, if the association occurs between transfers.
- $\mathbb{A}_{loss}$, if the association occurs between losses.

Note that to favour the apparition of associations, these costs will usually be negative (similar to co-events that are scored negatively in the global score).

$co-event'$ is the sum of the costs of all the associations between the target and the guide gene families. However in practice the conditions I defined imply that each association corresponds to exactly one event of the target family, so that I may

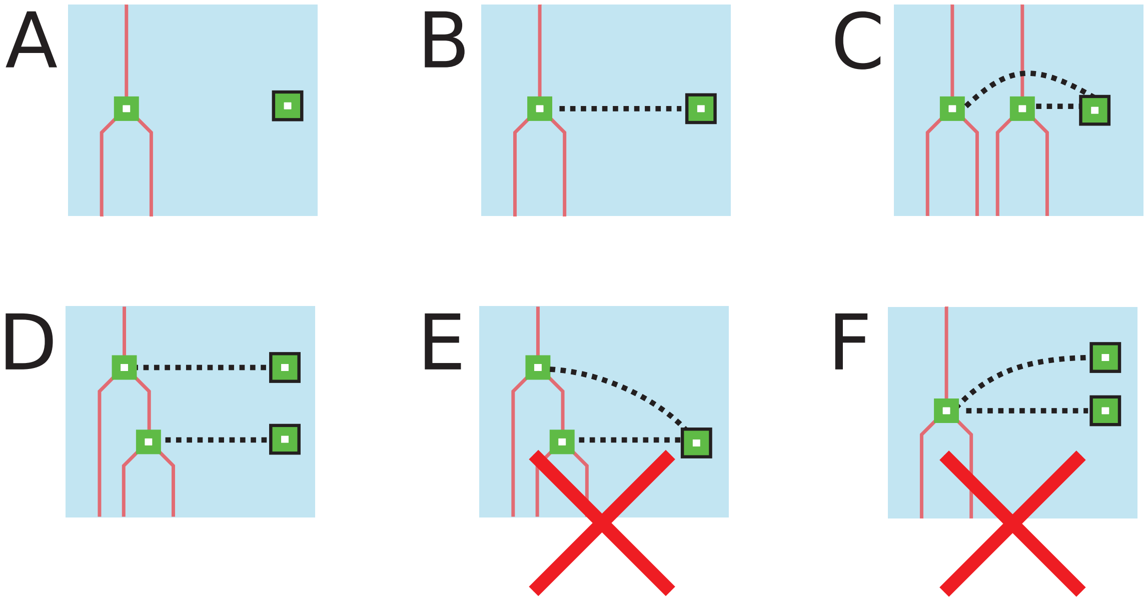


Figure 3.14: Illustration of the formation of associations between target duplications (green bordered square) and guide duplications (black bordered green bordered square). **A.** case of a simple duplication (without association). **B.** case of an association between the target and guide duplications. **C.** case of a guide duplication forming associations with two target duplications. **D.** case of two associations following each other (as one target event is the parent of the other). **E.** forbidden case, because a guide event cannot form an association with a target duplication and its ancestor at the same time. **F.** forbidden case, because a target event cannot form an association with more than one guide event.

consider them together in the score computation. I note the score of such an event and its association:

- $c_{Adup} = c_{dup} + \mathbb{A}_{dup}$, if the event is a duplication.
- $c_{Atr} = c_{tr} + \mathbb{A}_{tr}$, if the event is a transfer.
- $c_{Aloss} = c_{loss} + \mathbb{A}_{loss}$, if the event is a loss.

In these formulas, c_{dup} (resp. c_{loss}, c_{tr}) is the cost of a single duplication (resp. loss, transfer) event.

Given a solution (that I note R), corresponding to reconciled tree for the target gene family and with associations between guide and target reconciliation events, its cost (that I note $\mathbb{C}$) according the score that I define here can be written:

$$\mathbb{C} = topology + reconciliation + co - event' =$$

$$w_S \times -log\left(\frac{P_{CCP}(T)}{P_{CCP}(T_{max})}\right)+$$

$$\left(c_{dup}.n_{NAdup} + c_{loss}.n_{NAloss} + c_{tr}.n_{NAtr}\right)+$$

$$\left(c_{Adup}.n_{Adup} + c_{Aloss}.n_{Aloss} + c_{Atr}.n_{Atr}\right)+$$

, where w_S, $P_{CCP}(T)$ and $P_{CCP}(T_{max})$ were already defined in the previous section and in the global score, n_{NAdup} (resp. n_{NAloss}, n_{NAtr}) is the number of single duplication (resp. loss, transfer) events that are not associated to a event of the guide gene family and n_{Adup} (resp. n_{Aloss}, n_{Atr}) is the number of single duplication (resp. loss, transfer) events that are associated to a event of the guide gene family.

The algorithm that I describe in this section can sample topologies and reconciliations (and associations) for the target gene family according to this score. More formally, it takes as input a CCP distribution, a species tree, guide evolutionary events along the species tree (*i.e.*, events of duplication, losses or transfer receptions in the guide family reconciliation), costs for the different events, the topology weight and a temperature and it outputs a sample of topologies and reconciliations[16].

This method builds on the one presented in the previous section and, as I did there, I consider the boltzmann factor corresponding to the costs of solutions: $\mathcal{B}(R) = e^{-\frac{\mathbb{C}}{T}}$ (remember that T is a temperature that determines how easy it is to sample non-parsimonious solutions).

I then define the $partitionfunction$ of u and x as:

$$\mathcal{Z}(u,x) = \sum_{R \in \mathcal{R}(u,x)} \mathcal{B}(R)$$

,where u is a gene clade, x is a species tree node and $\mathcal{R}(u,x)$ is the set of all reconciliations of the subtree rooted at clade u and with its root in species x. Computing the partition function of the root clade of the CCP distribution over all species will let me sample reconciliations according to their Boltzmann factor.

In the framework of the Boltzmann-Gibbs sampling scheme the costs for a target

[16] In this sample, the probability to be sampled is inversely proportional to the $topology + reconciliation + co - event'$ cost of the solution.

event and an association become:

- $A\delta = e^{-\frac{c_{Adup}}{T}}$
- $A\tau = e^{-\frac{c_{Atr}}{T}}$
- $A\lambda = e^{-\frac{c_{Aloss}}{T}}$

Additional definitions: chained associations

In order to be able to compute the partition function while complying with the rules for the formation of associations with guide events, I have to introduce additional notations.

Consider, for any species x, x_D , x_L and x_T, respectively the number of duplication, loss and transfer reception guide events associated with x.

For any reconciliation of the clade u with species x, I call d the number of *chained duplication associations* occurring in this reconciliation. This number corresponds to the minimum number of guide events needed to explain the duplication associations between u or descendants of u and guide events in the species x. The Figure 3.15 illustrates the evolution of the value of d along a reconciled gene tree. On the left species branch, there are two associations, but as they occur in two sister lineages they can correspond to only one guide duplication (they could also correspond to two guide duplications, but I am interested in the minimal value here), hence the value of 1 for the d of their lowest common ancestor. On the right species branch the two duplication associations imply a node and its child, so that their associations cannot occur with the same guide event: this history needs at least 2 guide duplications to work. Also, note that in the common ancestor of the two species branches, d is again equal to 0 as it is a species specific measurement (changing species reinitializes it).

Following this, I define $\mathcal{Z}(u, x, d)$, the partition function of the reconciliations of u and x with d chained duplication associations.

Note that d is a integer ranging between 0 and x_D.

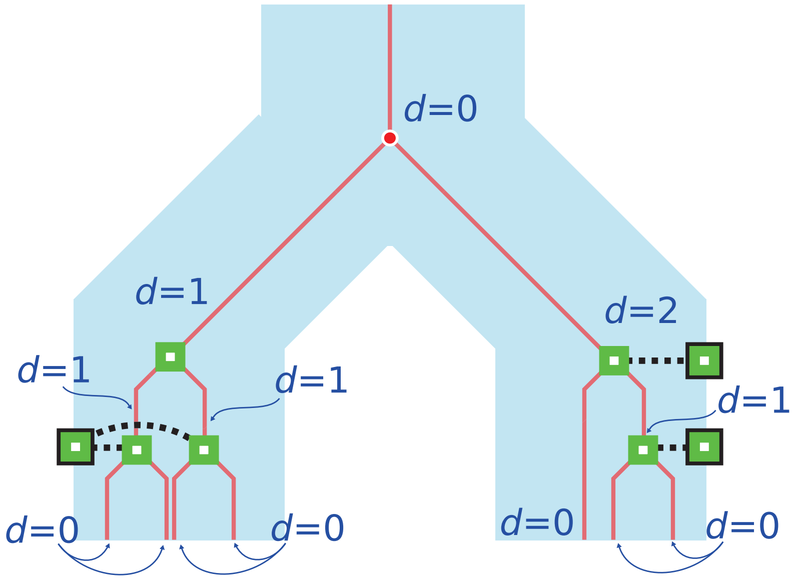


Figure 3.15: Reconciled target gene tree with some event associations. The value d (number of chained duplication associations) at each node is marked in blue.

Reconciliation partition function formulas in the presence of event associations

In general, I define:

$$\mathcal{Z}(u,x) = \sum_{0 \leq d < x_D} \Big(\mathcal{Z}(u,x,d) \Big)$$

If the species x is not a dead lineage,

$$\mathcal{Z}(u,x,0) = \mathcal{Z}_{d-dependant}(u,x,0) + \mathcal{Z}_{d-independant}(u,x) + \mathcal{Z}_{speciationOutLoss}(u,x)$$

and,

$$\mathcal{Z}(u,x,d) = \mathcal{Z}_{d-dependant}(u,x,d)$$

$\mathcal{Z}_{d-dependant}(u,x,d)$ is the part of the score of $\mathcal{Z}(u,x,d)$ that depends on the value of d and is defined as:

$$\mathcal{Z}_{d-dependant}(u,x,d) = \sum_{\pi \in \Pi_u} \Big(P_{CCP}(\pi|u)^{w_S} * \big(\mathcal{Z}_{duplication}(\pi,x,d) + \mathcal{Z}_{speciationOut}(\pi,x,d) \big) \Big) + \mathcal{Z}_{neutral}(u,x,d)$$

$\mathcal{Z}_{d-independant}(u,x,d)$ is the part of the score of $\mathcal{Z}(u,x)$ that does not depend on the value of d. It is only defined for $\mathcal{Z}(u,x,0)$ and corresponds to:

$$\mathcal{Z}_{d-independant}(u,x) =$$

	split of the gene tree	no split of the gene tree
duplication and loss	$\mathcal{Z}_{speciation}(\pi, x) =$ $\big(\mathcal{Z}(\pi[1], x_1) * \mathcal{Z}(\pi[2], x_2)\big)+$ $\big(\mathcal{Z}(\pi[2], x_1) * \mathcal{Z}(\pi[1], x_2)\big)$ $\mathcal{Z}_{duplication}(\pi, x, d) =$ $\Big(\mathcal{Z}(\pi[1], x, d) * \sum_{0<i\leq d} \big(\mathcal{Z}(\pi[2], x, i)\big) * \delta\Big)+$ $\Big(\mathcal{Z}(\pi[2], x, d) * \sum_{0<i<d} \big(\mathcal{Z}(\pi[1], x, i)\big) * \delta\Big)+$ $\Big(\mathcal{Z}(\pi[1], x, d-1) * \sum_{0<i\leq d-1} \big(\mathcal{Z}(\pi[2], x, i)\big) * A\delta\Big)+$ $\Big(\mathcal{Z}(\pi[2], x, d-1) * \sum_{0<i<d-1} \big(\mathcal{Z}(\pi[1], x, i)\big) * A\delta\Big)$	$\mathcal{Z}_{neutral}(u, x, d) =$ $\mathcal{Z}(u, x_1, d)$ $\mathcal{Z}_{speciationLoss}(u, x) =$ $\big(\mathcal{Z}(u, x_1) * \lambda\big)+$ $\big(\mathcal{Z}(u, x_2) * \lambda\big)+$ $\big(\mathcal{Z}(u, x_1) * A\lambda\big)+$, if $x_{2L} > 0$ $\big(\mathcal{Z}(u, x_2) * A\lambda\big)$, if $x_{1L} > 0$ $\mathcal{Z}_{leaf}(u, x) =$ 1 , if the leaf of u maps to x 0, otherwise
transfer	$\mathcal{Z}_{speciationOut}(\pi, x, d) =$ $\big(\mathcal{Z}(\pi[1], x, d) * \mathcal{Z}(\pi[2], \alpha_x)\big)+$ $\big(\mathcal{Z}(\pi[2], x, d) * \mathcal{Z}(\pi[1], \alpha_x)\big)$	$\mathcal{Z}_{speciationOutLoss}(u, x) =$ $\big(\mathcal{Z}(u, \alpha_x) * \lambda\big)+$ $\big(\mathcal{Z}(u, \alpha_x) * A\lambda\big)$, if $x_L > 0$
dead lineage	$\mathcal{Z}_{bifurcationOut}(\pi, \alpha) =$ $\Big(\mathcal{Z}(\pi[1], \alpha) * \mathcal{Z}(\pi[2], \alpha)\Big)$	$\mathcal{Z}_{tansferBack}(u, \alpha) =$ $\sum_{x \in S(ts_\alpha)} \Big($ $\mathcal{Z}_{NoSoL}(u, x) * \tau +$ $\mathcal{Z}_{NoSoL}(u, x) * A\tau$, if $x_T > 0$ $\Big)$

Table 3.2: Description of the formulas to reconcile a gene clade u, or a gene split π, with a species x or a dead lineage α and with different events and different number of chained duplication associations d. The formulas in gray are the formulas that have not changed from Table 3.1.

$$\sum_{\pi \in \Pi_u} \Big(P_{CCP}(\pi|u)^{w_S} * \big(\mathcal{Z}_{speciation}(\pi, x)\big)\Big)$$
$$+\mathcal{Z}_{speciationLoss}(u, x) + \mathcal{Z}_{leaf}(u, x)$$

Table 3.2 details the different components of these formulas.

The equation for $\mathcal{Z}_{duplication}(\pi, x, d)$ has 4 components. The first two components correspond to a simple duplication while the last two components correspond to a duplication that form an association with a guide event. When a simple duplication occurs, this means that d chained duplication associations must have happened in at least one child clade of u. When an association occurs, $d-1$ chained duplication associations must have happened in at least one child clade of u (as we add one). The first and third components of the equation represent the case where $\pi[1]$

has d, respectively $d-1$, chained duplication associations. The second and fourth components correspond to the same case, but with $\pi[2]$.

For the events of speciation, speciation and loss (*speciationLoss*), speciation toward an unsampled lineage and loss (*speciationOutLoss*), and leaf event, there is either no children, or these are not in the same species, thus breaking the chain of duplication associations. Hence these events are independent from d.

Where a gene loss occurs as part of an event, it may be part of a loss association. This is reflected in the cost formulas. In the equation for $\mathcal{Z}_{speciationLoss}(u,x)$, the two last components correspond to the case of a loss association in one of the children species. In the equation for $\mathcal{Z}_{speciationOutLoss}(u,x)$, the second case corresponds to a loss association in x.

If the species is a dead lineage, the formulas stays the same as in the previous section, with the exception of the one for the transfer back event. A transfer from a dead lineage back to the species tree can be part of a transfer reception association.

Sampling reconciled gene trees in the presence of event associations

Sampling in the presence of transfers can lead to a number of invalid solutions that one has to exclude from the results.

First, as before, when the species tree is not ultrametric, it is possible to sample time-inconsistent reconciliations that must be excluded from the sample (as mentioned in the previous section).

Additionally, transfers make it possible to create solutions where more duplication associations than possible may appear. This is due to the formulas for transfer-related events ($\mathcal{Z}_{speciationOutLoss}(u,x)$,$\mathcal{Z}_{speciationOut}(u,x)$ and $\mathcal{Z}_{tansferBack}(u,\alpha)$) which, for tractability purposes, do not carry up values of d, the number of chained duplication associations in a species. Thus, if a transferred descendant somehow comes back to the origination species and then does a duplication association, it will not be counted (or rather, it will be counted independently).

I call such a reconciliation where the number of associations exceeds the number of possible ones (*i.e.*, the number of of guide events in a given species) *association inconsistent* reconciliations (see Figure 3.16 for an example) and I exclude them from the sampled solutions as well[17].

[17] This is, again, an option that may be toggled off by the user.

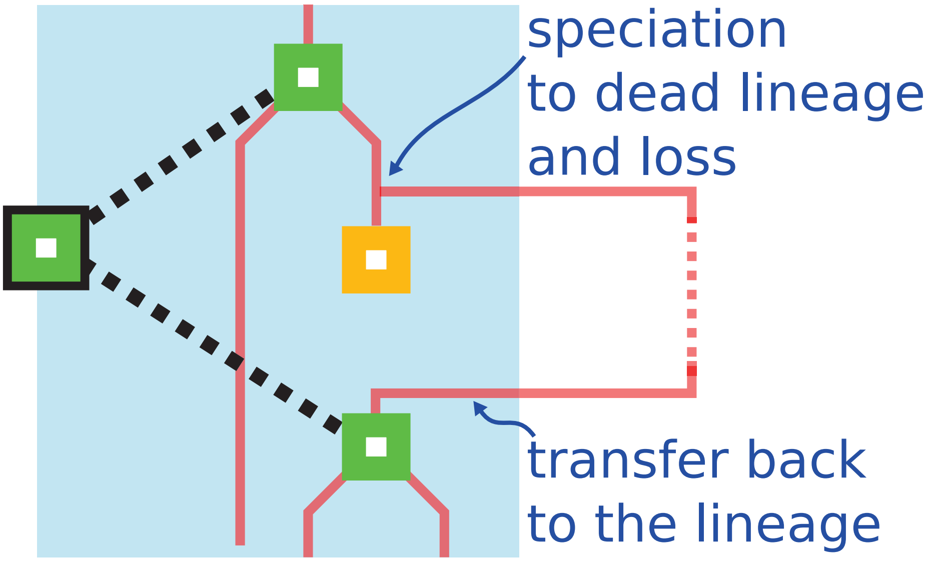


Figure 3.16: Illustration of *association inconsistency* where a guide duplication forms an association with a target duplication and one of its ancestor. This scenario is possible because of the transfer separating the two target duplications.

Choosing a guide tree

As the goal here is to bias the reconciliation of the target gene tree toward the guide gene tree, the choice of this guide tree is fundamental. The guide tree is used to form co-events with (or rather, associations as a proxy for co-events here), and co-events are detected using adjacency histories. So for a guide tree to fill its purpose it should be from a gene family which shares many adjacencies with the target gene family. To respect this idea, I simply propose to randomly choose a guide family among the families with whom the target family shares adjacencies, and with a probability proportional to the number of extant adjacencies they share.

Once the guide gene family is chosen, I extract *guide events*, from its reconciliation as shown in Figure 3.17. A guide event can either be a *duplication*, a *loss* or a *transfer reception* and are also associated to a species in the species tree.

Discussion

The defined partition function formulas allow the sampling of reconciled gene trees according to their likelihood in a CCP distribution, events of duplication, loss, transfers and event associations they form with a guide reconciliation, which serve as an approximation of co-events.

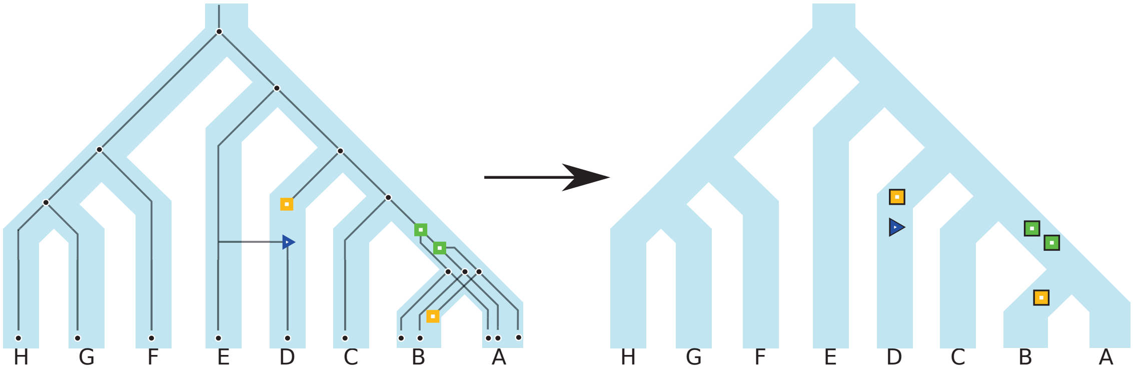


Figure 3.17: From a guide reconciled gene tree (left), I extract guide events (right). I represent guide events (duplications as green square, losses as an orange square, transfer reception as a blue triangle) with a black border.

In this Boltzmann-Gibbs sampling scheme the partition function must be computed for any possible u, x and also d when guide duplications are present. Consequently, the complexity of the proposed algorithm is $O(|S'| * |G| * d_{max})$, where d_{max} is the maximum number of guide duplications in a single species (*i.e.*, the maximum possible value for d)[18].

As I ultimately describe here a method with the goal to propose topologies and reconciliations for the optimization of my global score, I am particularly interested in the ability of Boltzmann-Gibbs sampling schemes to escape parsimonious solutions. However, much like one can transform parsimony formulas into a Boltzmann-Gibbs sampling scheme, one could quite easily transform the partition functions I defined into costs formulas by changing sums into $min()$ operators, products into sums, and using the original costs (rather than their exponential).

It is worth noting that the presented algorithm constitutes, independently from the global score framework, an interesting contribution to the study of reconciliation reconstruction in a context that is aware of genes evolutionary interactions.

Like the the algorithm that it is based on, the method described here was implemented in the form of an object oriented python program.

[18] In practice I do not expect d_{max} to be very big

3.4 Results

Having described a global score and methods to optimize it. I want to apply them to biological data-sets in order to verify that the optimization in indeed working (meaning that the score diminishes as rounds go), study how the gene trees change as the global score decreases and also gather estimates of biological measures. In particular I am interested in the sizes of large reconciliations events because, as I mentioned before, methods that consider gene families independently will tend to underestimate these (when they actually can tell something about it).

3.4.1 Data-sets

I applied the methods I developed on three data-sets. The first two are subsets of gene families evolving among a set of 36 mammals and were obtained from the Ensembl Compara database (release 87 of December 2016 ; [Yates *et al.*, 2016]). The third data-set corresponds to the fungi data-set used in Szöllősi *et al.* [2015] and is composed of gene families from the HOGENOM database (release 06 ; [Penel *et al.*, 2009]) evolving among 28 fungi species. This third data-set, unlike the two mammalian ones, includes horizontal gene transfers.

From the complete set of gene families in the databases, I applied two filters on the gene families. The first filter select families that have a gene copy in at least half of the species (13 species for the mammalian data-sets, 14 for the fungal one). This filter aims at removing the gene families that are only present in a few clades and whose history would either imply many gene losses or only span a fraction of the species tree. The second filter keep only the families which have at most as many genes as twice the number of species (72 for mammals, 56 for fungi). This second filter is here to keep only families whose evolution is not too complex and for whom the hypothesis of a parsimonious (or quasi-parsimonious) reconciliation is reasonable (as families with lots of genes are supposed to have a higher rate of duplication and transfer and thus a more complex history).

For all three data-sets the adjacencies used correspond to neighbourhood along a chromosome.

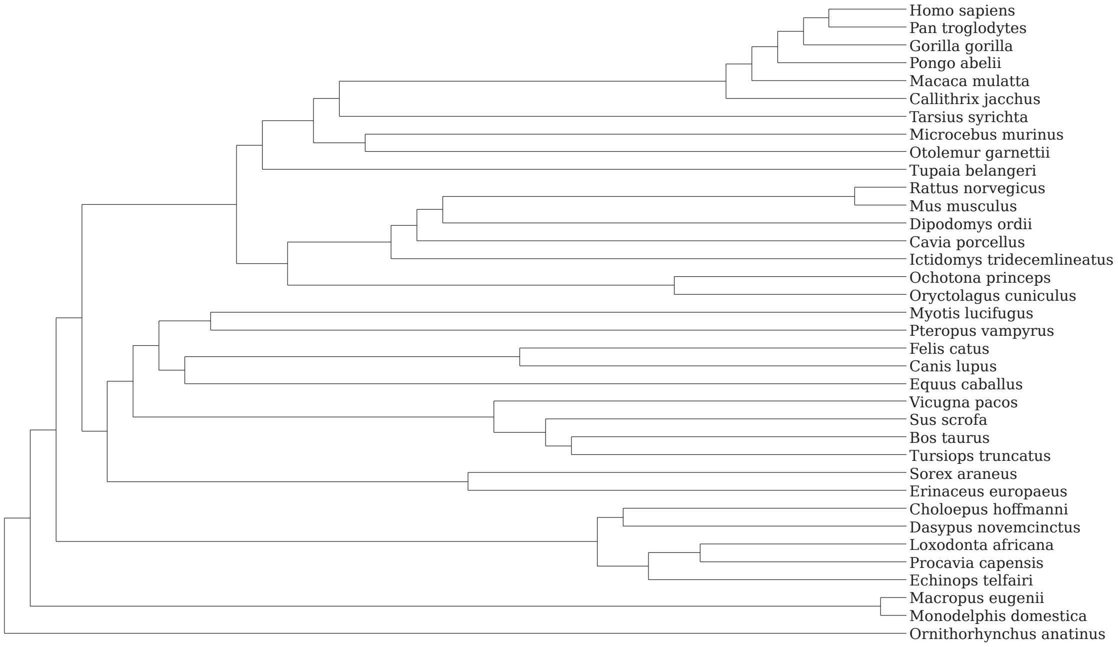


Figure 3.18: The species tree used in the mammalian data-sets. Note that tree branch lengths are not informative here.

The mammalian data-sets

As mentioned before, these data-sets consist of Ensembl protein families (Ensembl release 87 of December 2016 ; [Yates *et al.*, 2016]). Out of 6 695 families, 5 180 satisfy both filters and out of these I generated two data-sets containing respectively 500 and 1 000 randomly chosen gene families.

For each selected gene families, I use a sample of 10 000 bootstrap trees (obtained using IQtree [Nguyen *et al.*, 2015]) as the basis for the construction of their CCP distributions.

The species tree used correspond to the Ensembl mammalian species tree, as shown in Figure 3.18.

For each of these two data-sets, adjacencies where extracted from the position of genes along chromosomes (so adjacencies correspond to a neighbourhood relationships). The mammalian genomes included in this data-set correspond to the ones present in Ensembl and are not necessarily well assembled, which has consequences in terms of adjacencies. Indeed because of it many genes are isolated on their contigs and the number of adjacencies per genome ranges from 3 to 529 in the data set with 500 gene families, and from 14 to 1 079 for the one with 1 000 gene families (while genome show only little variation in terms of number of genes they contain). In consequence, during the upcoming analyses I will activate the scaffolding option of DeCoSTAR, which infers extant adjacencies and was developed for such a case (see Anselmetti *et al.* [2015] for details).

The fungal data-sets

Out of the 11 295 gene families used in the fungi data-set of Szöllősi *et al.* [2015], 1 974 remain after I applied both filters.

Out of these 1 974, I removed the 72 families which possessed more than 1 300 clades because the proof-of-concept implementation of the method I use to sample new solutions according to the first two or all three forms of co-evolution (see the section about sampling according the the CCP distribution and reconciliation and the following section) could not handle them in less than 10Gb of memory [19].

The CCP distribution of each selected gene families is constructed using an a posteriori Bayesian sample of 10 000 trees, as described in [Szöllősi *et al.*, 2015].

The species tree also corresponds to the Fungi tree A generated in [Szöllősi *et al.*, 2015] and that is shown in Figure 3.19.

Adjacencies for this data-set were obtained using the genes start position along chromosomes, contained in the HOGENOM database.

3.4.2 Getting the initial solution

Unless specified otherwise, my initial solution (the one onto whom I start doing rounds of optimization), uses topologies and reconciliations obtained using the TERA algorithm (with costs corresponding to the one I use in the global score) onto which I compute adjacency histories using DeCoSTAR and then detect co-events.

Using TERA has the consequence that the initial solution jointly maximizes the topology and reconciliation parts of the score without any concern toward the adjacency and co-event parts (as this is what TERA is designed to do). This means that any better solution will, at best, exhibit a sum of the topology and reconciliation part equal to that of the initial solution, so that the source of the score improvement (*i.e.*, diminution) are the adjacency and co-event parts of the global score.

[19] Note that 1 300 clades correspond to a large quantity of phylogenetic information as it is more than the number of clades among all the topologies with 10 leaves (1 023 clades).

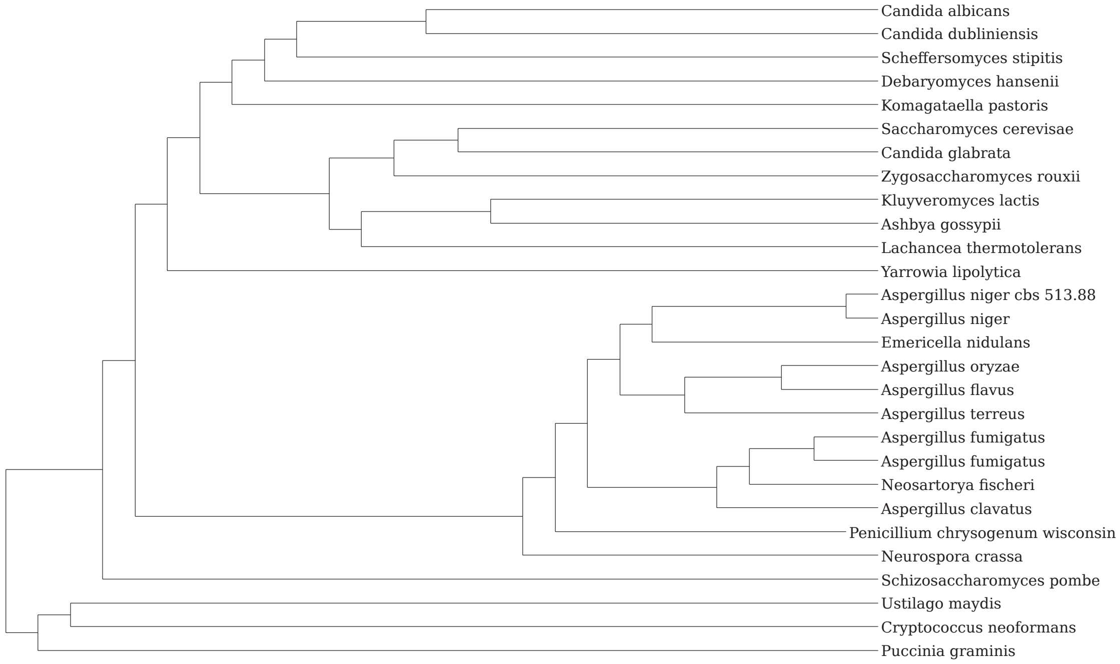


Figure 3.19: The species tree used in the fungal data-set. Note that tree branch lengths are not informative here.

3.4.3 Testing different parameters to optimize the global score

These analyses were performed on the data-set containing 500 mammalian gene families. As it is the smallest data-set, a round of optimization on it takes less computation time, which allowed me to test the global optimization procedure under many set-ups in an efficient manner.

I consider set-ups differing by the method used to propose new solution (see the previous section): uniform sampling of random topologies, sampling using the CCP distribution, sampling using the CCP distribution and reconciliation, and sampling using the CCP distribution, reconciliation and co-events (or rather, associations) with a guide family. I respectively refer to them as *Uniform*, *CCP*, *CCPRec* and *CCPRecCoev*.

I also considered cases where I examine 1 proposal before testing another gene family and another where I examine 10. I respectively call these conditions *1 sample* and *10 samples*.

For the last two proposal methods (sampling using the CCP distribution and reconciliation, and sampling using the CCP distribution, reconciliation and co-events with a guide family), a temperature parameter determine the expected deviation from maximum parsimony of sampled solutions. For these I tested two conditions : a *cold* one (temperature of 0.5) and a *hot* one (temperature of 2)[20].

Thus, when I refer to a set-up as *CCPRec 1sample cold*, it corresponds to sampling using the CCP distribution and reconciliation, with 1 proposal per gene family and a temperature of 0.5 for the proposals.

I consider a unique starting point (the solution obtained from TERA followed by DeCoSTAR) and make each set-up perform as many successive rounds of optimization as possible in 4 hours. The runs were performed on a desktop computer (I impose a time constraint here, because the different methods do not have the same complexity and thus a round duration varies with the method employed). The chosen temperature for the acceptance of new solutions is 0 (only solutions which actually decrease the score are chosen). After these, I compare the different set-ups in terms of the amount of optimization they could yield during this fixed time (the better set-ups result in lower global scores).

Figures 3.20 and 3.21 show the evolution the the global score during rounds of optimization using different methods for the proposition of new solutions. The curves corresponding to the *Uniform* protocol are not showed because they were unable to produce any solution that was accepted (*i.e.*, the global score remained the same).

Figure 3.20 shows that proposing new solutions that consider CCP distributions and reconciliation jointly (and also when adding co-events) with a low temperature lead to a lower score decrease than considering CCP distributions alone. However, with a higher temperature it leads to substantial global score improvements.

Figure 3.21 shows that protocols where only 1 proposal at a time are made consistently perform worse than protocols where 10 proposals per gene family are made (I only show the *hot* configurations here because they were the one that yielded the best score improvement). The increase in computational time per round (*i.e.*, less rounds could be done in 4 hours under these configurations) incurred when testing more times, is compensated by a faster global cost optimization[21].

[20] Note that this does not correspond to the temperature that determines whether or not we accept a proposal according to its global score.

[21] In these tests I compared the capacity of different methods to optimize the global score in a given amount of time. While this is legitimate because it accounts for the differences between methods in terms of computational cost in practice, it should also be noted that it leads implementation to play a role in the results. In particular, the method using only CCP distribution

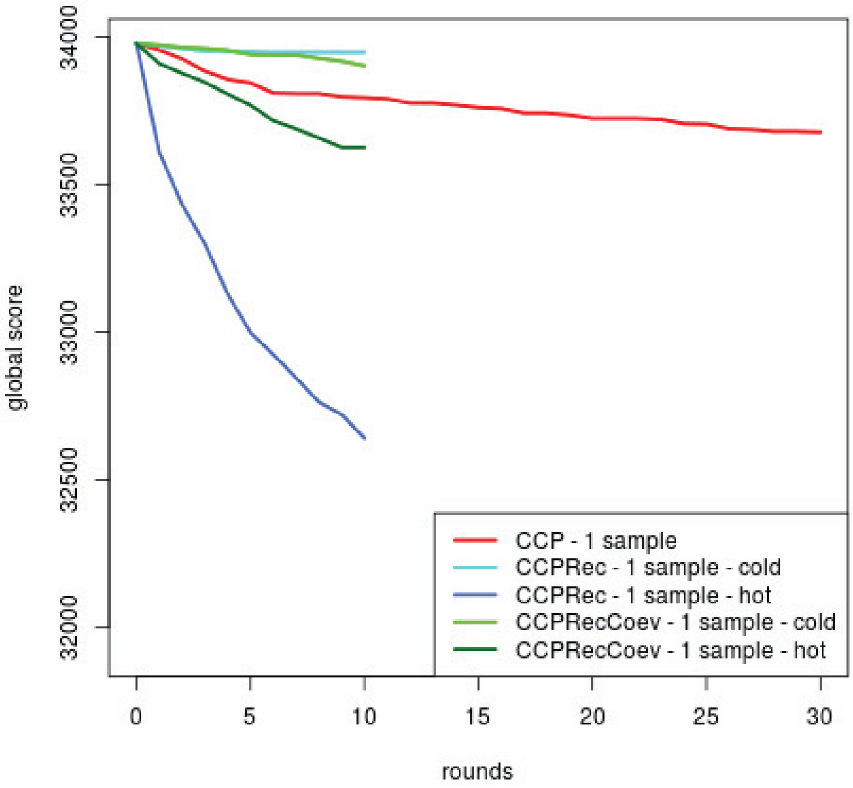


Figure 3.20: Evolution of the global score with different methods new solutions proposal. Each protocol was run for about 4 hours (30 rounds with the *CCP - 1 sample* method , 10 rounds with the other methods).

Overall, the method that yields the best global score improvement here is the one that considers jointly sequence (through the CCP distribution) and reconciliation information (*CCPRec*) with a high temperature (for both conditions of samples number).

The results suggest that when the method used for proposing new solutions is considering jointly CCP distribution and reconciliation (and co-events), a higher temperature leads to a faster global score optimization. The higher the temperature, the higher the chances of proposing solutions deviating from the most parsimonious

information relies on the original C++ implementation of DeCoSTAR while the other methods (*CCPRec* and *CCPRecCoev*) rely on a simpler "proof-of-concept" implementation in python. However, even this non-optimized implementations outperform CCP distribution sampling (when the temperature parameter is high enough) in terms of global score optimization speed, which is quite encouraging.

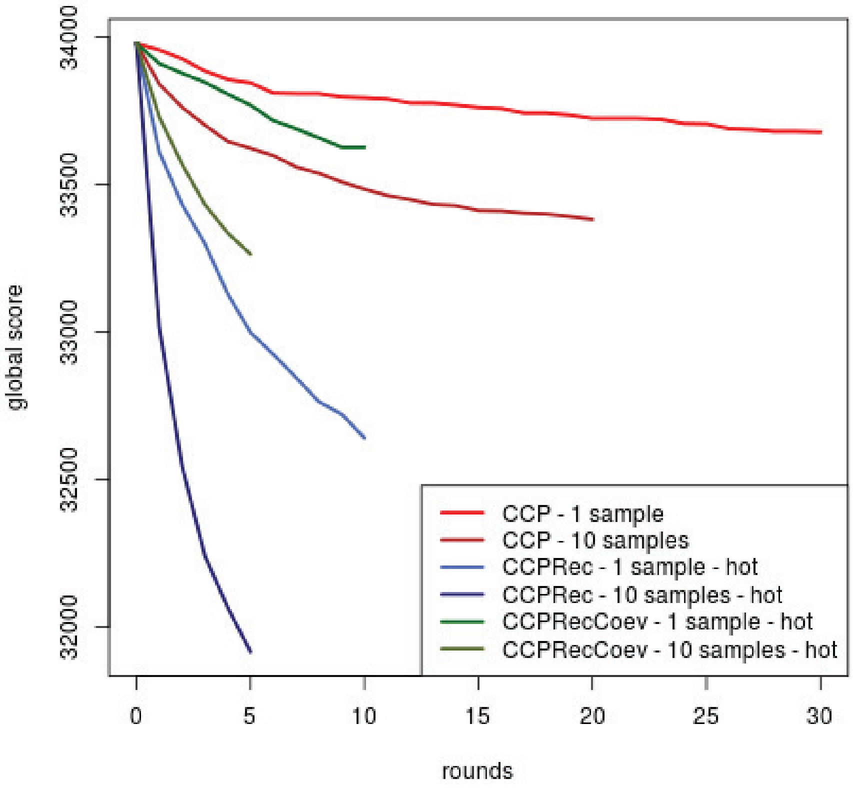


Figure 3.21: Evolution of the global score with different methods new solutions proposal. Each protocol was run for about 4 hours.

one which corresponds to the initial solution given by TERA. From there I can conclude that the good solutions (with respect to the global score) are not the most parsimonious ones (with respect to topology and reconciliation).

Overall, the methods lead to a score optimization (except for the undirected random topology sampling). In particular, the one that yields the best score optimization jointly considers the CCP distribution and reconciliation (*CCPRec*) and it would appear that adding the possibility to form co-events with neighbouring families (*CCPRecCoev*) is actually detrimental to the global score optimization. This may be due to the fact that gene families actually co-evolve with many other gene families while only one neighbouring family is considered when looking for guide events. Additional methodological developments to be able to take multiple guide families into account at once when reconciling may solve this problem.

I also compared the size of events (in terms of number of genes, obtained through the analysis of co-events) between the initial solution and the last round of the condition that yielded the best global score improvement. I observe that the average loss size is $\approx 12\%$ higher in the optimized solution (for a value of ≈ 1.2818 genes per loss). The average duplication size also increases by $\approx 7\%$, for a final value of ≈ 1.1208. While this was expected, it is a good sign to see that improving the global score indeed leads to the inference of longer co-events and therefore maybe to a better congruence between gene trees.

3.4.4 The effect of the global score optimization on independent measures of ancestral genome quality

When reconstructing the reconciliations and adjacency histories of a group of gene families, a way to evaluate the quality of the results is to assess the size of the ancestral genomes inferred through the reconciliations, and the linearity of the graph formed by the inferred ancestral adjacencies (where genes are nodes and adjacencies are edges between nodes)[22].

When reconciling, one actually infers ancestral genome content, and thus ancestral genome size. If we presume that, biologically speaking, the genome sizes should not be particularly different between ancestral and extant species, then we can use this difference as a basis to compare different sets of reconciliations. Figure 3.22 shows the variation, round after round, of inferred ancestral genome sizes under some of the configurations that yielded the best global score decrease in the previous analysis on the data-set containing 500 mammalian gene families. Ancestral genomes display a larger size than extant ones. The mean ancestral size seems to increase during the rounds and stabilizes a little above 650. Compared to the initial solutions, the genomes size did not increase by much ($\approx 3\%$ at the maximum) and the different methods seem to stabilize around the same size after a few rounds, despite the fact that they do not correspond to the same global score (cf. Figure 3.21). This stability and convergence is a good sign and it gives hope about the robustness of the optimization and it also shows that improvements in the global score are not necessarily done at the detriment of ancestral genome sizes.

[22] Indeed, in the article about DeCoSTAR we assessed the linearity of extant and ancestral genomes in the *Anopheles* data-set.

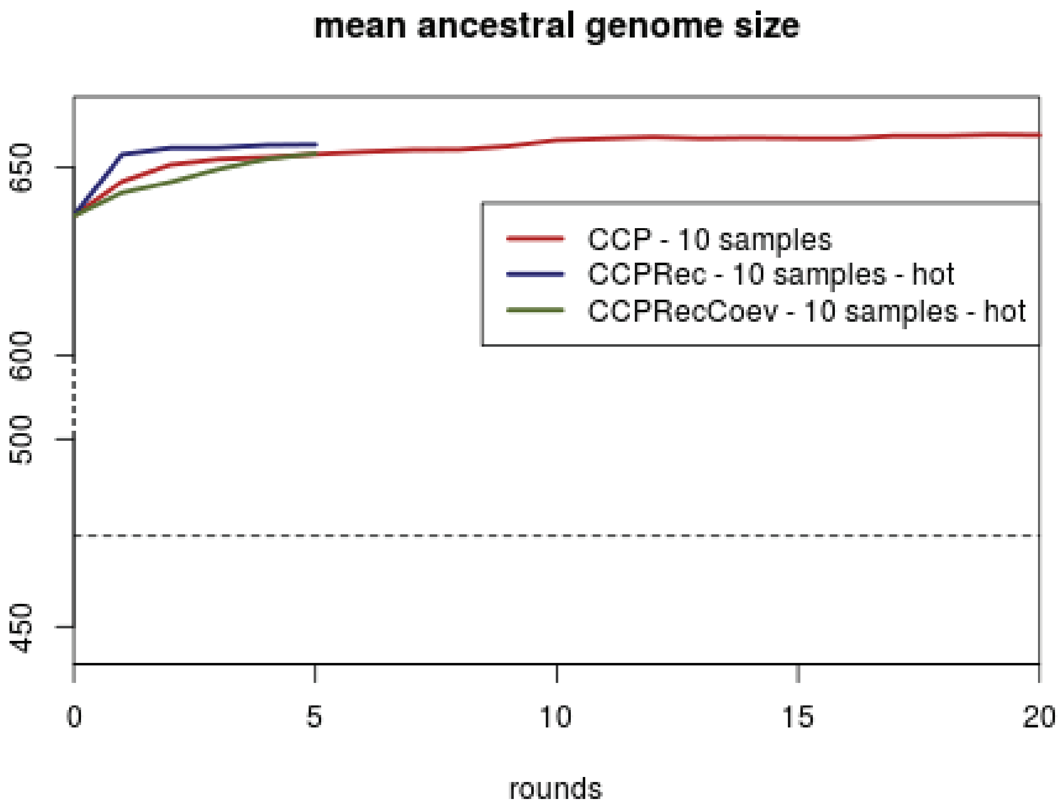


Figure 3.22: Evolution of mean ancestral genome size across rounds of optimization of the global score. The mean extant genome size is displayed as a dashed horizontal black line (474.1).

The second criterion that I use to assess the quality of the solutions is the linearity of ancestral genomes. Linearity is a relevant criterion only if we actually expect the ancestral adjacency graph to be linear, as may be the case when adjacencies are used to model gene order along chromosomes. This also comes down to making the hypothesis that good gene trees (biologically speaking) will lead the inference of linear ancestral genomes. While this may not always be the case (because of limitations in the method used to infer ancestral adjacencies for instance), this hypothesis still seems reasonable to a degree, as the extant genomes that we use as input are themselves linear and the non-linearity of ancestral genomes can be used to pin-point some incongruence between the trees of neighbouring genes (cf. the article about an ancestral *Yersinia pestis* genome inference).

Figure 3.23 presents the evolution of two metrics of the linearity of ancestral genomes. The first metric is the proportion of genes with degree 2 (*i.e.*, genes with exactly two neighbours) in ancestral genomes. A low value of this metric could either mean that the ancestral genomes contain many non-linear patterns (*i.e.*, many nodes with a degree above 2) or that they are very fragmented (*i.e.*, many nodes with a degree of 0 or 1). The second metric is the proportion of genes with a degree above 2 in ancestral genomes. A high value of this metric is the sign that ancestral genomes contain many non-linear patterns.

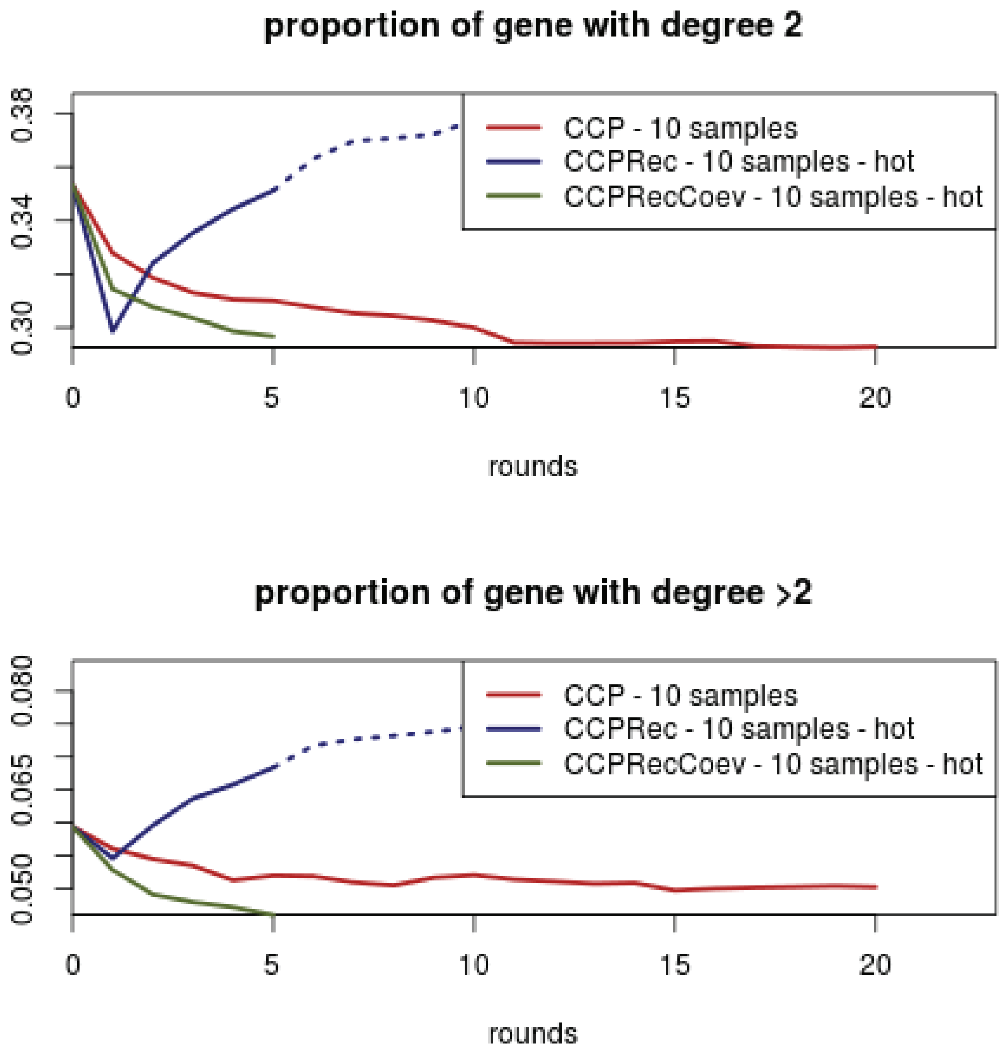


Figure 3.23: Variation of two measures of linearity for ancestral genomes during rounds of global score optimization under different conditions. Dashed lines represent additional rounds.

Of the three displayed conditions, two exhibit a diminution of these linearity measures. However the third one shows that, after an initial decrease, there is an increase of the measures up to the point of the initial solution and even above it. This is interesting because this curves correspond to the condition that yielded the best global score improvement in the previous analysis (cf. Figure 3.21): sampling of new solutions according to a score based on their CCP distribution and reconciliation with the species tree (testing 10 solution per gene family and with a temperature of 2). This trend of increase of the two linearity measures continues when additional rounds are done under these particular conditions and the two measures of linearity continue to go well above their value in the initial solution (dashed lines in the figure).

The increase of both measures indicates that the method that leads to the best global score improvements also seems to lead to the inference of more ancestral adjacencies, sometimes creating non-linear patterns in ancestral genomes. However, here the proportion of genes with degree 2 increased more than the proportion of genes with a degree above 2 so that overall the linearity of the ancestral genomes has increased.

3.4.5 The effect of an alternative starting point

In the previous analysis, I used TERA to obtain the initial solution's topologies and reconciliations. As mentioned before, this implies that the initial solution is quite particular because it specifically jointly optimizes the topology and reconciliation part of the score. The *cold* protocols failed to yield a good global score decrease: a low temperature here leads to a higher probability of the proposition of parsimonious solutions, that do not differ too much from the ones given by TERA. In contrast, the *hot* protocols lead to a better improvement of the global score. This indicates that the trees that were seen as better by the global score were less parsimonious than the ones proposed by TERA in the initial solution. One could therefore ask if this trend of increase in the complexity of the chosen reconciliations would be experienced irrespective of the initial solutions, in which case the optimization of the global score would just be a methods to get reconciliations with more events.

To verify the role of the initial solution in the choice of the global score optimization method, I devised an alternative initial solution, that I will call the random initial solution (to distinguish it from the TERA initial solution).

In this initial solution gene families topologies consist in a random draw in their respective CCP distribution. These topologies where then reconciled using the algorithm for sampling reconciliations with a temperature of 1 (note that when used with a fixed topology, this algorithm effectively becomes one that samples reconciliations with a probability inversely proportional to their cost). These reconciliations were then used in DeCoSTAR along with the extant adjacencies to finish the initial solution.

When tested on the data-set containing 500 mammalian gene families, this initial solution exhibits a much higher global score than the TERA initial solution (56 552.5 against 33 978.4). This difference is mainly due to higher reconciliation and adjacency parts of the score, and is partially compensated by a better topology part of the score (which is coherent with the fact that I chose the topologies according to the CCP distributions alone).

Figure 3.24 presents the evolution of the global score under different conditions when the initial solution is the random one. Contrary to what is observed with the TERA initial solution, the *cold* protocols now lead to a faster global score optimization. This can be explained by the fact that the random initial solution is far from optimal from the point of view of reconciliation and that solutions that fall closer to the most parsimonious reconciliation are favoured.

The observation that the initial solution computed by TERA optimizes faster when sampling less parsimonious reconciliations (higher temperature), but that the random initial solution does the converse (more parsimonious reconciliations are favoured) hints at the idea that the optimal solution reconciliations lie in between these two initial conditions (in terms of reconciliation costs).

3.4.6 Shuffling adjacencies

When choosing new topologies and reconciliations that lower the global score, I make the assumption that there is a relevant signal about the evolution of genes families in their neighbours. In other words, I presume that the gene families that are linked by adjacencies are more likely to be congruent than the ones that do not share any adjacencies.

Under this hypothesis, the signal coming from the gene sequence and the gene reconciliation with the species tree should bear the mark of this co-evolution between neighbours and should support solutions where neighbours are in agreement (which in the context of the global score, translates into more co-events and more parsi-

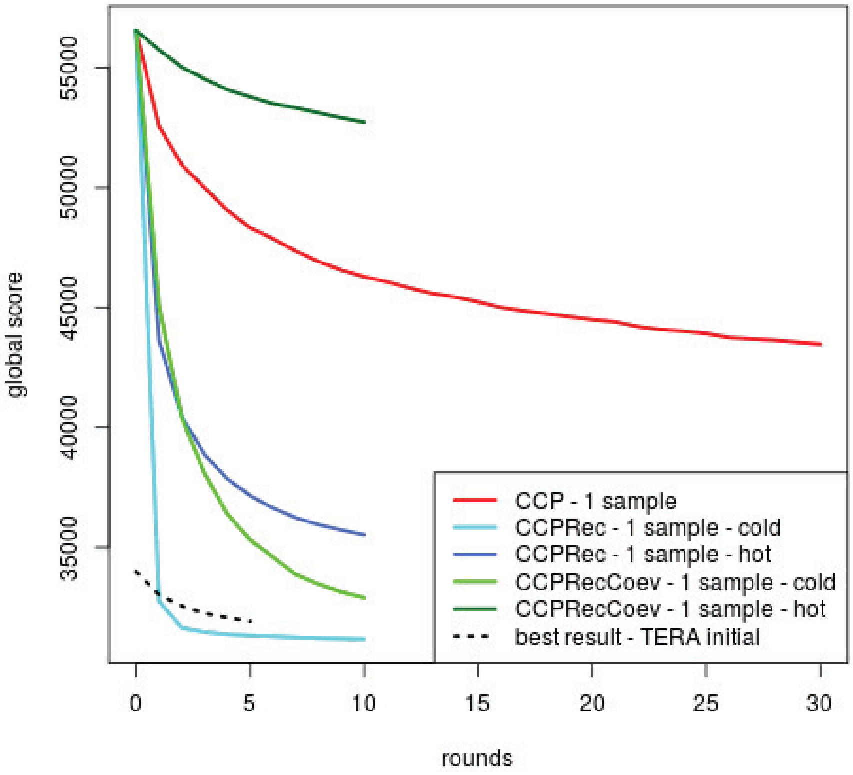


Figure 3.24: Evolution of the global score with different methods new solutions proposal and the random initial solution as a starting point. Each protocol was run for about 4 hours. The curve of the protocol that yielded the best improvement from the TERA inital solution is shown as well (black dashed line)

monious adjacencies histories). However, if this hypothesis is false, then solutions which display a better congruence with the neighbours would not be particularly well supported by the sequence and reconciliation information.

To test this, I performed attempts at global score optimization on the data-set containing 500 mammalian gene families where I randomly shuffled the adjacencies inside the extant genomes[23].

The idea behind this is that if there is indeed information to be found in the neighbours of gene families, then the data-set with shuffled adjacencies would not

[23] I generate a shuffled adjacency by randomly choosing two genes in the same species and putting an adjacency between them. Shuffling was done so that the number of adjacencies in each extant genomes did not change.

have access to this information and consequently I would expect a lower rate of global score optimization than with the original adjacencies (because with meaningful adjacencies, it is more likely that a solution satisfying one criterion will also satisfy the other). Conversely, If the rate of global score optimization stays the same with shuffled adjacencies, this means that the global score optimization process does not exploit any particular information. This in turn may come from the absence of such an information (but see this section in the introduction for elements of proof stating that we do expect some information to be present), or from a weight for adjacencies and co-events that is too high and which leads to neglecting sequence and reconciliation information.

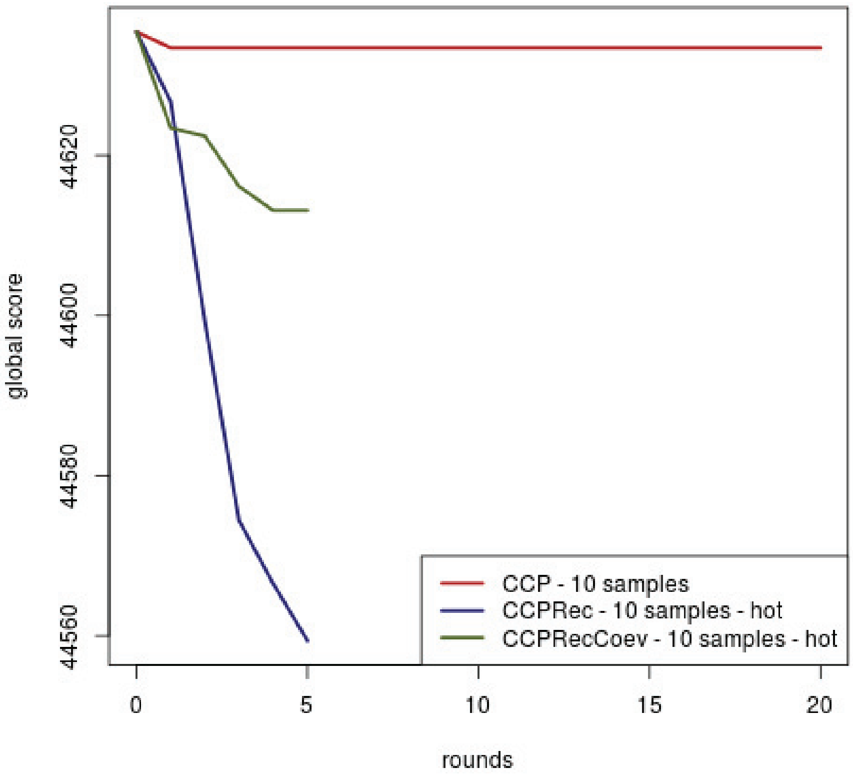


Figure 3.25: Evolution of the global score with different methods new solutions proposal when adjacencies have been shuffled. Each protocol was run for about 4 hours.

The results of global score optimization rounds under different protocols when extant adjacencies have been shuffled are shown in Figure 3.25. I observe both a higher initial score (44 635.4 against 33 978.4) and a lower rate of global score optimization (0.2% against 6.1% with the better protocol) than with the non-shuffled data-set. Co-events in the shuffled data-set are smaller (by $\approx 21\%$, for losses and $\approx 10\%$ for duplications) than in the non-shuffled one and the optimization also does not increase their size by more than $\approx 0.6\%$ (for more than ten times as much in the non shuffled data-set).

This is coherent with what I expected under the hypothesis that the signal coming from adjacencies is complementary (rather than opposed) to the one found in the sequence and the reconciliation and thus this hypothesis is not rejected.

In other words, there is some information in the adjacencies and co-events about the histories of individual genes that the global score improvement method retrieves.

3.4.7 1000 gene families mammalian data-set results

To obtain values for the different weights of the global score for this data-set, I used a heuristic inspired by the one presented in Scornavacca *et al.* [2014]. In order to reduce the number of parameters to consider, I fix individual event costs at their default values, which are widely used in reconciliation and adjacency analyses (namely, a duplication costs 2, a loss 1, an adjacency gain 2 and an adjacency breakage 1). The weights that have to be inferred are the weights of the global score parts (respectively the topology, reconciliation and adjacency weights)[24].

Using an initial value of 1 for all weights, I computed the equivalent of an initial solution (ecceTERA followed by DeCoSTAR). I then estimated new weights using the formula:

$$new\ weight_x = -log(\frac{x}{topology + N_{reconciliation\ event} + N_{adjacency\ event}})$$

Where x takes the values of $topology$, $N_{reconciliation\ event}$ or $N_{adjacency\ event}$ when I respectively estimate the new weight for the topology, reconciliation or adjacency parts of the global score. *topology* corresponds to the topologic disagreement with

[24] Consequently, even though I fixed both adjacency gain cost and duplication cost to 2, they are not necessarily equivalent in the global score because they are always weighted according to their nature (adjacency and reconciliation).

alignements (*i.e.*, the topology score, unweighted), $N_{reconciliation\ event}$ is the number of reconciliation events (losses and duplications, taking co-events into account) and $N_{adjacency\ event}$ is the number of adjacency events (adjacency gains and breakages)[25].

Using these new weights, I compute the initial solution whose global score I will optimize. Drawing on the previous analyses done on the smaller mammalian data-set, I did optimization rounds with the protocol that yielded the best results: sampling several new solutions using both topology and reconciliation information at a high temperature (protocol "CCPRec - 10 samples - hot" in Figure 3.21).

In a first optimization stage, the rounds where done with a temperature of 0 for new solution acceptance, meaning that the solution was only accepted if it lead to a global score decrease. Then, temperature was managed as follow : if an optimization round accepted new solutions for less than 10% of the gene families, the temperature was multiplied by 2 for the next round (if the temperature was 0, then it became 1). Conversely, if an optimization round lead to an increase in the global score (compared with the previous round), the temperature was divided by 2 for the next round. In this manner I hope to avoid local minima while still converging toward lower values of the global score.

I performed 35 rounds of optimization under these conditions. As expected, the global score effectively gets better from round to round. Furthermore, the results observed on the smaller mammalian data-set with respect to genome linearity are also seen here : I observe an increase from an initial value of proportion of gene with degree two at ≈ 0.351 to a final value of ≈ 0.384 after an initial decrease during the first rounds.

The final solution contains more reconciliation events that the initial one (both when taking into account co-events or not) (taking into account co-events, it goes from 14 227 losses and 4 133 duplications to 14 413 losses and 4 287 duplications, or respectively a $\approx 1\%$ and $\approx 3\%$ increase). This increase is accompanied by an increase in the size of co-events : from an average of 1.287 genes per loss to an average of 1.417 (an $\approx 10\%$ increase) and from an average of 1.099 genes per duplication to an average of 1.191 (an $\approx 8\%$ increase).

Figures 3.26 and 3.27 present the evolution of the repartition of of co-event sizes, respectively for losses and duplications, indeed showing a shift toward events with

[25] Here, the new weights are 0.7149079 for the topology, 1.212417 for the reconciliation 1.545124 for the adjacency.

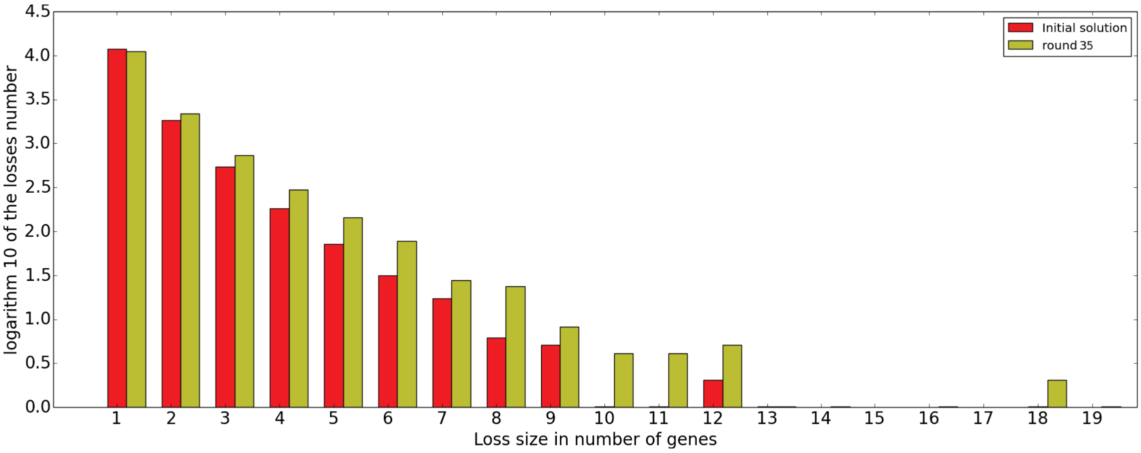


Figure 3.26: Repartition of the loss sizes in number of genes between the initial solution and the final one.

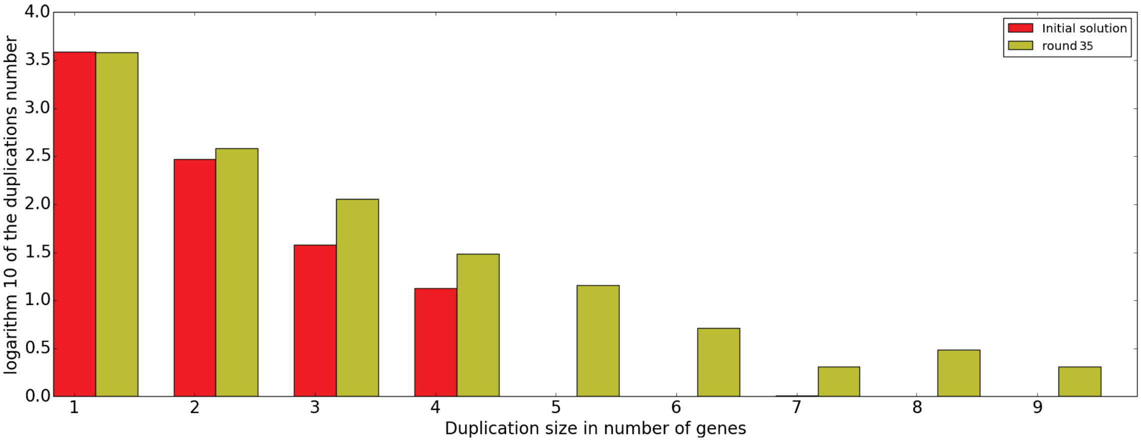


Figure 3.27: Repartition of the duplication sizes in number of genes between the initial solution and the final one.

a bigger size (but with an otherwise unchanged distribution shape).

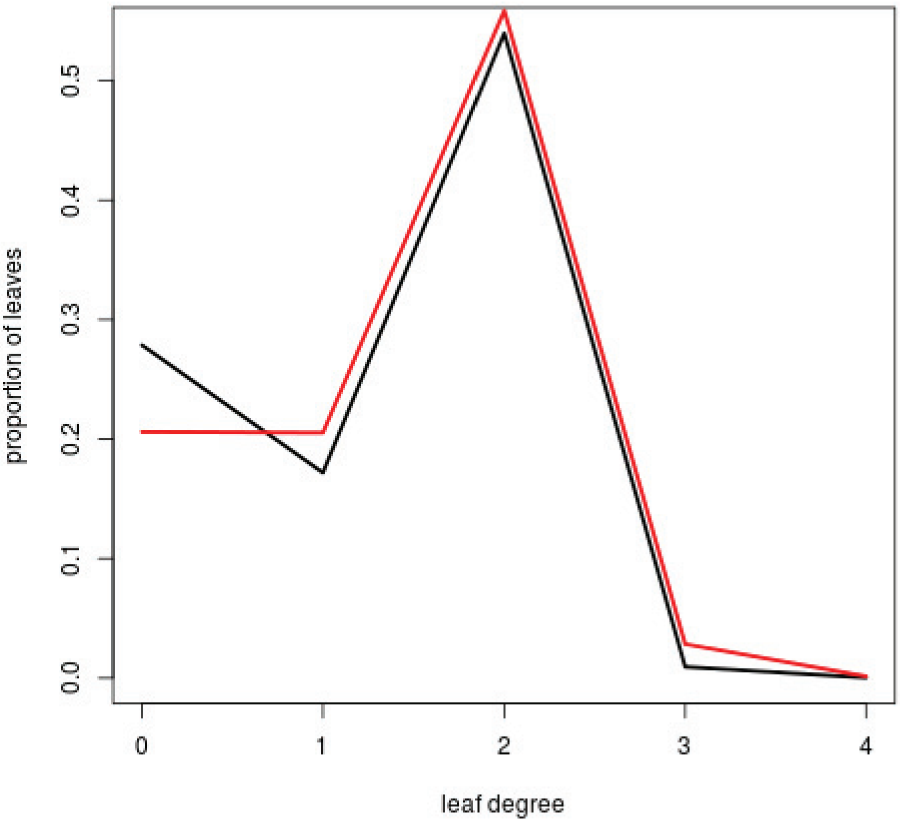


Figure 3.28: Comparison of the degrees of leaves in extant mammalian genomes between the initial (black line) and final solution (red line).

In the meantime the number of events of adjacencies histories have decreased. Adjacency gains have seen their number reduced by ≈ 5% (from 8 699 to 8 310) and the adjacency breakage number was reduced by ≈ 65% (from 2 795 to 956). This high decrease of the number of adjacency breakages is linked to the inference of new adjacencies in extant genomes (remember that some of these mammalian genomes assembly is of low quality and thus many extant adjacencies are missing from the input data) : we go from 1 193 newly inferred extant adjacencies in the initial solution to 3 456 in the final one. Figure 3.28 shows that this difference mainly corresponds to adjacencies gained by nodes that previously had none and that while the final solution implies more leaves with more than 3 neighbours (non-linear patterns that I want to avoid) this is not the case for the majority of impacted nodes.

The observation that the optimization of the global score is linked to a better linearity in ancestral genomes and better extant assemblies (*i.e.*, new extant adjacencies) is truly supportive of my idea that a more integrative view of gene histories leads to more congruence among them and a more coherent history of genomes. The global score optimization also lead to a general increase in the size of events of duplications and losses (compared to the initial solution), which is supportive as well. Through my methods I am able to deliver a *co-evolution-aware* estimate that differ from the estimates yielded by methods that consider genes independently[26].

3.4.8 Fungal data-set results

Methods to obtain the initial solution, weights for the score and managing acceptance temperature were the same as the ones used for the data-set of 1000 mammalian gene families (but adding a cost for horizontal transfer events, which were absent from the mammalian analysis, of 3)[27].

Using the observation on the 500 mammalian gene families data-set that the efficiency of different optimization methods could reliably be estimated after only one round of optimization, I performed a first round of optimization on this fungal data-set and determined that the protocol that yielded the highest rate of global score decrease per unit of time was the sampling in CCP distribution only, with multiple samples per family.

23 rounds of optimization were performed under these conditions. Contrary to what was observed in the mammalian data-sets, there does not appear to be an increase in linearity measures during the optimization (there is actually a decrease: from an initial value of proportion of gene with degree two of ≈ 0.420 to a final value of ≈ 0.311).

Table 3.3 presents the changes in the number and average size (in number of genes) of different kind of events (when taking co-events into account) while Figures 3.29, 3.30 and 3.31 exhibit the changes in the repartition of these events sizes (in

[26] However note that it is unlikely that the sizes reported here are the real ones, in part because the global score optimization is not complete for this data-set, and in other part because of imperfections of the model.

[27] Here the weights are 0.7465697 for topologies, 1.927965 for reconciliations and 0.9660943 for adjacencies.

	duplication	horizontal transfer	loss
Initial number	926	3 331	6 141
Final number	917	3 324	5 974
%change in number	$\approx -1\%$	$\approx -0.2\%$	$\approx -7\%$
Initial average size	≈ 1.161	≈ 1.065	≈ 1.244
Final average size	≈ 1.162	≈ 1.211	≈ 1.341
% change in size	$\approx 0.1\%$	$\approx 14\%$	$\approx 8\%$

Table 3.3: Evolution of the number and average size (in number of genes) of co-events in the fungal data-set.

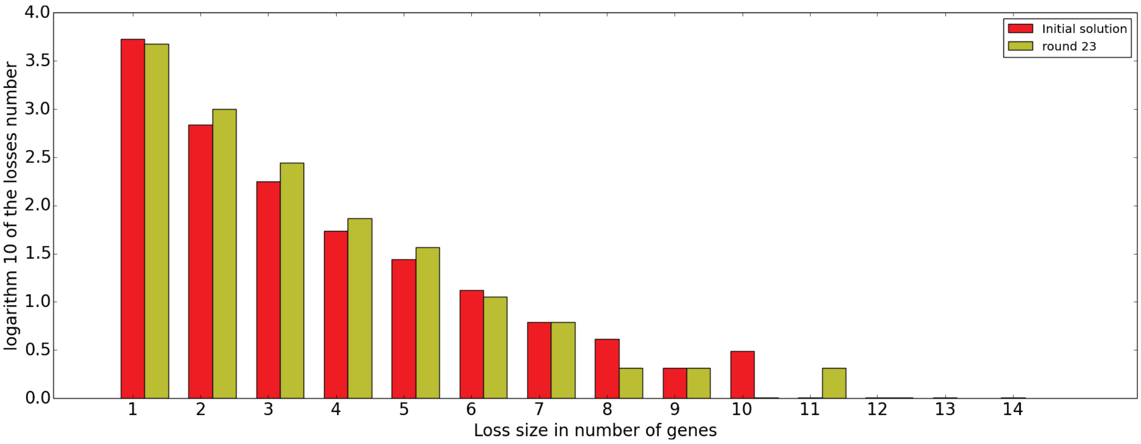


Figure 3.29: Repartition of the loss sizes in number of genes between the initial solution and the final one.

number of genes).

The number of events decreases or stagnates (rather than increases, as was the case in the mammalian data-set). For duplications, this is accompanied by a decrease in the number of gene duplications, which is coherent with the relative stability ($\approx 0.1\%$ of change) of the average size of duplications where horizontal transfer and losses lengthen.

During the optimization, I observe a slight increase in the number of adjacency gains occuring (from 25 103 in the initial solution to 25 164, corresponding to an $\approx 0.2\%$ increase) while the number of adjacency breakages decreases by approximately 4% (from 2 422 adjacency breakages to 2 319). Again, these results are different from the observation made on the mammalian data-set where both events saw their number decrease, in particular adjacency breakages. However, in the mammalian data-set these decreases were linked with the inference of new extant adjacencies, something that was not relevant and thus not made possible in this fungal data-set (because it is composed of fully annotated and assembled genomes).

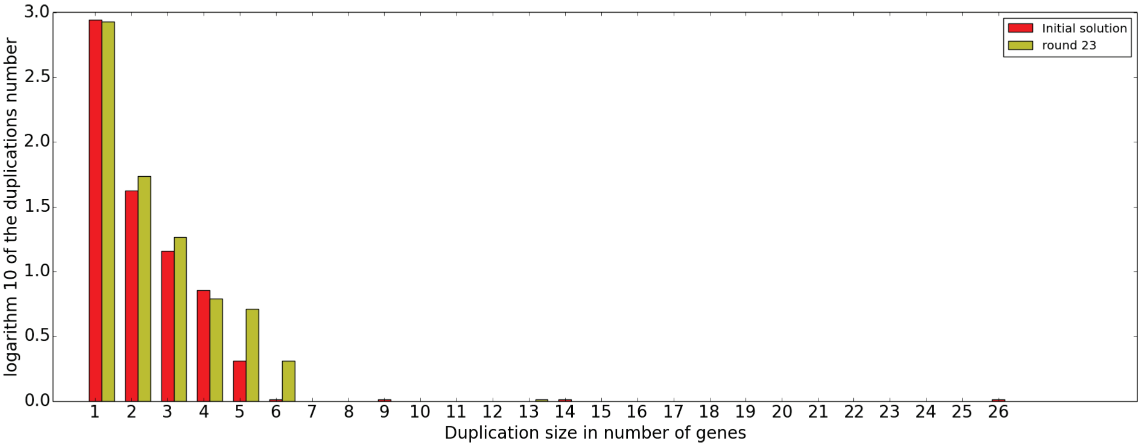


Figure 3.30: Repartition of the duplication sizes in number of genes between the initial solution and the final one.

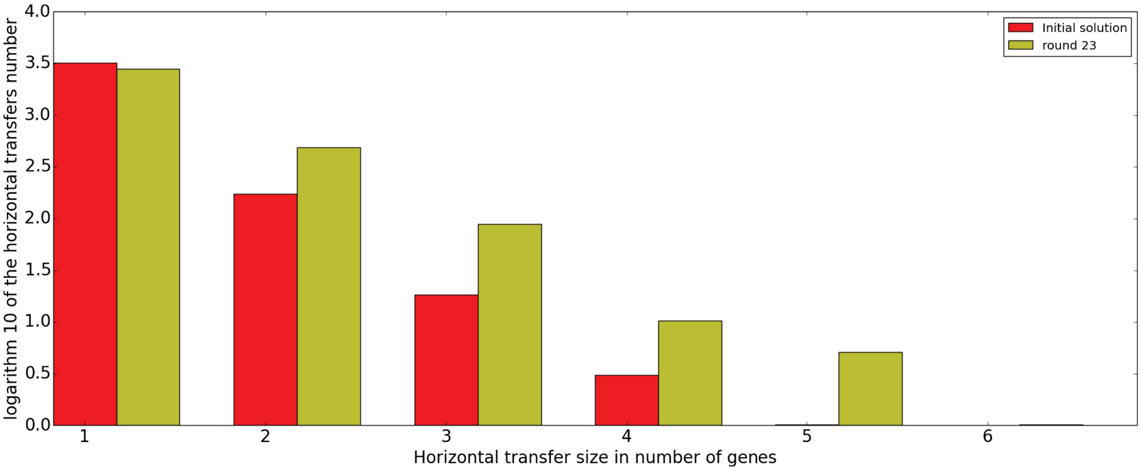


Figure 3.31: Repartition of the horizontal transfer sizes in number of genes between the initial solution and the final one.

Aside from this difference coming from the nature of extant adjacencies in the data-sets, these results on the fungal data-set and in the presence of horizontal transfer still show several key differences with the results obtained on the mammalian data-set.

It is worth noting that the bigger size of the data-set and the addition of transfers significantly increased the time needed for a round of optimization. In particular the presence of transfers have greatly impacted the complexity of the two methods to propose new solutions that make use of more co-evolution information (sequence and reconciliation, and sequence, reconciliation and co-event). This has played in the choice of the proposition method for this data-set and I expect an overall lower per-round optimization rate than on the mammalian data-set.

This may explain the results on the evolution of linearity in this data-set : in the mammalian data-set the (see Figure 3.23) increase in linearity measure was preceded by a decrease phase. If this data-set optimizes more slowly, it is possible that it is still in that initial decreasing phase. Alternatively, using the CCP distribution based proposition method perhaps skews the results too much toward non-linear solutions. Finally it is also possible that this is dependent on the data itself and that linear solutions are just hard to attain.

On that note, when comparing the mammalian and fungal data-set another point of divergence lies in the weights that were inferred for the different parts of the score. In particular, the mammalian data-set gives more weight to adjacencies than reconciliations (1.5 for adjacencies against 1.2 for reconciliations) while the fungal data-set consider adjacency to be the less important of the two (1.0 for adjacencies against 1.9 for reconciliations).

This implies that fungal global score (compared to its mammalian counterpart) particularly favours solutions with less reconciliation events (accounting for co-events) even when they imply more adjacency events (or, in other words, the number of necessary reconciliation events to loose in order to compensate an additional adjacency event in the global score is lower in the fungal data-set than in the mammalian one).

The relatively low weight of the adjacency in this data-set is due to the observation of many events of adjacency gains and breakages during the estimation of the costs. If we consider that the fungal data-set spans about 5 times as much evolutionary time than the mammalian one[28], we understand that, comparatively, much more events of rearrangement have happened in the fungal data-set and as such the signal of co-evolution contained in the adjacencies has had more time to get lost. This would explain the relatively low importance given to adjacencies as a source of information in the fungal data-set. Note however that the weight of adjacencies is non-negligible so that there is still some signal found in them.

[28] Divergence times estimated with `timetree.org` *[last accessed 5th of August 2017]*.

The optimization of this data-set leads to, on average, longer events, in particular horizontal transfers that see their size increase the most, to the point where they become longer, on average, than duplications (the longer events remain the losses). Interestingly, despite them having a longer average size than duplication, several duplications are longer than the longest transfer, maybe hinting at a *maximal size* for viable transfers (considering that to be successful a transfer has to be integrated in its host genome), or at least at a lower variance in the size of transfers.

The results also show that, as a mechanism to increase the number of copies of a gene, horizontal transfer seems to be preferred to duplications (despite a lower cost for duplications), which is coherent with the observations made in [Szöllősi *et al.*, 2015] (on the same data-set).

Chapter 4

Discussion / General Conclusion

The global score and the methods I developed to improve it can be seen as a strategy to infer better individual gene trees thanks to the integration of information from multiple sources. But it could also be seen as a work on the reconstruction of the history of whole genomes that takes into account the heterogeneity of the entities (genes families, adjacencies) they contain. In contrast with the hypotheses that are made on the evolution of nucleotides when building a gene history (namely, total co-evolution of nucleotides in the same gene, null co-evolution between nucleotides of different genes), this *genome history reconstruction* does not make the hypothesis that the different genes of the same genome co-evolve completely (*i.e.*, they each have a different tree and reconciliation), nor that they do not co-evolve at all (*i.e.*, the co-events and adjacency part of the score favour and describe co-evolution between adjacent genes).

The score that I propose and the heuristic to optimize it are, in their current implementations, tools that may benefit from a few further methodological improvements (aside from simple optimization work).

It would be very interesting in particular to adapt them to an adjacency problem other than the ancestral gene order one. As I already mentioned the formulas of DeCoSTAR are inherently tuned to this problem but they could be modified to accommodate other biological networks such as metabolisms or protein interaction networks. Aside from the inference of ancestral metabolisms/protein interaction networks, which is interesting in itself, such a development could then be used to answer several questions about the rate of appearance/disappearance of protein interactions (which are linked to biological functions), the conservation of interactions after a duplication, or the gain of new ones after a successful horizontal gene transfer.

Another improvement that I would find interesting is about the way in which co-events are scored. I made the choice to score a co-event as much as a single event, but one could make it depend on the co-event size. For instance, imitating the way gaps are scored in some alignment algorithms, there could be an *opening cost* (paid once per co-event) and an *elongation cost* (paid for each gene in the co-event)[1]. Indeed the way in which the co-events are scored is bound to have an effect on their inferred length, which is one of the thing that I would like to measure.

However even in its current form the method I propose can prove useful. As I show on a mammalian and fungal data-set, it is able to increase the congruence between gene trees inferred independently, thereby leading to more coherent genome histories as evidenced by longer inferred co-events (which requires congruence between neighbouring genes to occur). This increase in congruence is accompanied by a departure of the individual gene histories from parsimony (in term of joint topology and reconciliation). This hints at the idea that gene evolution is indeed non-parsimonious, but also that the signal of reconciliation events that may be lost to the individual gene can be found in its co-evolutionary partners (here, neighbouring genes).

In the mammalian data-set, the global score improvement also leads to the inference of more extant adjacencies, potentially leading to better assemblies. It is also causing the inference of ancestral genomes that are more linear, which may be taken again as the sign of more congruent gene histories.

Without adjacencies, events are seen at the scale of the single gene, but an evolutionary event (a loss for instance) does not happen to a gene: it happens to a chromosomal fragment. It seems more likely that this chromosomal fragment will either be longer or smaller than the gene rather than correspond exactly to its size. More so, this chromosomal fragment may only overlap partially with the gene. The concept of co-events between adjacent genes that I develop here still uses the gene as its smallest unit and as such it does not consider cases where an event is smaller than the gene, or partially overlaps two genes (these questions are closer to the homology detection and alignment problems). However co-events allow access to the large events of loss, duplication and transfer that span multiple genes.

[1] Technically, what I did came down to fix the co-event *opening cost* at the cost of a single event, and fix the *elongation cost* at 0.

Indeed, for the mammalian and fungal data-set I am able to provide distributions of the sizes of events (at least in number of genes) both before and after the global improvement. Furthermore, I show that, as I mentioned I expected, the improvement of the global score leads to the observation of longer events on average (showing that independent inferences of gene histories are likely to underestimate the size of co-events and that the global score can help provide another, *co-evolution-aware* estimate). In both the fungal and mammalian data-set, losses are seen as the (on average) longest events. When they are present (*i.e.*, in the fungal data-set), transfers are the events whose size increased the most during the global score improvement, to the point that they are reported as longer than duplications (while they were shorter before the optimization).

Overall, I think there is reason to believe that the new gene trees obtained through global score optimization are on average better (biologically speaking) than the initial ones. As I already mentioned, they display more congruence between each other, thereby implying a more coherent history of genomes. Moreover, the global score takes into account some signal that is proper to each family (*i.e.*, the topology part of the score) which still allows them to possess an history of their own. In other words, the construction of the global score accounts for cases where there is enough signal in the alignment that the gene has an history different from its neighbours. However if there is strength in the compromise between the different sources of information that I consider, one can also see that it is heavily dependent on the values of the parameters of the score, in particular the weights associated to it different parts. I showed in my results a way to infer them but it makes no doubt that a more complete exploration of the space of parameters, based on a probabilistic model, would be more satisfying (however, I already mentioned that for adjacencies, no model that takes transfers into account exists yet, so that many methodological developments seems needed before a fully probabilistic version of the global score is attainable).

In any case, it could be interesting to design tests to check whether or not the global score indeed leads to better gene trees. One way to do this could be through simulations. However such a simulation should also model adjacencies and gene co-evolution. Another method could be to cut (or duplicate) a single gene in two and then reconstruct the history of each half (or duplicate) both independently and with the global score (considering adjacencies between the different halves/duplicates)

However such an example is more a test of the robustness of the gene tree inference and ignores cases where the co-evolution between tree are not total, but partial (and these cases could be argued to be the more interesting because when there is a total co-evolutionary relationship, one could just put the sequences in the same alignment).

On another note, there are things to be said about the gene as an evolutionary unit. I (re-)defined it as the sum of different co-evolutionary relationships, actually making it a somewhat abstract entity disconnected from its sequence (as it co-evolves with it). In many cases in this document, the gene as a unit could seem limited, either because it was too small (and we add to link genes together to detect bigger events), too big (for the detection of smaller rearrangements for instance). Given the problem it causes, one could be tempted to discard the gene entirely and operate directly at the scale of nucleotides (this comes down to the idea of each nucleotide being its own mini-gene that I mentioned in the introduction). Of course a nucleotide by itself rarely contains enough signal to reconstruct its history. But with a method such as the ones based on the global score, that can combine sequence signal with the signal that is in neighbours, reconstructing the history of the single nucleotide may become accessible. Surely such a thing is not tractable for instances with more than a few nucleotides and it may be too extreme to discard the gene as an evolutionary unit entirely as there indeed exists blocks of conserved nucleotides, whose link together is important enough that we group them together. However such considerations help put into perspective the hypotheses that we make when we define an evolutionary unit and infer its history, be they for biological, statistical or computational purposes.

Bibliography

Abby, S. S., Tannier, E., Gouy, M., and Daubin, V. (2012). Lateral gene transfer as a support for the tree of life. *Proceedings of the National Academy of Sciences*, **109**(13), 4962–4967.

Adams, E. N. (1972). Consensus techniques and the comparison of taxonomic trees. *Systematic Biology*, **21**(4), 390–397.

Altschul, S. F., Gish, W., Miller, W., Meyers, E. W., and Lipman, J. D. (1990). Basic Local Alignment Search Tool. *Journal of Molecular Biology*, **215**(3), 403–410.

Anselmetti, Y., *et al.* (2015). Ancestral gene synteny reconstruction improves extant species scaffolding. *BMC Genomics*, **16**(Suppl 10), S11.

Arnaoudova, E., *et al.* (2010). Statistical phylogenetic tree analysis using differences of means. *Frontiers in Neuroscience*, **4**(AUG), 1–7.

Arvestad, L., Berglund, A. C., Lagergren, J., and Sennblad, B. (2003). Bayesian gene/species tree reconciliation and orthology analysis using MCMC. *Bioinformatics*, **19**(SUPPL. 1), 7–15.

Bansal, M. S., Alm, E. J., and Kellis, M. (2012). Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss. *Bioinformatics*, **28**(12), 283–291.

Bansal, M. S., Wu, Y. C., Alm, E. J., and Kellis, M. (2015). Improved gene tree error correction in the presence of horizontal gene transfer. *Bioinformatics*, **31**(8), 1211–1218.

Barker, D., Meade, A., and Page, M. (2007). Constrained models of evolution lead to improved prediction of functional linkage from correlated gain and loss of genes. *Bioinformatics*, **23**(1), 14–20.

Bérard, S., *et al.* (2012). Evolution of gene neighborhoods within reconciled phylogenies. *Bioinformatics (Oxford, England)*, **28**(18), i382–i388.

Blumenthal, T. (2004). Operons in eukaryotes. *Briefings in Functional Genomics and Proteomics*, **3**(3), 199–211.

Boc, A., Philippe, H., and Makarenkov, V. (2010). Inferring and validating horizontal gene transfer events using bipartition dissimilarity. *Systematic Biology*, **59**(2), 195–211.

Bonizzoni, P., Della Vedova, G., and Dondi, R. (2005). Reconciling a gene tree to a species tree under the duplication cost model. *Theoretical Computer Science*, **347**(1-2), 36–53.

Boussau, B., Guéguen, L., and Gouy, M. (2009). A mixture model and a Hidden Markov Model to simultaneously detect recombination breakpoints and reconstruct phylogenies. *Evolutionary Bioinformatics*, **2009**(5), 67–79.

Boussau, B., *et al.* (2013). Genome-scale coestimation of species and gene trees. *Genome research*, **23**(2), 323–30.

Campbell, V., Legendre, P., and Lapointe, F.-J. (2011). The performance of the Congruence Among Distance Matrices (CADM) test in phylogenetic analysis. *BMC evolutionary biology*, **11**, 64.

Chauve, C., Ponty, Y., and Zanetti, J. P. P. (2015). Evolution of genes neighborhood within reconciled phylogenies: an ensemble approach. *Lecture Notes in Computer Science (Advances in Bioinformatics and Computational Biology)*, **8826 LNBI**, 49–56.

Codoñer, F. M. and Fares, M. A. (2008). Why should we care about molecular coevolution? *Evolutionary Bioinformatics*, **2008**(4), 29–38.

Cotton, J. A. and Wilkinson, M. (2007). Majority-rule supertrees. *Systematic Biology*, **56**(3), 445–452.

Dagan, T. and Martin, W. (2006). The tree of one percent. *Genome biology*, **7**(10), 118.

Daskalakis, C. and Roch, S. (2015). Species Trees from Gene Trees Despite a High Rate of Lateral Genetic Transfer: A Tight Bound. *SIAM ACM Symposium on Discrete Algorithms (SODA'16)*, **1508.01962**(D), 1621–1630.

Daubin, V. and Szöllősi, G. J. (2016). Horizontal Gene Transfer and the Universal Tree of Life. *Cold Spring Harb Perspect Biol*, pages 1–9.

Daubin, V., Moran, N. A., and Ochman, H. (2003a). Phylogenetics and the Cohesion of Bacterial Genomes. *Science*, **301**, 829–832.

Daubin, V., *et al.* (2003b). The source of laterally transferred genes in bacterial genomes. *Genome Biology*, **4**(9), R57.

David, L. A. and Alm, E. J. (2011). Rapid evolutionary innovation during an Archaean genetic expansion. *Nature*, **469**(7328), 93–96.

de Koning, A. P. J., Gu, W., Castoe, T. A., Batzer, M. A., and Pollock, D. D. (2011). Repetitive elements may comprise over Two-Thirds of the human genome. *PLoS Genetics*, **7**(12).

de Oliveira Martins, L., Leal, É., and Kishino, H. (2008). Phylogenetic detection of recombination with a Bayesian prior on the distance between trees. *PLoS ONE*, **3**(7).

Diruggiero, J., *et al.* (2000). Evidence of recent lateral gene transfer among hyperthermophilic Archaea. *Molecular Microbiology*, **38**(4), 684–693.

Doolittle, F. W. (1999). Phylogenetic Classification and the Universal Tree. *Science*, **284**(June), 2124–2128.

Doyon, J.-p., Scornavacca, C., Ranwez, V., and Berry, V. (2010). An efficient algorithm for gene / species trees parsimonious reconciliation with losses , duplications , and transfers. In *Proceedings of the 2010 International Conference on Comparative Genomics, RECOMB-CG'10*, pages 93–108.

Dutheil, J. Y. (2012). Detecting coevolving positions in a molecule: Why and how to account for phylogeny. *Briefings in Bioinformatics*, **13**(2), 228–243.

Earl, D., *et al.* (2014). Alignathon: A competitive assessment of whole genome alignment methods. *Genome research*, **24**(12), 2077–89.

Edgar, R. C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, **32**(5), 1792–7.

Ehrlich, P. R. and Raven, P. H. (1964). Butterflies and Plants: A Study in Coevolution. *Evolution*, **18**(4), 586–608.

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

Felsenstein, J. (1985). Confidence limits on phylogenies: An approach using the bootstrap. *Evolution*, **39**(4), 783–791.

Felsenstein, J. (1988). Phylogenies From Molecular Sequences : Inference and reliability. *Annual review of genetics*, **22**, 521–65.

Fitch, W. M. (1971). Towards defining the course of evolution: minimal change for a specified tree topology. *Syst. Zool.*, **20**(4), 406–416.

Fitch, W. M. and Margoliash, E. (1967). Construction of Phylogenetic Trees. *Science*, **155**(3760), 279–284.

Fitch, W. M. and Markowitz, E. (1970). An improved method for determining codon variability in a gene and its application to the rate of fixation of mutations in evolution. *Biochemical Genetics*, **4**(5), 579–593.

Fuellen, G. (2008). Homology and phylogeny and their automated inference. *Naturwissenschaften*, **95**(6), 469–481.

Fujimoto, M. S., Suvorov, A., Jensen, N. O., Clement, M. J., and Bybee, S. M. (2016). Detecting false positive sequence homology: a machine learning approach. *BMC Bioinformatics*, **17**(1), 101.

Gadagkar, S. R., Rosenberg, M. S., and Kumar, S. (2005). Inferring species phylogenies from multiple genes: concatenated sequence tree versus consensus gene tree. *Journal of experimental zoology. Part B, Molecular and developmental evolution*, **304**(1), 64–74.

Galtier, N. and Daubin, V. (2008). Dealing with incongruence in phylogenomic analyses. *Philosophical Transactions of the Royal Society B: Biological Sciences*, **363**(1512), 4023–4029.

Gavrilets, S. (2014). Models of speciation: Where are we now? *Journal of Heredity*, **105**(S1), 743–755.

Goodman, M., Czelusniak, J., Moore, G. W., Romero-Herrera, A. E., and Matsuda, G. (1979). Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Systematic Biology*, **28**(2), 132.

Gordon, A. D. (1986). Consensus supertrees: The synthesis of rooted trees containing overlapping sets of labeled leaves. *Journal of Classification*, **3**(2), 335–348.

Gregory, T. R. (2001). Coincidence, coevolution, or causation? DNA content, cell size, and the C-value enigma. *Biological reviews of the Cambridge Philosophical Society*, **76**(1), 65–101.

Guindon, S., *et al.* (2010). New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. *Systematic biology*, **59**(3), 307–21.

Heled, J. and Drummond, A. J. (2010). Bayesian Inference of Species Trees from Multilocus Data. *Molecular Biology and Evolution*, **27**(3), 570–580.

Hernandez-Rosales, M., *et al.* (2012). From gene trees to species trees. *BMC bioinformatics*, **13**(Suppl 19), 729–752.

Hogeweg, P. and Hesper, B. (1984). The alignment of sets of sequences and the construction of phyletic trees: An integrated method. *Journal of Molecular Evolution*, **20**(2), 175–186.

Höhna, S. and Drummond, A. J. (2012). Guided tree topology proposals for Bayesian phylogenetic inference. *Systematic Biology*, **61**(1), 1–11.

Hurst, L. D., Pál, C., and Lercher, M. J. (2004). The evolutionary dynamics of eukaryotic gene order. *Nature Reviews Genetics*, **5**(4), 299–310.

Innan, H. and Kondrashov, F. (2010). The evolution of gene duplications: classifying and distinguishing between models. *Nature reviews*, **11**(2), 97–108.

Jacob, F., Perrin, D., Sanchez, C., and Monod, J. (1960). The Operon : A Group of Genes Whose Expression is Coordinated by an Operator. *COMPTES RENDUS DES SEANCES DE L'AcADEMIE DES SCIENCES*, **250**, 1727–1729.

Jacox, E., Chauve, C., Szöllősi, G. J., Ponty, Y., and Scornavacca, C. (2016). ecceTERA : Comprehensive gene tree-species tree reconciliation using parsimony. *Bioinformatics (Oxford, England)*, pages 1–3.

Jean, P.-A. (2013). Algorithmique pour l'évolution des interactions géniques. Technical report, Université de Montpellier, Montpellier.

Juan, D., Pazos, F., and Valencia, A. (2008). High-confidence prediction of global interactomes based on genome-wide coevolutionary networks. *Proceedings of the National Academy of Sciences of the United States of America*, **105**(3), 934–939.

Kahn, D., Rezvoy, C., and Vivien, F. (2008). Parallel large scale inference of protein domain families. *Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS*, pages 72–79.

Lacey, J. C., Weber, A. L., and White Jr., W. E. (1975). A model for the coevolution of the genetic code and the process of protein synthesis: Review and assessment. *Orig Life*, **6**(1-2), 273–283.

Lafond, M., Swenson, K. M., and El-mabrouk, N. (2012). An Optimal Reconciliation Algorithm for Gene Trees with Polytomies. In *WABI*, pages 106–122.

Lafond, M., Semeria, M., Swenson, K. M., Tannier, E., and El-Mabrouk, N. (2013). Gene tree correction guided by orthology. *BMC Bioinformatics*, **14**(Suppl 15), S5.

Larget, B. (1999). Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees Markov chain Monte Carlo. *Molecular Biology and Evolution*, **16**(6), 750–759.

Lee, J. M. and Sonnhammer, E. L. L. (2003). Genomic gene clustering analysis of pathways in eukaryotes. *Genome Research*, **13**(5), 875–882.

Legendre, P. and Lapointe, F.-J. (2004). ASSESSING CONGRUENCEAMONG DISTANCE MATRICES: SINGLE-MALT SCOTCH WHISKIES REVISITED. *Australian & New Zealand Journal of Statistics*, **46**(March 2001), 615–629.

Liang, Z., Xu, M., Teng, M., Niu, L., and Wu, J. (2010). Coevolution is a short-distance force at the protein interaction level and correlates with the modular organization of protein networks. *FEBS Letters*, **584**(19), 4237–4240.

Liu, L., Yu, L., Kubatko, L., Pearl, D. K., and Edwards, S. V. (2009). Coalescent methods for estimating phylogenetic trees. *Molecular Phylogenetics and Evolution*, **53**(1), 320–328.

Löytynoja, A. and Goldman, N. (2005). An algorithm for progressive multiple alignment of sequences with insertions. *Proceedings of the National Academy of Sciences of the United States of America*, **102**(30), 10557–62.

Luo, F., *et al.* (2007). Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory. *BMC Bioinformatics*, **8**(1), 299.

Ma, J., *et al.* (2006). Reconstructing contiguous regions of an ancestral genome. *Genome Research*, **16**, 1557–65.

Maddison, W. (1997). Gene Trees in Species Trees. *Systematic Biology*, **46**(3), 523–536.

McCann, A., Cotton, J. a., and McInerney, J. O. (2008). The tree of genomes: an empirical comparison of genome-phylogeny reconstruction methods. *BMC evolutionary biology*, **8**, 312.

Minh, B. Q., Nguyen, M. A. T., and Von Haeseler, A. (2013). Ultrafast approximation for phylogenetic bootstrap. *Molecular Biology and Evolution*, **30**(5), 1188–1195.

Minin, V. N., Dorman, K. S., Fang, F., and Suchard, M. A. (2005). Dual multiple change-point model leads to more accurate recombination detection. *Bioinformatics*, **21**(13), 3034–3042.

Moore, A. D., Grath, S., Schüler, A., Huylmans, A. K., and Bornberg-Bauer, E. (2013). Quantification and functional analysis of modular protein evolution in a dense phylogenetic tree. *Biochimica et biophysica acta*, **1834**(5), 898–907.

Nakhleh, L. (2013). Computational approaches to species phylogeny inference and gene tree reconciliation. *Trends in Ecology and Evolution*, **28**(12), 719–728.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, **48**(3), 443–453.

Nguyen, L. T., Schmidt, H. A., Von Haeseler, A., and Minh, B. Q. (2015). IQ-TREE: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular Biology and Evolution*, **32**(1), 268–274.

Ochman, H., Lawrence, J. G., and Groisman, E. a. (2000). Lateral gene transfer and the nature of bacterial innovation. *Nature*, **405**(6784), 299–304.

Overbeek, R., Fonstein, M., D'Souza, M., Pusch, G. D., and Maltsev, N. (1999). The use of gene clusters to infer functional coupling. *Proceedings of the National Academy of Sciences of the United States of America*, **96**(6), 2896–2901.

Pais, F. S.-M., Ruy, P. d. C., Oliveira, G., and Coimbra, R. S. (2014). Assessing the efficiency of multiple sequence alignment programs. *Algorithms for Molecular Biology : AMB*, **9**, 4.

Pasek, S., Risler, J.-L., and Brézellec, P. (2006). Gene fusion/fission is a major contributor to evolution of multi-domain bacterial proteins. *Bioinformatics (Oxford, England)*, **22**(12), 1418–23.

Patterson, M., Szöllősi, G., Daubin, V., and Tannier, E. (2013). Lateral gene transfer, rearrangement, reconciliation. *BMC Bioinformatics*, **14**(Suppl 15), S4.

Pazos, F. and Valencia, A. (2001). Similarity of phylogenetic trees as indicator of protein-protein interaction. *Protein engineering*, **14**(9), 609–614.

Pazos, F., Ranea, J. a. G., Juan, D., and Sternberg, M. J. E. (2005). Assessing protein co-evolution in the context of the tree of life assists in the prediction of the interactome. *Journal of Molecular Biology*, **352**(4), 1002–1015.

Pellegrini, M., Marcotte, E. M., Thompson, M. J., Eisenberg, D., and Yeates, T. O. (1999). Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *Proceedings of the National Academy of Sciences*, **96**(8), 4285–4288.

Penel, S., *et al.* (2009). Databases of homologous gene families for comparative genomics. *BMC bioinformatics*, **10 Suppl 6**, S3.

Pouyet, F., Bailly-Bechet, M., Mouchiroud, D., and Guéguen, L. (2016). SENCA: A Multilayered Codon Model to Study the Origins and Dynamics of Codon Usage. *Genome Biology and Evolution*, **8**(8), 2427–2441.

Promponas, V. J., Ouzounis, C. A., and Iliopoulos, I. (2014). Experimental evidence validating the computational inference of functional associations from gene fusion events: a critical survey. *Briefings in bioinformatics*, **15**(3), 443–54.

Rannala, B. and Yang, Z. (2003). Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci. *Genetics*, **164**(4), 1645–1656.

Ranwez, V., *et al.* (2007). PhySIC: a veto supertree method with desirable properties. *Systematic biology*, **56**(5), 798–817.

Rasmussen, M. D. and Kellis, M. (2012). Unified modeling of gene duplication, loss, and coalescence using a locus tree. *Genome research*, **22**(4), 755–65.

Rosewich, L. U. and Kistler, C. (2000). Role of Horizontal Gene Transfer in the Evolution of Fungi. *Annu. Rev. Phytopathol.*, **38**, 325–63.

Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molec.Biol.Evol.*, **4**(4), 406–425.

Sapp, J. (1994). *Evolution by Association*. Oxford University Press.

Scornavacca, C., Jacox, E., and Szollősi, G. J. (2014). Joint amalgamation of most parsimonious reconciled gene trees. *Bioinformatics*, **31**(6), 841–848.

Semeria, M., *et al.* (2015). Probabilistic modeling of the evolution of gene synteny within reconciled phylogenies To cite this version : Probabilistic modeling of the evolution of gene synteny within reconciled phylogenies. *BMC bioinformatics*, **16**(Suppl 14), 1–11.

Shimodaira, H. and Hasegawa, M. (1999). Multiple comparisons of log-likelihoods with applications to phylogenetic inference. *Mol. Biol. Evol.*, **16**, 1114–1116.

Sjöstrand, J., *et al.* (2014). A bayesian method for analyzing lateral gene transfer. *Systematic Biology*, **63**(3), 409–420.

Sokal, R. R. and Michener, C. D. (1958). A Statistical Method for Evaluating Systematic Relationships. *The University of Kansas Science Bulletin*, **38**(February), 1409–1438.

Soyer, O. S. (2012). *Evolutionary systems biology*. New York : Springer.

Stamatakis, A. (2014). RAxML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, **30**(9), 1312–1313.

Stolzer, M., *et al.* (2012). Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees. *Bioinformatics*, **28**(18), 409–415.

Suchard, M. A. (2005). Stochastic models for horizontal gene transfer: Taking a random walk through tree space. *Genetics*, **170**(1), 419–431.

Szöllősi, G. J., Boussau, B., Abby, S. S., Tannier, E., and Daubin, V. (2012). Phylogenetic modeling of lateral gene transfer reconstructs the pattern and relative timing of speciations. *Proceedings of the National Academy of Sciences of the United States of America*, **109**(43), 17513–8.

Szöllősi, G. J., Rosikiewicz, W., Boussau, B., Tannier, E., and Daubin, V. (2013a). Efficient exploration of the space of reconciled gene trees. *Systematic biology*, **62**(6), 901–12.

Szöllősi, G. J., Tannier, E., Lartillot, N., and Daubin, V. (2013b). Lateral gene transfer from the dead. *Systematic Biology*, **62**(3), 386–397.

Szöllősi, G. J., Tannier, E., Daubin, V., and Boussau, B. (2015). The inference of gene trees with species trees. *Systematic Biology*, **64**(1), e42–e62.

Tavano, C. L. and Donohue, T. J. (2006). Development of the bacterial photosynthetic apparatus. *Curr Opin Microbiol*, **9**(6), 625–631.

Than, C., Ruths, D., and Nakhleh, L. (2008). PhyloNet: a software package for analyzing and reconstructing reticulate evolutionary relationships. *BMC Bioinformatics*, **9**, 322.

Thomas, C. M. and Nielsen, K. M. (2005). Mechanisms of, and barriers to, horizontal gene transfer between bacteria. *Nat.Rev.Microbiol.*, **3**(1740-1526 (Print)), 711–721.

Tofigh, A., Hallett, M., and Lagergren, J. (2011). Simultaneous Identification of Duplications and Lateral Transfers. *Computer*, **8**(2), 517–535.

Vernot, B., Stolzer, M., Goldman, A., and Durand, D. (2008). Reconciliation with non-binary species trees. *Journal of computational biology : a journal of computational molecular cell biology*, **15**(8), 981–1006.

Villa-Vialaneix, N., *et al.* (2013). The Structure of a Gene Co-Expression Network Reveals Biological Functions Underlying eQTLs. *PLoS ONE*, **8**(4).

Wapinski, I., Pfeffer, A., Friedman, N., and Regev, A. (2007). Automatic genome-wide reconstruction of phylogenetic gene trees. *Bioinformatics*, **23**(13), 549–558.

Watson, J. D.; Crick, F. H. C. (1953). Molecular Structure of Nucleic Acids URL : .

Wehe, A., Bansal, M. S., Burleigh, J. G., and Eulenstein, O. (2008). DupTree: A program for large-scale phylogenetic analyses using gene tree parsimony. *Bioinformatics*, **24**(13), 1540–1541.

Wu, Y.-C., Rasmussen, M. D., and Kellis, M. (2012). Evolution at the subgene level: domain rearrangements in the Drosophila phylogeny. *Molecular biology and evolution*, **29**(2), 689–705.

Wu, Y.-C., Rasmussen, M. D., Bansal, M. S., and Kellis, M. (2013). TreeFix: Statistically Informed Gene Tree Error Correction Using Species Trees. *Systematic Biology*, **62**(1), 110–120.

Yanai, I., Mellor, J. C., and DeLisi, C. (2002). Identifying functional links between genes using conserved chromosomal proximity. *Trends in Genetics*, **18**(4), 176–179.

Yates, A., *et al.* (2016). Ensembl 2016. *Nucleic Acids Research*, **44**(D1), D710–D716.

Zuckerkandl, E. and Pauling, L. (1965). Molecules as documents of evolutionary history. *Journal of theoretical biology*, **8**(2), 357–366.

Annexes

A Supplementary file for "DeCoSTAR: Reconstructing the Ancestral Organization of Genes or Genomes Using Reconciled Phylogenies"

DeCoSTAR: Ancestral genome or gene organizations using reconciled phylogenies - Supplementary Materials

February 20, 2017

Scaffolding mode

DeCoSTAR can be used to infer some extant adjacencies, typically to account for badly assembled genomes. In this case, the cost to put a new adjacency between two leaves (of the same species) is defined by the formulas:

$c_1(a,b) = -T * log(p)$
$c_0(a,b) = -T * log(1-p)$

Where: $p = F_{adj} * BP$
$T = \frac{Break}{SPI*log(\frac{1-BP}{BP})})$

And: $BP = \frac{\#ctg - \#chr}{2*\#ctg*(\#ctg-1)}$

$\#ctg$: number of contigs

$\#chr$: expected number of chromosomes

F_{adj} is defined as follow. We make the assumption that a genome's organisation is linear: a gene can have at most two neighbors.

If the extremity of the adjacency a (respectively b) already has two neighbors (ie. is in the middle of a contig), then this adjacency is not possible: $F_{adj} = 0$ (leading to $c_1(a,b) = \infty$)

If both extremities of the adjacency already have one neighbor, $F_{adj} = 1$.

If the extremity of the adjacency a (respectively b) already has one neighbor and the extremity b (resp. a) has no neighbors, then $F_{adj} = 2$, to account for the fact that a could be either one of the two neighbors of b.

If both extremities of the adjacency have no neighbors, $F_{adj} = 4$, to account for the different senses in which the two genes a and b could be linked together.

In the special case where DeCoSTAR is used with oriented adjacencies, a and b aren't genes, but extremities of genes (ie. a gene start or stop). We consider that gene extremity is always linked to the other extremity of the same gene. As a consequence, F_{adj} is always computed considering that a and b have one more neighbor than before.

SPI, or Scaffolding Propagation Index, is a parameter that accounts for the ditribution of poorly assembled genomes along the species tree. More precisely, it is the size of the clade c where

DeCoSTAR can still infer new adjacencies even though the adjacency has no extant homologues in c (ie. the only extant adjacency that support the new one is on the outgroup of this clade).

Data-sets

18 Anopheles genomes

The species tree is taken from (Fontaine *et al.*, 2015) , and constructed from a concatenate of genes on the X chromosome. Gene families have been obtained from (Neafsey *et al.*, 2015). There are 17 780 gene families in the database, and we discarded gene families with only 1 gene, families containing a gene included in another gene, and a few families which did not pass an alignment quality filter (no resulting site after GBlock). This resulted in 14 940 families. Gene trees were first inferred by RaxML with Muscle alignments as input, then corrected with ProfileNJ, keeping only 100% bootstrap support branches. Adjacencies were then computed as the set of consecutive genes on the same scaffold. DeCoSTAR was used with scaffolding mode and Boltzmann sampling with a temperature of 0.05.

References

Anselmetti, Y., *et al.* (2015). Ancestral gene synteny reconstruction improves extant species scaffolding. *BMC Genomics*, **16**(Suppl 10), S11.

Bérard, S., *et al.* (2012). Evolution of gene neighborhoods within reconciled phylogenies. *Bioinformatics (Oxford, England)*, **28**(18), i382–i388.

Chauve, C., Ponty, Y., and Zanetti, J. P. P. (2015). Evolution of genes neighborhood within reconciled phylogenies: an ensemble approach. *Lecture Notes in Computer Science (Advances in Bioinformatics and Computational Biology)*, **8826 LNBI**, 49–56.

Fontaine, M., *et al.* (2015). introgression in a malaria vector species complex revealed by phylogenomics. *Science*, **347**.

Neafsey, D. E., *et al.* (2015). Highly evolvable malaria vectors: The genomes of 16 anopheles mosquitoes. *Science*, **347**(6217), 1258522.

Patterson, M., Szöllősi, G., Daubin, V., and Tannier, E. (2013). Lateral gene transfer, rearrangement, reconciliation. *BMC Bioinformatics*, **14**(Suppl 15), S4.

Szöllosi, G. J., Tannier, E., Lartillot, N., and Daubin, V. (2013). Lateral gene transfer from the dead. *Systematic Biology*, **62**(3), 386–397.

DeCoSTAR's Documentation

DeCoSTAR: Detection of Co-evolution version 1, 24/11/2016

Given a species tree S, a set of gene family (unrooted) tree distributions G, a set of extant adjacencies A and a set of costs for adjacencies events (adjacency gain and adjacency breakage), DeCoSTAR can compute:

- reconciled gene trees R from the gene trees in G, such that these forms a most parsimonious reconciliation between S and G according to the TERA algorithm described in [1].

- an history of the given adjacencies in A along the reconciled gene trees R such that this history minimizes an adjacency gains and breakages cost(with respect to their relative costs); according to the models described in [2][3].

The history of adjacencies comes in the form of one or several adjacency trees (phylogenetic trees in which nodes represents adjacencies between the nodes of gene trees). We refer to such an history as an adjacency forest. Note that DeCoSTAR can, but usually does not output adjacency forest but instead provide lists of inferred adjacencies in ancestral species.

Rather than an adjacency forest minimizing the number of adjacency gains and breakages, DeCoSTAR can sample adjacency forests in such a manner that adjacency forests with a lower adjacency cost have a higher chance to be sampled. This is done according to the algorithm described in [4], extended to include transfers.

It is also possible to use DeCoSTAR to infer adjacencies in extant species as described in [5] by using the scaffolding mode.

There are three required parameters. The first is the species tree file (`species.file`), which contains a tree in newick format. The second is a file containing the names (one per line) of files containing gene tree distributions with one distribution per gene family (`gene.distribution.file`). Even if each gene tree distribution only contains one tree (for instance if you used a maximum likelihood approach to get that tree), DeCoSTAR ask for one file per gene family. The third is a file containing the extant adjacencies (`adjacencies.file`). There must be one adjacency per line. An adjacency correspond to a couple of leaves names separated by a space.

Description of the software parameters

The parameters that are boolean (ie. whose default is true or false) should be specified using 0 or 1 (for, respectively, false and true). Parameters must be given using the format `<name>=<value>` (note the absence of `’-’` before the option name as well as the absence of space after the `’=’`. For instance, a typical command line might look like:

```
DeCoSTAR species.file=my.sp.tree.txt gene.distribution.file=my/distribs.txt adjacencies.file=adjs.tx
```

For a non dated (but with transfer) analysis where the cost of a single adjacency gain is 3 (the other events will have their default costs).

Input parameters:

- `species.file` required. Species tree file (newick). NB: default behavior wants it to be ultrametric if transfers are used (use `dated.species.tree=0` to circumvent)
- `parameter.file` no default value. A file with input parameters (one per line)
- `gene.distribution.file` required. Gene distribution files file (one file name per line)
- `adjacencies.file` required. Adjacencies file (one adjacency per line; leaf names separated by a space)
- `dated.species.tree` default : true. The species tree is ultrametric and dates will be used to subdivide the trees in time slices.

- `char.sep` default : `'_'`. Character separating gene names in gene tree files. One character only. Caveat : I you decide to use `char.sep='|'`, be aware that the character `'|'` is also used as a separator in the newick output of reconciled trees (if you use `write.newick=1`).
- `ale` default : false. Gene tree distribution are ALE files
- `already.reconciled` default : false. Gene tree distribution are reconciled gene trees in recPhyloXML format. Will skip the reconciliation phase
- `rooted` default : false. Specify that the root of the given gene trees must be kept. This option turns off amalgamation when switched on.

Reconciliation parameters:

In its default form, DeCoSTAR performs the reconciliation of gene families using the TERA algorithm [1]. As such, it include some of the algorithm's options.

The options that concern reconciliation are:

- `with.transfer` default: true. Allows transfers in the reconciliation and adjacency histories reconstruction.
- `dupli.cost` default: 2. cost of a single gene duplication
- `HGT.cost` default: 3. cost of a single Horizontal Gene Transfer
- `loss.cost` default: 1. cost of a single gene loss
- `try.all.amalgamation` default: true. try all possible amalgamation when reconciling gene trees. Otherwise only the best tree (ie. The most frequent) of the distribution is used.
- `Topology.weight` default: 1. In the case of amalgamation, this is the weight associated to the topology part of the score guiding the reconciliation

Not all options are included. In the case were some specific, non-included, options are required, it is recommended to perform the reconciliation independently using ecceTERA (or any other reconciliation software) and then to directly give the reconciliation to DeCoSTAR (using `already.reconciled=1`).

Basic adjacency history parameters:

- `Again.cost` default: 2. Cost of a single adjacency gain
- `Abreak.cost` default: 1. Cost of a single adjacency breakage

Adjacency history sampling parameters:

- `use.boltzmann` default: false. Use Boltzmann sampling for the adjacencies history computation.
- `boltzmann.temperature` default: 0.1. Temperature to use in the Boltzmann sampling (if used)
- `nb.sample` default: 1. Number of samples to get from the adjacency matrix. NB: it can be used together with `use.boltzmann` or not.

Adjacency history assembly related parameters (scaffolding mode [5]):

- `scaffolding.mode` default : false. Use scaffolding algorithm to improve extant genomes scaffolding/assembly.

- `chromosome.file` no default value. A file containing the number of chromosome in each each species (one species per line, each line comprised name of the species followed by the number of chromosome, separated by a tabulation)

- `adjacency.score.log.base` default :10000. Used in the case where the adjacency file also contains a score between 0 and 1. Base of the logarithm applied to this score.

- `scaffold.includes.scored.adjs` default : false. Used in the case where the adjacency file also contains a score between 0 and 1 AND `scaffolding.mode` is true. If true, include the adjacencies with a score ¡ 1 in the computation of the number of contigs.

Advanced adjacency history parameters:

- `C1.Advantage` default: 0.5. Between 0 and 1. Probability to choose C1 (presence of adjacency) over C0 (absence of adjacency) in case of a score tie at the root of an equivalence class

- `all.pair.equivalence.class` default: false. Compute adjacency histories for all pair of gene families (even if they share no adjacencies).

- `bounded.TS` default: false. Use bounded time slices in adjacency history computations (only if the species tree is dated)

- `always.Again` default: true. Always put an adjacency Gain at the top of an equivalence class tree

- `absence.penalty` default: -1. If set to -1 (the default), nothing changes. Otherwise, specify the cost of having an adjacency at a pair of leaves which are not part of the list of adjacencies given at initialization

- `substract.reco.to.adj` default: false. If set to 1, the weighted cost of a reconciliation event will be used to favor co-event in the adjacency matrix computation. Unavailable for Boltzmann computation.

- `Reconciliation.weight` default: 1. Weight of the reconciliation events when `substract.reco.to.adj=1`

- `Adjacency.weight` default: 1. Weight of the adjacency events when `substract.reco.to.adj=1`

Output parameters:

- `verbose` default: 1. Show progress and timing.

  0: nothing is reported short of error.
  1: basic report (default).
  2: various information about reconciliation, adjacency matrix and backtracking
  3: maximal amount of information

- `write.newick` default: false. Use newick format rather than phyloXML-like format for ouputs.
- `hide.losses.newick` default: false. If true, losses and the branches leading to them will be removed from newick formatted output.
- `write.adjacencies` default: true. Write the adjacencies inferred in ancestral species in a file.
- `write.genes` default: false. Write the genes inferred in ancestral and extant species.
- `output.dir` default: none. Directory to print files in.
- `output.prefix` default: none. A prefix to prepend to all output files.
- `write.adjacency.trees` default: false. Write the inferred adjacency trees.

Input formats:

The default parameters assume an ultrametric dated (i.e., with branch lengths), binary species tree. An undated species tree is input using `dated.species.tree=0`.

The gene trees are expected to be unrooted and in a newick format (unless the rooted option is used). Leaves names should be composed of the name the the species in which the leaf is and the gene name, linked by a separator (by default, this separator is `'_'` and can be changed using `char.sep`). Rather than gene trees, the user may supply ale files instead (specified with `ale=1`), which are files that sum up a gene distribution in the form of conditional clades probabilities. Such files can be obtained with the ale [6] software. If the gene family ave already been reconciled (for instance if you don't want to use the same method as DeCoSTAR), they can be provided instead of the gene tree distributions (with `already.reconciled=1`) . Reconciled trees should be provided using the recPhyloXML format (see http://phylariane.univ-lyon1.fr/recxml/ for a description of the format).

The adjacencies given in the file (`adjacencies.file`) may present two additional fields describing the orientation of the genes forming the adjacencies. These orientations are specified using the `'+'` and `'-'` character (respectively for a sense and anti-sense gene). This will cause DeCoSTAR to treat the extremities of a gene as two different entities when it comes to adjacencies (but not reconciliation : the two extremities of a same gene have the same history).

Additionally, the adjacencies given in the file (adjacencies.file) may have a third field that should be a number between 0 and 1. This number will be used as a score denoting the confidence that the adjacency really exists (1 meaning that the adjacency is certain; 0 that the adjacency is not possible) that DeCoSTAR algorithm will take into account, allowing the possibility to create adjacency histories without this adjacency. This is an advanced functionality, and it is linked to the option (`adjacency.score.log.base`) which determine the base of the logarithm that is used to go from this 0-to-1 score to a parsimony cost.

If both orientation and score are specified for an adjacency, they should come in that order: orientation, then score; such that a valid line could look like:

```
g1 g2 + + 0.9
```

Rather than being all provided in the command lines, arguments can be given in a file specified with the parameter.file argument. In that file, parameters can be given using the format

`<name>=<value>`. Any parameters given on the command line will take precedence and the duplicated parameter will be ignored.

Output formats:

If the `output.dir` option has been used, all file will be written in the specified directory. Otherwise they are written in the current directory.

If `write.adjacencies` is set to true (it is by default), DeCoSTAR will output the a file containing the adjacencies inferred at ancestral speciation nodes and leaves such that each line represent an adjacency. The fields of these lines are separated by spaces and correspond to, in order:

- the species the adjacency is in
- the gene forming the first extremity if the adjacency
- the gene forming the second extremity if the adjacency
- the orientation of the first extremity if the adjacency (as described in the input format section)
- the orientation of the second extremity if the adjacency (as described in the input format section)
- the eventual score given to that adjacency at input (NB: ancestral adjacency have an input score of 0)
- the frequency of observation of the adjacency (ie. how many time the adjacency was observed across all sample divided by the number of samples)

If `write.genes` is set to true (it is set to false by default), DeCoSTAR will output a file describing all extant and ancestral genes. Each line correspond to one gene and begins with the code of the species the gene is in followed by the gene name followed by the list of the gene's extant descendants (all separated by spaces).

DeCoSTAR will also output the species tree and will create a `reconciliations.suffix` file. If `write.adjacencies.trees` is set to true (it is set to false by default), then an `adjacencyTrees.suffix` file will also be written. Here `suffix` is either `'xml'` or `'newick'` depending on the chosen output format. These files contain respectively the reconciled tree of each gene family and the adjacency trees computed from the extant adjacencies.

By default, all trees are written in XML format. The species tree follows a classical phyloXML format. The reconciled gene tree are in the recPhyloXML format (see http://phylariane.univ-lyon1.fr/recxml/ for a detailed description of the format).

Adjacency trees follow a format close to the recPhyloXML one, adapted to include adjacency related events. As each node represent an adjacency, its given name is actually the name (or the id, if they have no name) of the two genes it links, separated by `'-'`. Each clade has a `<eventsAdj>` tag that contains an ordered list of event the adjacency has undergone. Each event has a coevent property. If the `coevent` property is set at `\1"`, then this indicate that the event spanned both end of the adjacency at the same time. If it is set at `\0"`, then it means that the event only spanned one end of the adjacency. There is an additional event when compared to recPhyloXML: the `adjBreak` event which marks an adjacency breakage. Furthermore, the different adjacency equivalence class families (ie. a group of adjacencies linking gene from the same couple of gene families) are grouped together under the `<EquivalenceClassFamily>` tag which specifies which gene families are linked. If several samples were done, then an additional `<sample>` tag is present.

If the option `write.newick` has been activated, reconciled gene trees and adjacency trees will be written in a newick format where reconciliation or adjacency information will be written in place of the bootstrap.

This information consists in the name of the gene (or its id in the case of an internal node), or the name of the two genes it links for adjacency trees, followed by the event associated with the node, the species it is in and the time slice it occurs at (if applicable). These fours fields are separated by a `'|'` character. For reconciliation trees, events may be:

- `Extant` : for leaves
- `Spe` : speciation
- `Loss` : gene Loss
- `Dup` : gene duplication
- `SpeOut` : speciation to an extinct/unsampled lineage (otherwise called SpeciationOut)
- `Reception` : transfer reception
- `Null` : no event (to account for time slices)
- `BifOut` : bifurcation in an extinct/unsampled lineage (otherwise called BifurcationOut)

For adjacency trees, the following events are added:

- any reconciliation event might have the prefix `\co-"` marking the fact that both extremities of the adjacency underwent the same event at the same time (for instance, a `co-duplication` means that the two adjacent genes were duplicated together)
- `Breakage` : adjacency breakage

The trees linking different gene families are separated by an information line beginning by `'>'` and specifying which families are linked and, if necessary, the sample the trees belong to.

NB: in this model of reconciliation, lateral gene transfer are modeled as a process where a gene first undergoes a speciation to an extinct or unsampled lineage of the species tree (otherwise called `SpeciationOut`) where it evolves for a certain time before being transferred from this unsampled lineage to a sampled lineage (ie. a branch) of the species tree (transfer reception). See [1] or [5] for a more detailed view of this process when applied to reconciliation inference.

NB2: adjacency trees do not explicitly contain any adjacency gains because any adjacency gains actually gives rise to a new adjacency tree. In other words, there is an implicit adjacency gain at the root of every adjacency tree.

REFERENCES

1. Celine Scornavacca, Edwin Jacox, and Gergely Szöllősi. Joint Amalgamation of Most Parsimonious Reconciled Gene Trees. Bioinformatics (2014): btu728.

2. Sèverine Bérard, Coralie Gallien, Bastien Boussau, Gergely J. Szöllősi, Vincent Daubin and Eric Tannier. Evolution of gene neighborhoods within reconciled phylogenies. Bioinformatics (Oxford, England) Vol. 28, No. 18 (2012) p. i382-i388

3. Murray Patterson, Gergely Szöllősi, Vincent Daubin, Eric Tannier. Lateral gene transfer, rearrangement, reconciliation. BMC Bioinformatics Vol. 14, Suppl. 15 (2013) S4

4. Cedric Chauve, Yann Ponty and João Paulo Pereira Zanetti. Evolution of genes neighborhood within reconciled phylogenies: an ensemble approach. BMC Bioinformatics Vol. 16 Suppl. 19 (2015) S6

5. Yoann Anselmetti, Vincent Berry, Cédric Chauve, Annie Chateau, Eric Tannier and Sèverine Bérard. Ancestral gene synteny reconstruction improves extant species scaffolding. BMC genomics Vol. 16, Suppl. 10 (2015) S11

6. Gergely Szöllősi, Wojciech Rosikiewicz, Bastien Boussau, Eric Tannier and Vincent Daubin. Efficient Exploration of the Space of Reconciled Gene Trees. Systematic biology Vol. 62, No. 6 (2013) pp. 901-912

B HyLiTE: accurate and flexible analysis of gene expression in hybrid and allopolyploid species

This section corresponds to the work I did during an internship in the year 2014 and resulted in a published article where I am the first author.

B.1 Article

Duchemin *et al. BMC Bioinformatics* (2015) 16:8
DOI 10.1186/s12859-014-0433-8

BMC Bioinformatics

SOFTWARE Open Access

HyLiTE: accurate and flexible analysis of gene expression in hybrid and allopolyploid species

Wandrille Duchemin[1,2], Pierre-Yves Dupont[1], Matthew A Campbell[1], Austen RD Ganley[3] and Murray P Cox[1*]

Abstract

Background: Forming a new species through the merger of two or more divergent parent species is increasingly seen as a key phenomenon in the evolution of many biological systems. However, little is known about how expression of parental gene copies (homeologs) responds following genome merger. High throughput RNA sequencing now makes this analysis technically feasible, but tools to determine homeolog expression are still in their infancy.

Results: Here we present HyLiTE – a single-step analysis to obtain tables of homeolog expression in a hybrid or allopolyploid and its parent species directly from raw mRNA sequence files. By implementing on-the-fly detection of diagnostic parental polymorphisms, HyLiTE can perform SNP calling and read classification simultaneously, thus allowing HyLiTE to be run as parallelized code. HyLiTE accommodates any number of parent species, multiple data sources (including genomic DNA reads to improve SNP detection), and implements a statistical framework optimized for genes with low to moderate expression.

Conclusions: HyLiTE is a flexible and easy-to-use program designed for bench biologists to explore patterns of gene expression following genome merger. HyLiTE offers practical advantages over manual methods and existing programs, has been designed to accommodate a wide range of genome merger systems, can identify SNPs that arose following genome merger, and offers accurate performance on non-model organisms.

Keywords: Hybrid, Allopolyploid, Homeolog, RNA-seq, Read assignment

Background

While evolution is usually a gradual process, the creation of a new species through the merger of different parent species occurs near instantaneously [1]. Although increasingly recognized as an important process in the evolution of many biological systems [2-5], how different gene copies (homeologs) are expressed following genome merger remains a major outstanding question [6,7]. Most studies have been restricted to observing just a few genes, thus limiting the ability to study interactions between competing gene regulation systems [8]. High throughput mRNA sequencing now permits whole-genome screening of hybrid and allopolyploid gene expression [9,10]. However, identifying the parental origin of mRNA reads remains challenging, especially for researchers without advanced bioinformatics skills [11].

To fill this gap, we have developed HyLiTE – *Hy*brid *Li*neage *T*ranscriptome *E*xplorer – to produce tables of homeolog expression data from raw mRNA read files in a single step. HyLiTE automatically i) maps reads to a reference genome, ii) masks gene regions with low read coverage, iii) identifies polymorphisms that are diagnostic of parental lineages, iv) classifies reads to parental types, and v) produces detailed summary reports of gene expression in both the hybrid or allopolyploid and its parent species. The final product – tables of homeolog read counts – can be used immediately for downstream analyses (such as determining differential expression between biological conditions, and between the new species and its parents).

Implementation

The primary design directives behind HyLiTE were i) ease of use, ii) runtime efficiency, and iii) use with non-model organisms (which encompasses most hybrid and allopolyploid species). Other key features include:

*Correspondence: m.p.cox@massey.ac.nz
[1] Statistics and Bioinformatics Group, Institute of Fundamental Sciences, Massey University, Palmerston North, New Zealand
Full list of author information is available at the end of the article

- Accommodating any number of parent species (for instance, three-parent allopolyploids such as modern hexaploid wheat) [12].
- The ability to study systems with both haploid or diploid parents, thus allowing hybrids or allopolyploids with different homeolog and allelic copies.
- Using gene references from any species closely related to the study system (hybrid and allopolyploid species often lack good genome resources).
- Accommodating any number of biological replicates (and boosting SNP identification by combining information across replicates).
- Identifying new polymorphisms that have arisen within the hybrid or allopolyploid (especially important in species derived from older merger events).
- Improving SNP calling by using (optional) genomic DNA information in addition to high throughput mRNA sequences.
- Providing statistical validation of SNP calls and automatically masking 'polymorphisms' with low statistical support.
- An experimental feature that identifies putative chimeric genes (i.e., genes in which the homeologs have recombined within the hybrid or allopolyploid) [13], but see Additional file 1 for details on current limits of accuracy.

The standard HyLiTE analysis, which will be adequate for most users, comprises a single, short command line. However, advanced users have complete flexibility to override individual steps. For instance, by default, Bowtie2 is used for read mapping, but HyLiTE can be run with any mapping software that returns the standard SAM mapping format.

Because HyLiTE analyzes each gene independently, the software has low RAM requirements and runtime is linear with the number of genes under study. This independence between genes also allows HyLiTE to be parallelized via optional executables (see project website for details; http://hylite.sourceforge.net). HyLiTE regularly autosaves the run state, and analyses can therefore be stopped and re-started from the last checkpoint. Extensive documentation about the algorithms implemented in HyLiTE, software validation and benchmarking against alternative pipelines is provided in Additional file 1.

Results and discussion

Output

The main output of HyLiTE comprises a list of read counts for each homeolog in each biological replicate. Using presence and absence of diagnostic parental SNPs, reads are classified as i) derived from a given parent, ii) consistent with two or more parents (i.e., lacking diagnostic SNPs), or iii) unknown (i.e., masked due to low read coverage). The last two classes are equally uninformative for determining homeolog expression, but can distinguish whether improvements may be possible with additional sequence data (the 'unknown' category) or whether part of the gene is simply uninformative for ancestry (no diagnostic parental SNPs identified). Finally, each read is marked with an additional flag if one or more new SNPs are detected within the hybrid or allopolyploid.

Software comparison

A major point of difference between HyLiTE and alternative approaches (e.g., PolyCat [14]) is its robust statistical assessment of SNP calls and automatic masking of 'polymorphisms' with low statistical support. Due to the substantial error rate of high throughput sequencing technologies, sequencing errors can easily be confused with genuine polymorphisms in genes with low expression (and hence, low read coverage). The probability that a polymorphism at any given nucleotide position is a SNP rather than an error is given by a binomial distribution conditioned on the coverage level. Nucleotides with coverage less than this threshold are masked, but because coverage varies widely across even a single gene, typically only small, uninformative regions of any given gene are masked. This 'dynamic masking' substantially improves the accuracy with which reads are assigned to homeologs for genes with low to moderate expression. Detection of expression levels can be improved further by including genomic DNA reads due to the accuracy this imparts to SNP calling (see Additional file 1 for details).

Worked examples

Fungi. Species in the fungal genera *Epichloë* and *Neotyphodium*, already well known for their symbiotic relationships with grasses in temperate pastoral systems, are increasingly becoming the dominant model system for studying genome merger in fungi [9,15,16]. The most well studied example is Lp1, an economically important allodiploid asexual species formed from the union of a haploid sexual species, *E. typhina*, and a haploid asexual species, *N. lolii* (∼5% divergence). As HyLiTE had not yet been developed, the Cox *et al.* study instead applied a two-reference approach: gene references were generated separately for *E. typhina* and *N. lolii* using ancestry informative SNPs, and homeolog expression was then ascertained via high stringency mapping. Although estimates of gene expression are highly correlated ($r = 0.83, P \ll 0.0001$), HyLiTE assigns an average of five times as many reads to homeologs as the two-reference approach, an improvement almost entirely due to reduced gene masking (Figure 1A). 86% of reads are assigned to homeologs, with the remainder classified as parental uninformative or unknown. PolyCat [14] assigned fewer reads to homeologs (Figure 1B), particularly for genes with low to moderate expression (see Additional file 1 for details).

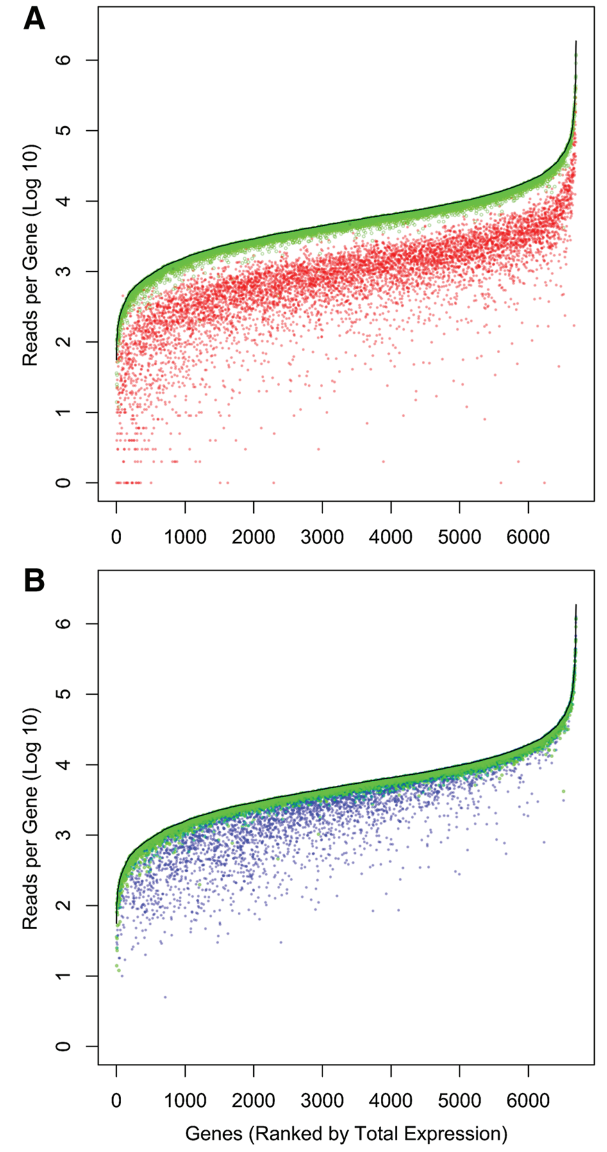


Figure 1 Comparison between HyLiTE and A) the results of the Cox *et al.* study [9] and B) PolyCat [14] for *Epichloë* fungal data. The black lines indicate the total number of reads that map to each gene, ranked by expression level. Green points indicate the number of reads assigned to homeologs by HyLiTE. Red points in **A)** indicate the number of reads assigned to homeologs in the Cox *et al.* study, while blue points in **B)** indicate the number of reads assigned to homeologs by PolyCat. The substantial improvement in read assignment by HyLiTE was obtained using its default settings.

Plants. To show application to a plant system, we also analyzed gene expression in a natural cotton allotetraploid, *Gossypium hirsutum*, together with diploid representatives of the A (*G. arboreum*) and D (*G. raimondii*) genomes (∼3% divergence) [10]. Assignment accuracy was tested by classifying known reads from the two diploid species. HyLiTE assigned reads to homeologs with a very low error rate (1.6%; see Additional file 1 for details). It also identified 46,206 new SNPs specific to *G. hirsutum*.

Animals. Finally, we analyzed gene expression in a synthetic allotetraploid fish derived from diploid goldfish (*Carassius auratus*) and diploid common carp (*Cyprinus carpio*) (∼6% divergence) (NCBI BioProject accession number: PRJNA82763). The very small number of reads available per gene (an average of only 15) caused HyLiTE to reject most SNP calls and therefore classify the majority of reads as parentally uninformative. However, the reads for which sufficient information was available to assign parental ancestry showed a very low error rate (0.22%).

Conclusions

The formation of a new species from the merger of two or more different parent species is important in the evolutionary history of many eukaryotic lineages. Hybrid and allopolyploid species carry multiple copies of each gene (homeologs), and while homeolog expression levels can be determined from high throughput RNA sequence data, assigning reads is extremely challenging. Here, we have developed HyLiTE to automate the process of moving from raw mRNA sequence files to tables of homeolog expression in a hybrid or allopolyploid and its parent species. This single-step analysis is specifically designed for ease-of-use, particularly for non-computational scientists. HyLiTE therefore allows gene expression patterns to be explored on a whole-genome scale even for species with very complex patterns of genome merger.

Availability and requirements

Project name: HyLiTE
Project home page: http://hylite.sourceforge.net
Operating systems: Linux, OS X, Windows
Programming language: Python
Other requirements: None
License: GNU GPL v. 3.0
Any restrictions to use by non academics: None

Additional file

Additional file 1: Algorithms, validation and benchmarking. Documentation of algorithms, software validation and benchmarking against alternative pipelines.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

WD designed and developed HyLiTE, and drafted the manuscript. MPC, PYD and ARDG contributed to software design and analyses, and drafted the manuscript. MAC contributed to analyses, and drafted the manuscript. All authors read and approved the final manuscript.

Acknowledgements

Research support was provided to MPC by the Royal Society of New Zealand via a Rutherford Fellowship (RDF-10-MAU-001) and by the BioProtection Research Center, a New Zealand Center of Research Excellence (CoRE), via a Principal Investigator award. These funding bodies played no role in study design; collection, analysis or interpretation of data; writing of the manuscript; or the decision to submit this manuscript for publication.

Author details

[1] Statistics and Bioinformatics Group, Institute of Fundamental Sciences, Massey University, Palmerston North, New Zealand. [2] Present address: Laboratoire de Biométrie et Biologie Évolutive, UMR CNRS 5558, Université Lyon I, F-69622 Villeurbanne, France. [3] Institute of Mathematical and Natural Sciences, Massey University, Auckland, New Zealand.

Received: 27 June 2014 Accepted: 16 December 2014
Published online: 16 January 2015

References

1. Wendel JF. Genome evolution in polyploids. Plant Mol Biol 2000;42: 225–249.
2. Dehal P, Boore JL. Two rounds of whole genome duplication in the ancestral vertebrate. Plos Biol 2005;3:1700–1708.
3. Sémon M, Wolfe KH. Preferential subfunctionalization of slow-evolving genes after allopolyploidization in *Xenopus laevis*. Proc Natl Acad Sci USA 2008;105:8333–8338.
4. Soltis PS, Soltis DE. The role of hybridization in plant speciation. Ann Rev Plant Biol 2009;60:561–588.
5. Jiao Y, Wickett NJ, Ayyampalayam S, Chanderbali AS, Landherr L, Ralph PE, Tomsho LP, Hu Y, Liang H, Soltis PS, Soltis DE, Clifton SW, Schlarbaum SE, Schuster SC, Ma H, Leebens-Mack J, dePamphilis CW. Ancestral polyploidy in seed plants and angiosperms. Nature 2011;473: 97–100.
6. Doyle JJ, Flagel LE, Paterson AH, Rapp RA, Soltis DE, Soltis PS, Wendel JF. Evolutionary genetics of genome merger and doubling in plants. Annu Rev Genet 2008;42:443–461.
7. Sémon M, Wolfe KH. Consequences of genome duplication. Curr Opin Genet Dev 2007;17:505–512.
8. Adams KL, Wendel JF. Novel patterns of gene expression in polyploid plants. Trends Genet 2005;21:539–543.
9. Cox MP, Dong T, Shen G, Dalvi Y, Scott DB, Ganley ARD. An interspecific fungal hybrid reveals cross-kingdom rules for allopolyploid gene expression patterns. PLoS Genet 1004;10:180.
10. Yoo M-J, Szadkowski E, Wendel JF. Homoeolog expression bias and expression level dominance in allopolyploid cotton. Heredity 2013;110: 171–180.
11. Buggs RJA, Renny-Byfield S, Chester M, Jordon-Thaden IE, Viccini LF, Chamala S, Leitch AR, Schnable PS, Barbazuk WB, Soltis PS, Soltis DE. Next-generation sequencing and genome evolution in allopolyploids. Am J Bot 2012;99:372–382.
12. Brenchley R, Spannagl M, Pfeifer M, Barker GLA, D'Amore R, Allen AM, McKenzie N, Kramer M, Kerhornou A, Bolser D, Kay S, Waite D, Trick M, Bancroft I, Gu Y, Huo N, Luo M-C, Sehgal S, Gill B, Kianian S, Anderson O, Kersey P, Dvorak J, McCombie WR, Hall A, Mayer KFX, Edwards KJ, Bevan MW, Hall N. Analysis of the bread wheat genome using whole-genome shotgun sequencing. Nature 2012;491:705–710.
13. Gaeta RT, Pires JC. Homoeologous recombination in allopolyploids: The polyploid ratchet. New Phytologist 2010;186:18–28.
14. Page JT, Gingle AR, Udall JA. PolyCat: A resource for genome categorization of sequencing reads from allopolyploid organisms. G3 2013;3:517–525.
15. Moon CD, Craven KD, Leuchtmann A, Clement SL, Schardl CL. Prevalence of interspecific hybrids amongst asexual fungal endophytes of grasses. Mol Ecol 2004;13:1455–1467.
16. Schardl CL, Leuchtmann A, Tsai HF, Collett MA, Watt DM, Scott DB. Origin of a fungal symbiont of perennial ryegrass by interspecific hybridization of a mutualist with the ryegrass choke pathogen, *Epichloë typhina*. Genetics 1994;1307–1317.

B.2 Supplementary material

ADDITIONAL FILE 1

HyLiTE: Supplementary Materials

Wandrille Duchemin, Pierre-Yves Dupont, Matthew A Campbell, Austen RD Ganley and Murray P Cox

1 Algorithm for Detecting SNPs

Error model

A single nucleotide mismatch in a read relative to the reference can have two causes: either a genuine polymorphism, or an error introduced by the sequencing technology.

Consider a unique ε, the probability that an error is generated on a read by the sequencing technology at a given position. Every base in a read has probability $\frac{\varepsilon}{3}$ of being an error. (The denominator is three because, for any given position, one base is the reference, while the other three bases are tested independently as being either SNPs or errors).

The distribution of errors along reads is not uniform. However, for a given coordinate in a gene (with the exception of boundary conditions at gene extremities), the corresponding position in aligned reads does tend to be uniformly distributed. Further, it is common practice to trim bases from reads when quality scores drop below some low threshold, thus guaranteeing a minimum level of sequence quality for the entire read [1]. We favor the SolexaQA package to perform this trimming (`http://solexaqa.sourceforge.net`).

It follows that the occurrence of a specific incorrect base (*i.e.,* a base that does not represent the true genotype) at a given coordinate in the reference gene follows a binomial law with parameters

$p = \frac{\varepsilon}{3}$
n = number of reads (from the same organism, but across all genomic DNA and RNA samples) mapping at that position (*i.e.,* the local coverage)

If the number of reads carrying an observed mismatch is a statistical outlier under such a binomial law, the variant can be considered a SNP rather than an error. (Note that the reverse does not necessarily hold: a gene with low expression might produce so few reads that a true SNP cannot be distinguished from sequencing error. See section 4 on the benefits of using genomic DNA reads to improve expression estimates below).

Parameter Values

Using the SolexaQA package [1], the error rate ε of the Cox *et al.* dataset [2] was determined to be 0.02. (This value is typical of current Illumina sequence datasets; unpublished results). Therefore

$$p = \frac{\varepsilon}{3} = \frac{0.02}{3} \qquad (1)$$

The value n was determined as the number of reads at each coordinate along the gene (*i.e.,* the local coverage).

The statistical threshold α was determined in the following way. As testing for the presence of SNPs is performed at every coordinate in every gene, a very large number of independent statistical tests are necessarily run. Therefore, the standard statistical α (*e.g.,* 0.05 or 0.01) must be corrected for multiple testing. Because *a priori* we do not know the number of tests m that need to be performed, other than $m \gg 1$, we employ the Rough False Discovery Rate (RFDR) to correct our α in the limit $m \to \infty$

$$\alpha_{adjusted} = \frac{\alpha(m+1)}{2m} \qquad (2)$$

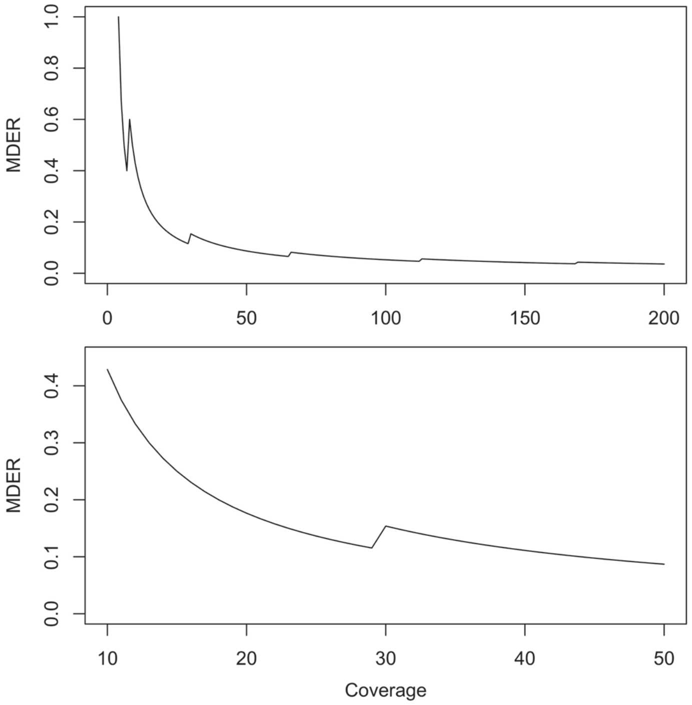


Figure S1 Decline in the Minimum Detected Expression Ratio (MDER) with increasing coverage. Upper panel: coverage from zero to 200; lower panel: enlargement of region showing coverage from 10 to 50.

and for large m

$$\alpha_{adjusted} \approx \frac{\alpha}{2} \tag{3}$$

It follows that a global probability threshold of 0.01 would require an adjusted $\alpha = 0.005$. However, we note that there are currently major concerns around statistical testing in scientific research, particularly around low P value thresholds. Therefore, following the guidelines of Johnson *et al.* [3], we advocate for a more conservative α of 0.001.

Minimum Coverage Rates

The local coverage rate n is a key parameter that allows us to distinguish between genuine SNPs and errors. As the binomial law is discrete, small values of n return probability estimates of whether a mismatch is a genuine SNP (or an error) in choppy, discrete steps (see Figure S1). Because the local coverage n changes along the gene, we have developed an algorithm that determines the lower limit of n for which the binomial law returns probabilities with poor reliability.

Consider, for example, local coverage of just two reads. The expected number of errors is effectively zero, which implies that a true error at that gene coordinate would be called as a SNP. This is obviously undesirable.

For haploid organisms, only one genotype is expected at any gene coordinate. Given binomial probabilities, coverage of three reads is sufficient to detect genuine SNPs with high reliability. If coverage is less than three reads, the gene coordinate is masked and is not used to classify reads to parental types.

For polyploid organisms, multiple genotypes are possible. Further, RNA-seq data does not guarantee equal representation of each allele (either by chance, or because the different alleles do not have the same expression). Therefore, we consider the ratio of expression between the least expressed allele and the sum of the remaining alleles. Consider, for example, that one allele in a parent is expressed ten times, while the other is expressed only once. The expression ratio would be $\frac{1}{10} = 0.1$. This lets us define the Minimum Detected Expression Ratio (MDER) as the limit below which a SNP on the least expressed allele could be detected as an error. So, for given α, p and n, the MDER corresponds to the expected occurrence of a specific error (ε_{exp}) divided by $n - \varepsilon_{exp}$

$$MDER = \frac{\varepsilon_{exp}}{n - \varepsilon_{exp}} \tag{4}$$

Figure S1 (upper panel) shows the improvement in MDER with increasing coverage. (The limit tends to p as $n \to \infty$). Note that the 'choppy' curve results from the discrete nature of the binomial law. In short, however, the higher the coverage, the better a large difference in expression level between parental alleles can be distinguished. Figure S1 (lower panel) shows an enlargement of the upper panel, with a focus on coverage levels that are currently in a more cost effective range.

At 20-fold coverage, the MDER is 0.18 (exactly $\frac{3}{17}$), which only improves (*i.e.,* declines) as the coverage increases. We propose this value as a minimum coverage level for polyploid species. Put differently, an MDER greater than or equal to $\frac{3}{17}$ (*i.e.,* a minor allele frequency greater than or equal to 0.15) will always allow SNPs to be detected with high reliability if the coverage level is greater than or equal to 20. Under default settings, gene coordinates with coverage lower than 20 reads are masked. Both of these thresholds (haploid and polyploid) can be changed by the user.

Implementation

With the statistical framework for SNP detection in place, describing the SNP calling algorithm implemented in HyLiTE is relatively straightforward. At a given gene coordinate for a given organism:

1 Temporarily aggregate all genomic DNA and RNA reads. Thus, the local coverage becomes the sum of local coverages across the different samples.
2 Count every genotype and compare the values obtained to their expected counts under the binomial distribution.
3 Select the k best genotypes (where k is the ploidy of the organism).
4 Consider each statistically validated genotype that differs from the reference as a true SNP.

Note that temporarily combining data across samples increases the total coverage available for an organism, decreases the MDER, and therefore reduces the number of gene coordinates that need to be masked. (See section 4 on the use of genomic DNA reads to improve RNA read classification).

For example, at a given gene coordinate with reference **A**, a hybrid or allodiploid might have two biological replicates with counts:

```
sample1: A:14 T:1 G:54 C:2
sample2: A:5  T:0 G:14 C:0
```

After temporarily aggregating the two replicates, the counts become:

```
        A:19 T:1 G:68 C:2
```

The total coverage is 90 reads, which greatly exceeds the minimum suggested coverage threshold for polyploids ($n = 20$). According to the binomial distribution at this coverage level, calling a SNP requires the same mismatch to be observed in at least five reads. Therefore, only two possible genotypes remain: **A** and **G**. (Correspondingly, the observed mismatches at **T** and **C** are called as sequencing errors). Because the organism is diploid in this example, up to two possible genotypes can be present at the same gene coordinate and both of the called genotypes are retained. As **A** is the reference state, we conclude that a SNP exists with the alternative state **G**, and that both **A** and **G** states are present.

Alternative Coverage Thresholds

We show above that the default coverage thresholds used by HyLiTE offer a good trade-off between sensitivity and reasonable coverage goals. However, users are encouraged to tune these thresholds depending on their biological questions and the quality of their sequence data. This can be easily managed by changing the two options '–min_coverage_haploid' and '–min_coverage_polyploid' in the HyLiTE command line.

Users can also change the desired α value and expected error rate by employing an alternative parameter file with the command flag '–alternative_params'. The alternative parameter file would look similar to this:

```
#SNP detection
MIN_COVERAGE_HAPLOID = 3      # alternative minimum coverage for haploid organisms
MIN_COVERAGE_POLYPLOID = 20   # alternative minimum coverage for polyploid organisms
EXPECTED_ERROR_RATE = 0.02/3  # alternative total error rate
ALPHA = 0.001                 # alternative alpha value
```

2 Algorithm for Classifying Reads

The primary purpose of HyLiTE is to determine the parental origin of high throughput RNA reads from a hybrid or allopolyploid. The following sections describe how this goal is achieved.

Fingerprints

We define the sequence of SNPs present, absent or masked (due to poor coverage) at specific coordinates along a gene as a 'fingerprint'. Specifically, we distinguish two types of fingerprint:

- Parent fingerprints: where information about the presence, absence and masking of SNPs are stored, gene-by-gene, for each parent.
- Child fingerprints: where every read in the hybrid or allopolyploid has its own fingerprint, referencing presence or absence of every SNP on that specific read.

Note that child fingerprints do not allow masking. As noted above, we encourage poor quality read segments to be trimmed from the dataset (*e.g.*, using the 'DynamicTrim' function of the SolexaQA package [1]).

Read Tagging for Diploid Parents

Diploid parents can have up to two fingerprints at each SNP position to allow for allelic heterozygosity. Accounting for this heterozygosity is achieved locally through read tagging (*viz.* linkage analysis).

Read tags have three possible values: unassigned, gene copy 0, and gene copy 1. Each parent is initiated with an unassigned tag. When a heterozygous position is detected, the algorithm first looks for existing reads with an assigned tag. (Note that HyLiTE steps through each gene from beginning to end, and therefore processes multiple reads in parallel). If no earlier read has an assigned tag (as occurs when starting a new gene), gene copy 0 or 1 is assigned arbitrarily to the allele, and each subsequent read carrying that SNP variant is given the same tag. However, if one or more reads have already been tagged (*i.e.,* they have already been assigned to an allele), HyLiTE determines which allele coincides with which tag and then propagates those earlier tags to all reads bearing the same genotype at the new gene coordinate.

Algorithm

HyLiTE detects SNPs sequentially along the gene for each organism simultaneously. This means that when HyLiTE moves to a new gene coordinate, all leftward SNPs have already been detected in both the parents and the hybrid or allopolyploid descendant. This is an important characteristic: as soon as a read from the hybrid or allopolyploid has been processed, all SNPs (present, absent and masked) have already been detected and referenced in every organism. Thus, HyLiTE can immediately classify that read.

After removing masked SNPs (*i.e.,* SNPs with poor coverage in at least one parent), the remaining SNPs can be denoted by 0s and 1s, where 0 indicates the absence of a SNP and 1 indicates its presence. The same process can be performed for SNPs in each parent.

Consequently, classification comes down to a series of comparisons between the parent and child fingerprints. For reasons of computational speed, these are treated as a list of boolean values and analyzed with bitwise operators. The process is:

1 Eliminate any child-specific SNPs, as these cannot help (but can hinder) comparison of the parent and child fingerprints. If any SNP is eliminated, set an ‘N’ flag showing that there is at least one ‘new’ non-ancestral SNP on the read.
2 For remaining SNPs, perform a bitwise XNOR operation between the child fingerprint and the fingerprint of each parent.
3 If this XNOR operation returns a list composed only of ones for any parent (*i.e.,* the fingerprints match perfectly), consider the read as coming from this parent.
4 If no perfect match is found, recursively try to ‘recombine’ parent fingerprints until a perfect match is found.

If multiple parents exhibit a perfect match after step 3, the read is classified as equally consistent with coming from more than one parent. For step 4, the ‘crossing’ operation is simply implemented as an OR operation between parent XNOR results. Scenarios that imply recombination between fewer parents are preferred.

Worked Examples

Simple Example

Consider a read r from an allodiploid with two haploid parents, P1 and P2. Let r span coordinates 35 to 124 on *gene1*. The coordinates of SNPs in this region, with absence/presence/masking information for the allodiploid read r and parents P1 and P2 is:

position	r	P1	P2
40	1	1	-1
52	1	0	0
65	1	1	1
96	1	0	1
113	1	1	1

Where ‘1’ signifies the presence of the SNP, ‘0’ signifies its absence, and ‘-1’ signifies a masked SNP (*i.e.,* coverage falls below the allowed threshold for that organism at that gene coordinate).

The first step eliminates the masked SNP at position 40. The fingerprints for each organism then appear as follows:

```
r:  1111
P1: 0101
P2: 0111
```

The SNP at position 52 is child-specific (*i.e.,* it is found only in the allodiploid, but neither of its parents). This SNP is removed and the new SNP 'N' flag raised. The fingerprints now appear as follows:

```
r:  111
P1: 101
P2: 111
```

An XNOR operation is performed between the fingerprint of the allodiploid read r and the fingerprints of each of the parents:

```
r XNOR P1: 101
r XNOR P2: 111
```

In this step, r XNOR P2 yields a result with all values 1, while r XNOR P1 yields a mixture of 0s and 1s. We therefore conclude that P2 is the likely origin of the allodiploid read r.

The category assigned to r is '(P2)+N', indicating an origin in parent P2, as well as the presence of a new child-specific SNP in the allodiploid. (Note that the use of parentheses seems redundant in this simple case, but quickly becomes crucial when dealing with multiple parents).

Complex Example

Consider a read r from an allotriploid with three parents, P1, P2 and P3. All preliminary steps, including removal of masked and child-specific SNPs leads to the fingerprints:

```
r:  1101
P1: 1011
P2: 1110
P3: 0001
```

The XNOR operations yield:

```
r XNOR P1: 1001
r XNOR P2: 1100
r XNOR P3: 0011
```

None of these results contains only 1s , so r is likely a chimeric read (*i.e.,* the result of recombination event(s) between two or more parental types). First, we test for biparental crossovers by performing a bitwise OR operation between pairs of results produced by the previous XNOR operation:

Category	Unknown	*E. typhina*-like	*N. lolii*-like	*E. typhina* or *N. lolii*	*E. typhina*/*N. lolii* chimeric reads
Manual	45	76	133	0	1
HyLiTE	22	73	130	1	29

Table S1 Comparison of manual read classification versus read classification by HyLiTE. Each column corresponds to a different read class, as defined by HyLiTE. *E. typhina* and *N. lolii* are the haploid parents of the allodiploid, Lp1. Note that due to the reduced accuracy with which chimeric reads are detected, these are reported by HyLiTE as 'parentally uninformative'.

```
P1 + P2: 1101
P1 + P3: 1011
P2 + P3: 1111
```

One result yields only 1s and therefore indicates that the allotriploid read r likely derives from recombination between the sequences of parents P2 and P3. Note that the resulting classification – (P2+P3) – actually implies the bitwise operation (r XNOR P2)OR(r XNOR P3). Note that due to poor accuracy rates (see validation sections), chimeric reads are reported as 'parentally uninformative'.

3 Validating Read Classification

To determine the accuracy with which HyLiTE assigns reads to parental types, we adopted a combination of simulation and manual validation approaches. Assignment accuracy was tested for hybrids and allopolyploids with both haploid and diploid parents.

Haploid Parents

We analyzed a dataset from the fungal allodiploid system described in the main text [2]. EfM3.000420 is a gene 1.1 kb in length with regions of both good and poor coverage, thus allowing us to assess the functionality of HyLiTE in both positive and negative conditions. Real RNA reads for the gene EfM3.000420 were extracted, classified automatically with HyLiTE, analyzed manually by visualizing the reads in the Integrative Genomics Viewer (IGV) [4] and then assigned to parental types one-by-one.

Both analyses identified regions where poor coverage in one of the parent species demanded masking. This typically occurs at the ends of genes, but can occur internally as well. EfM3.000420 was masked over bases 1–242, 630–648 and 854–1101 (where coordinate 1101 is the end of the gene). Low RNA read coverage in the parent species required nearly half of the gene to be masked. (Note that genomic DNA reads would circumvent this masking problem, as described in section 4 below). 35 SNPs were detected in the remaining unmasked regions. Both HyLiTE and the manual annotation identified exactly the same set of SNPs.

These 35 SNPs were used to classify reads to parental types, as quantified in Table S1. The manual classification very closely matches the classification made by HyLiTE. The main difference is that HyLiTE classified a number of reads as chimeric (*i.e.,* recombinant reads between the *E. typhina* and *N. lolii* parent sequences). When these reads were examined further, we identified the following mitigating conditions:

- 14 reads were located close to the masked region 630–648 and showed poor quality read alignment (*i.e.,* they were artifacts of Bowtie2 mapping errors, not HyLiTE).
- 12 reads were located near a SNP found only in the *E. typhina* parent, but not in the descendant allodiploid.

The few remaining misallocated reads were present in low complexity regions where indels and sequencing errors are common. While these types of error can be identified by eye, neither the mapping software Bowtie2 nor HyLiTE can make such subtle distinctions. Features of this nature explain all of the observed differences between the manual classification and the classification of HyLiTE.

We emphasize that manual classification of this single gene was a long process (approximately half a day's work) and mentally taxing. Although manual classification of reads outperforms the computational classification of HyLiTE, the results are remarkably similar for most read classes. Of course, manual classification is simply not possible for hundreds of millions to billions of reads.

We conclude that HyLiTE performs well under good quality read alignment conditions, as is usually the case for genes. Where alignment quality decreases, HyLiTE misclassifies some reads, and genes with large numbers of putatively chimeric reads are particularly prone to error. We also emphasize that the reference gene sequences must be chosen carefully because regions of poor quality alignment will increase as the reference sequence diverges from the transcriptomes under study.

Diploid Parents

To test the accuracy of read classification for tetraploid allopolyploids with diploid parents, the same simulated data was used as described below to study the effect of genomic DNA reads on RNA read classification. Mutations were inserted randomly. Because this is simulated data, both SNPs and read coverage tend to be relatively uniform along the gene sequence. For this reason, none of the 'mapping' problems identified in the real haploid dataset above were observed.

All simulated FASTQ read files were mapped to the gene sequence using the same parameters as HyLiTE, and the resulting mappings were displayed in IGV. All SNPs were identified correctly by HyLiTE. As the origin of all mutations was known, it was relatively straightforward to classify all reads manually. As all reads contained at least one diagnostic SNP (due to the 5% divergence rate for our system), the parent and allele from which every read was derived could be identified. For these simulated data, no discrepancies were found between the manual classification and the classification of HyLiTE. We note, however, the limitations of using simulated datasets, as they typically do not exhibit the complexity of real biological data (as illustrated above).

4 Using Genomic DNA Reads to Improve RNA Read Classification

Genomic DNA reads can be used to improve the classification of RNA reads by raising the call rate of diagnostic parental SNPs. To quantify this effect, we simulated DNA and RNA read data for two systems: i) haploid parents giving rise to an allodiploid, and ii) diploid parents giving rise to an allotetrapolyploid. For the haploid parent case, we simulated a 1700 bp gene with polymorphism rates of 4.5% (*i.e.,* 1 mutation per 22 bp) for the first parent and 1% (*i.e.,* 1 mutation per 100 bp) for the second parent, thus mimicking polymorphism rates observed in the fungal allodiploid system described in the main text. For the diploid parent case, we simulated a 1700 bp gene with the same polymorphism rates, but with 60% of mutations on one allele and 40% on the other. The allopolyploid was created by merging the parent gene copies, and adding new polymorphisms in the allopolyploid at a low rate (0.25%; *i.e.,* one mutation per 400 bp on average, thus corresponding to approximately one new polymorphism per gene copy). Genomic DNA and RNA reads were created by drawing 100 nucleotide sequences from these simulated genes with random start positions and strands. These simulated reads were written to different FASTQ files (one for each of the parents and the allopolyploid). As the purpose of this simulation was not to test the mapping efficiency of Bowtie2, quality scores were arbitrarily set to 'H', corresponding to a high Illumina 1.8+ Phred+33 value. The number of reads required for each coverage level were computed using:

$$N = \frac{LC}{l} \tag{5}$$

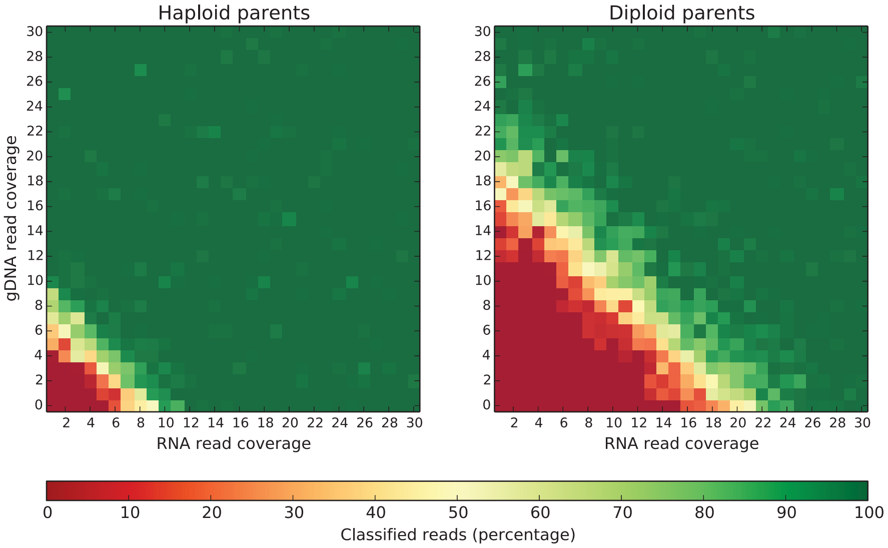


Figure S2 Improvement in RNA read classification when using genomic DNA reads. The percentage of classified reads is shown on a color scale from red (0%) to green (100%) for different genomic DNA and RNA read coverage levels.

where N is the number of reads, l is the length of the reads, L is the length of the gene and C is the desired coverage level. The percentage of reads classified by HyLiTE was determined for a range of genomic DNA and RNA coverage levels.

As expected, genomic DNA reads can improve expression estimates for genes with low to moderate expression (Figure S2). Because genes with low expression produce few reads, diagnostic parental SNPs often cannot be called, and hence, RNA reads cannot be assigned to one parent or the other. In such cases, genomic DNA reads can improve SNP calling, and thus lead to improved gene expression estimates. If no genomic DNA reads are available, expression estimates are poor for genes with less than ∼10-fold RNA coverage (allodiploids with haploid parents), or ∼25-fold RNA coverage (allotetraploids with diploid parents). Because RNA and DNA reads are interchangeable for SNP calling purposes, these limits also hold for genomic DNA reads: ∼10-fold DNA coverage (haploid parents) or ∼25-fold DNA coverage (diploid parents) is sufficient to assign nearly all RNA reads to a parental type regardless of the expression level of the gene. We emphasize that genomic DNA reads are, of course, not included in the read counts for expression estimates; they are only used for SNP calling purposes.

5 Comparing HyLiTE and PolyCat

The following comparisons employ the same datasets as in the "Worked Examples" section in the main text. In short, mRNA sequences were assigned to parental lineages (homeologs) using HyLiTE and PolyCat [5]. For comparability, the same mapping software was used with both programs, and where possible, runtime parameters were set to be as similar as possible. To illustrate its simplicity, HyLiTE was run with its default settings.

Fungi. HyLiTE results for the *Epichloë* dataset were obtained using a single command line:

```
HyLiTE -r ref.fasta -f protocol.txt -n PolyCatTest
```

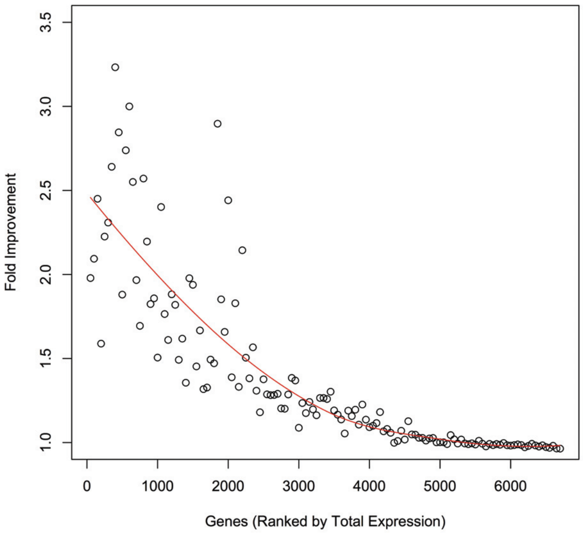


Figure S3 Median fold improvement of read assignment by HyLiTE compared to PolyCat relative to gene expression level for the *Epichloë* dataset. Fold improvement is plotted for sets of 50 genes ranked from low to high expression (points, left to right). The red line shows the local regression to the mean.

PolyCat results were obtained using the following pipeline:

1 Map reads to reference genes using Bowtie2 [6] separately for both of the parent species (*E. typhina* and *N. lolii*) and the allodiploid Lp1.
2 Create, sort and index *.bam files using SAMtools [7] separately for both the parent species and the allodiploid Lp1.
3 Run InterSNP (a companion program to PolyCat required to build the SNP index) separately on *.bam files from both the parent species.
4 Run PolyCat on the *.bam file from the allodiploid species Lp1.
5 Determine read counts from multiple output *.bam files using SAMtools and custom grep commands.

Steps 1, 2, 4 and 5 have analogs in HyLiTE, but are performed automatically. In addition, PolyCat requires a SNP index to be built for the parent species (step 3). Once built, this index can be used for multiple experiments. HyLiTE does not require a separate SNP index, and instead identifies this information gene-by-gene automatically.

The number of reads assigned to homeologs for each gene is shown in Figure 1 in the main text. HyLiTE determined homeolog expression for 6,693 of 6,694 genes in the reference (99.99%), compared to 6,638 genes in the Cox *et al.* study [2] (99.16%) and 5,995 genes for which homeolog expression was determined by PolyCat (89.56%). Although PolyCat determined homeolog expression for fewer genes than the two-reference mapping approach [2], homeolog assignment rates were substantially improved for those genes that were called (compare panels A and B in Figure 1 in the main text). Consequently, we suggest that specific software solutions (such as HyLiTE or PolyCat) should be strongly favored over alternative manual approaches.

In a direct comparison, HyLiTE assigned more reads to homeologs than PolyCat (the thin band of green points versus the cloud of blue points in Figure 1B in the main text). The number of reads assigned by HyLiTE approached the theoretical maximum (indicated by the black line) across the full range from low to high expression. Importantly, the rate at which HyLiTE assigned reads to homeologs has very low variance across this entire expression range (*i.e.,* the green 'band' has approximately equal width regardless of the expression level). Conversely, PolyCat was much more dependent on gene expression level (Figure S3). HyLiTE assigned a median of 2.5 times as many reads to homeologs than PolyCat for genes with very low expression (and in the case of some individual genes, as much as an order of magnitude more). However, HyLiTE and PolyCat assigned almost the same number of reads to homeologs for genes with high expression (with a slight bias in favor of PolyCat for highly expressed genes).

Plants. To show application to a plant system, we also analyzed gene expression in a natural cotton allotetraploid, *Gossypium hirsutum*, together with two diploid parent genome-type representatives, *G. arboreum* and *G. raimondii* [8]. HyLiTE and PolyCat [5] assigned reads to homeologs at comparable rates, with PolyCat assigning more reads (Figure S4). Assignment accuracy was tested by classifying known reads from the parent species:

$$error\ rate = \frac{misassigned}{misassigned + correctly\ assigned} \tag{6}$$

PolyCat exhibited a higher proportion of reads (8.0%) that were incorrectly assigned than HyLiTE (1.6%). This difference seems to reflect i) poor identification of chimeric reads (which we propose above mostly results from mapping errors) and ii) alternative choices in SNP calling strategies. Of 147,453 total SNPs, PolyCat treats 23,208 SNPs (16%) as having fixed differences between the parents (say, A versus G), and therefore uses these markers to classify the parental origin of reads. In contrast, HyLiTE recognizes these positions as polymorphic in at least one of the parent species (say, A+G versus A), and therefore masks the shared state (here, A) as uninformative for read classification. Manual screening on a subset of these SNPs confirmed that most are genuinely polymorphic. Consequently, while HyLiTE assigns fewer reads, it does so with greater accuracy.

We also note that the PolyCat software was validated on a dataset containing 1,140,550,335 reads for the first parent (*G. raimondii*) and 4,070,680,434 reads for the second parent (*G. arboreum*). In contrast, the analyses described here were performed on datasets that are two orders of magnitude smaller, and therefore directly comparable to most hybrid and allopolyploid studies.

Animals. Finally, we analyzed gene expression in a synthetic allotetraploid fish derived from diploid goldfish (*Carassius auratus*) and diploid common carp (*Cyprinus carpio*) (NCBI BioProject accession number: PRJNA82763). This dataset employed the 454 sequencing technology. As with the cotton example, PolyCat assigned more reads, but also showed a higher error rate (32%) than HyLiTE (0.22%), with most errors due to incorrectly called chimeric reads. The very small number of reads available per gene (an average of only 15) caused HyLiTE to reject most SNP calls and therefore classify the majority of reads as parentally uninformative. While HyLiTE consequently assigned many fewer reads to homeologs, the proportion of misassigned reads was nearly 150 times lower.

We note that PolyCat incorrectly assigned many reads as chimeric (295,538), although this feature appeared to validate well on their original cotton example [5]. Excluding chimeric reads, misalignments still result in an error rate ∼5 times greater than HyLiTE. Consequently, PolyCat, which was developed and validated on the model system cotton [5], appears to perform less well on non-model systems or alternative data types.

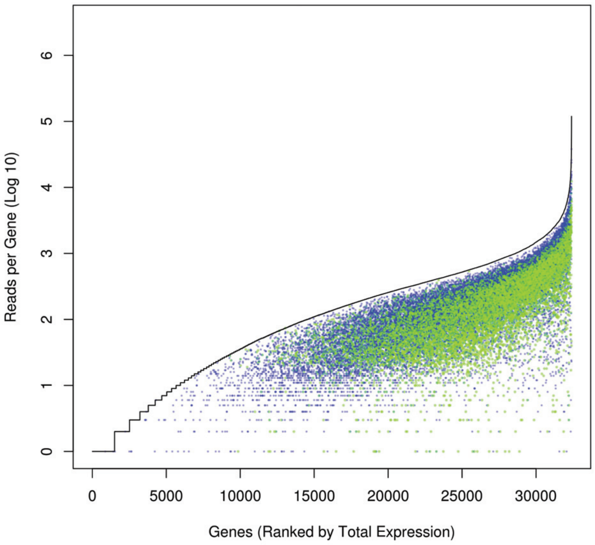


Figure S4 Comparison between HyLiTE and PolyCat for plant cotton data. The black line indicates the total number of reads that map to each gene, ranked by expression level. Green points indicate the number of reads assigned to homeologs by HyLiTE, while blue points indicate the number of reads assigned to homeologs by PolyCat. Default settings were used for both programs. Note that although PolyCat assigns more reads than HyLiTE, the number of misassigned reads is also substantially higher.

References

1. Cox, M., *et al.*: SolexaQA: At-a-glance quality assessment of Illumina second-generation sequencing data. BMC Bioinf **11**, 485 (2010)
2. Cox, M.P., *et al.*: An interspecific fungal hybrid reveals cross-kingdom rules for allopolyploid gene expression patterns. PLoS Genet **10**, 1004180 (2014)
3. Johnson, V.E.: Revised standards for statistical evidence. Proc Natl Acad Sci USA **110**, 19313–19317 (2013)
4. Thorvaldsdóttir, H., *et al.*: Integrative genomics viewer (IGV): High-performance genomics data visualization and exploration. Briefings Bioinf **14**, 178–192 (2013)
5. Page, J.T., *et al.*: PolyCat: A resource for genome categorization of sequencing reads from allopolyploid organisms. G3 **3**, 517–525 (2013)
6. Langmead, B., Salzberg, S.: Fast gapped-read alignment with Bowtie 2. Nat Methods **9**, 357–359 (2012)
7. Li, H., *et al.*: The sequence alignment/map format and SAMtools. Bioinformatics **25**, 2078–2079 (2009)
8. Yoo, M.-J., *et al.*: Homoeolog expression bias and expression level dominance in allopolyploid cotton. Heredity **110**, 171–80 (2013)

C Digital and Handcrafting Processes Applied to Sound-Studies of Archaeological Bone Flutes

This article was published in the conference proceedings of the conference on digital heritage EUROMED 2016, pp.184-195.

Digital and Handcrafting Processes Applied to Sound-Studies of Archaeological Bone Flutes

Etienne Safa[1], Jean-Baptiste Barreau[2,3], Ronan Gaugne[4],
Wandrille Duchemin[6], Jean-Daniel Talma[7], Bruno Arnaldi[3],
Georges Dumont[5], and Valérie Gouranton[3](✉)

[1] Université de Bourgogne/ArTeHiS UMR 6298, Dijon, France
[2] CNRS/CReAAH UMR 6566, Rennes, France
[3] INSA de Rennes/IRISA UMR 6074/Inria-Rennes, Rennes, France
Valerie.Gouranton@irisa.fr
[4] Université de Rennes 1/IRISA UMR 6074/Inria-Rennes, Rennes, France
[5] ENS de Rennes/IRISA UMR 6074/Inria-Rennes, Rennes, France
[6] LBBE UMR CNRS 5558, University of Lyon 1, Lyon, France
[7] Atelier El Bock, Chatel-Montagne, France

Abstract. Bone flutes make use of a naturally hollow raw-material. As nature does not produce duplicates, each bone has its own inner cavity, and thus its own sound-potential. This morphological variation implies acoustical specificities, thus making it impossible to handcraft a true and exact sound-replica in another bone. This phenomenon has been observed in a handcrafting context and has led us to conduct two series of experiments (the first-one using handcrafting process, the second-one using 3D process) in order to investigate its exact influence on acoustics as well as on sound-interpretation based on replicas. The comparison of the results has shed light upon epistemological and methodological issues that have yet to be fully understood.

This work contributes to assessing the application of digitization, 3D printing and handcrafting to flute-like sound instruments studied in the field of archaeomusicology.

Keywords: Acoustics · Statistics · Handcrafting · Raw-materials · Digitization · 3D printing · Music archaeology

1 Introduction

Elaborating a research project in close collaboration with a craftsman and a research team dedicated to digitization of cultural heritage was the trigger point to different kinds of experiments meant to investigate the morphological variability of bones and its influence on the emitted sounds when carved as flutes. Dealing with this "Sound-morphology" is the main part of a craftsman's work, which is why it was decided to run the project of an apprenticeship that would last for one year [18]. During this time, particular attention was paid to the creation and use of prototypes, i.e. a bone flute manufactured in order to try

M. Ioannides et al. (Eds.): EuroMed 2016, Part I, LNCS 10058, pp. 184–195, 2016.
DOI: 10.1007/978-3-319-48496-9_15

and understand the sound specificities of a particular bone, and then used as a guide in order to ease the adaptation process. Indeed, each bone has its own morphology and needs to be considered as an individual. The flute-maker proceeds then with a precise observation of each individual and takes every morphological specificity into consideration in order to craft series of bone flutes with similar sounds and identical tuning, even if this has to result in objects that do not look the same. Otherwise, he would risk to create an inefficient object, or a completely different flute.

These observations have raised specific issues regarding the use of bone flute's replicas for tone scales interpretations in archaeological surveys, as their manufacture never seems to take into consideration the bone's morphology as part of its acoustical specificities [6,8,16]. They have also led us to conduct "twin experiments" in the hope of reaching consistent results that would spare no methodological tracks (past, actual and yet-to-come sound-reconstruction methods) in order to explore their limitations as well as their potential. This way, we hope to contribute to better the epistemological landscape of archaeological flute's research.

The work presented in this paper focuses on the comparison of the sound results given by both series of experiments.

2 Context of the Work

2.1 Approach

Flutes are not all the same. They are grouped into several kinds which are distinguished by the way the air stream is directed toward the edge. Each kind has its own sound aesthetics, but gives also more or less freedom to the flute-player in choosing the pitch and the sound's characteristics, thanks to the blowing angle variability (Fig. 1). Oblique-, pan-, vessel- and transverse-flutes are amongst the most malleable kinds of flutes. We chose duct-flutes as they are the opposite.

In term of organology, these objects can be mentioned as 421.221.12 in the S/H classification system (Sachs/Hornbostel), which means: Internal duct-flute (straight and single) with finger holes and an open end.

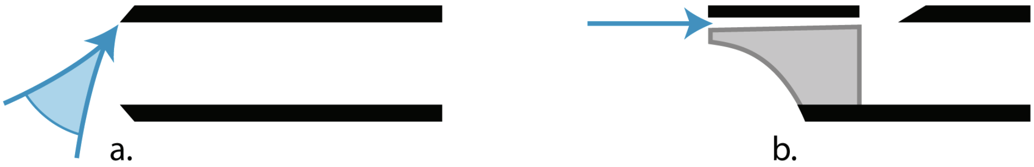


Fig. 1. Blowing angle variations regarding two different organological kinds of flutes: (a) oblique flute, (b) duct-flute

2.2 Partnership

This “two-front approach” demands to assemble a consistent amount of knowledge, which can only be achieved through partnership.

- **Handcrafting process:** the work gathered a traditional flute-maker and a statistician in computational biology.
- **3D process:** the work was based on an existing collaboration between archaeologists and computer scientists on advanced imaging for archaeology, the CNPAO [2]

2.3 Terminology

This paper will use the following terminology according to the acoustical specificities of bone flutes:

- **Morphology:** refers to the natural inner and outer shapes of the bone.
- **Geometry:** refers to the handcrafted inner and outer shapes carved deliberately or not onto the bone’s surface.
- **Sound-morphology:** refers to the acoustical sections of the morphology, which define the sound potential of the bone (i.e. the inner cavity). By definition, each bone has a different sound-morphology.
- **Sound-geometry:** refers to the acoustical sections of the geometry, which are involved in the definition of the instrument’s final sound, whether they were meant (deliberately carved) or not (unintentional and/or unconscious geometry). As an example: the shapes of the internal duct, of the edge, of the finger holes, etc. By definition, the sound-geometry rules out the outer shaping as long as it does not change the finger holes depth.
- **T0, T1, T2, etc.:** refers to the finger holes’ combination. T0 means all holes closed. T1 means that the lower finger hole (the first one) is open. T2 means that the two lower finger holes (the first and the second one) are open, etc.
- **F0, F1, F2, etc.:** systematic identification numbers of the experimental flutes. F0 refers to the control flute, whereas F1, F2, F3, etc. refers to each replica copying the control flute.

2.4 Related Works

Nowadays, 3D technologies allow outer and inner contact-free investigation on complex geometries [15]. As such they contribute to answer both preservation and sound studies issues and are more and more used in the actual archaeomusicological research. If their consequences on our interpretations are still to be defined, they allow different kind of approaches and studies that aim to get a better understanding of ancient sounds. They can be applied to any organological material [10], such as string instruments [4,13,20,21] but also aerophones [3,8,9,11], among which archaeological “flutes”, and objects presumed to be flutes, figure [1,14,22,23].

Eventually, the music-archaeology research may even explore new possibilities in sound reconstruction studies, as its data can be applied to sound simulators and sound-scape reconstructions [7,12,24].

3 The Sound-Morphology Principle

Naturally hollow rawmaterials, such as bones, hornes, shells or reeds, present a morphological variability between one individual and another. Those variations can be observed both regarding their shapes, their scale and their volumetric and spatial configuration (Fig. 2). Some of them are involved in the sound-morphology. For example, a larger bone will produce a lower pitch for the same length. Likewise, an important and sudden increase or decrease of the bone's conicity tends to distort the efficiency of a close-range finger-hole.

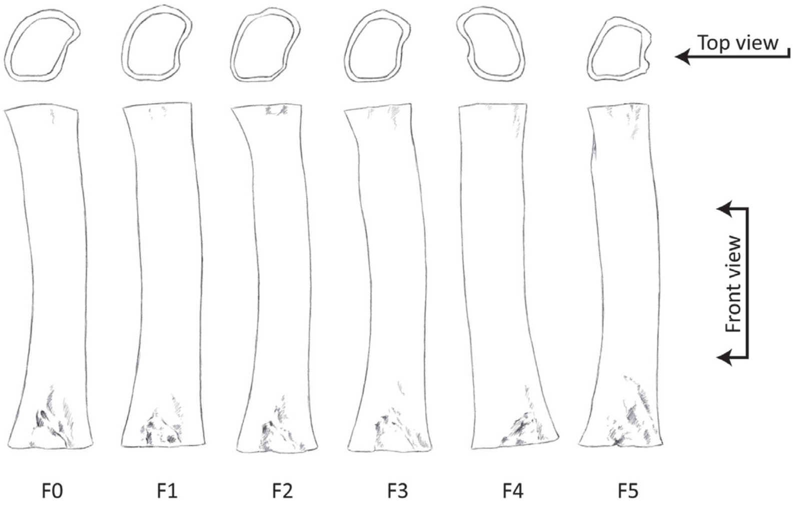


Fig. 2. Morphological variations between bones used for crafting F0 to F5 in the Handcrafting process experiments explained below. Deer femurs show several constants, such as a bulge characterizing the distal part of the epiphysis, a triangular and irregular depression characterizing its proximal part, and a slimming zone in the concave area of the bone's bean-like cross-section. Despite those constants, there never are two identical bones.

In order to illustrate this phenomenon, we chose to handcraft a unique replica of a bone flute in another similar bone (Fig. 3).

The control flute was made in a goat's tibia. It was made very simply, using only steel knife and file, evoking archaeological flutes found in northern Europe for medieval period [5]. The handmade replica was made very carefully, using several measurement tools (caliper, compass, etc.). Also, as the depth of the block changes the pitch, we chose depth 0 (Fig. 4). This calibration is easier to reproduce. We also tried our best to give both blocks a similar soil angle. As a result, the two flutes gave different sounds, with a deviation going from half a tone to more than one tone, increasing as we open the finger holes (Figs. 9 and 10).

This replication test shows how much the sound of a bone flute replica may be deviant from the sound of the control flute it's related to. This phenomenon illustrates the notion of "sound-morphology" as it reveals that every bone has a sound-potential of its own.

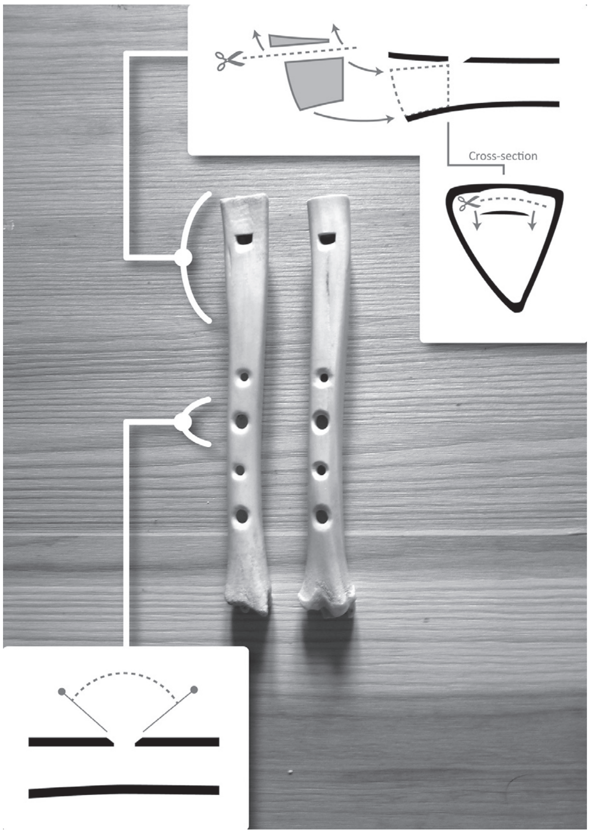


Fig. 3. The control flute (left) and its replica (right) both made out of goat's tibias.

4 Handcrafting Experiments

4.1 Handcrafting Replication Process and Technical Specifications

- **Objectives:** those experiments aim to define the extent of the limitation caused by sound-morphology, as well as to explore the acoustical specificities of this phenomenon. The approach is then different from what we can see in experimental archaeology, as we need here a well-known, functional and replicable bone flute in order to compare its actual sounds with our interpretations.
- **Control and sample:** we chose 6 similar deer femurs with morphological variations. 5 replicas is the minimum sample required for statistical analysis.
- **Chosen sound-geometry:** inner duct-flute with rectangular opening and straight edge (Fig. 5). Combined with a straight geometry, this configuration creates powerful blowing constraints and is easier to reproduce.
- **Manufacture:** handcrafted in January 2016.
- **Sound capture and analysis:** because of lack of means, we had to use a common recording device (smartphone) and a free software (audacity). Having no mechanical blower nor anechoic chamber available at the time, we had to record the sound using natural blowing (as homogeneous as possible) and the same context (a chosen room). Thankfully, the studied phenomenons are contrasting enough to be well illustrated even with a lack of technical means.

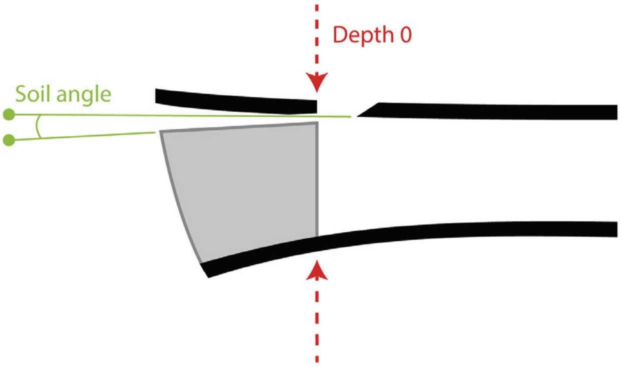


Fig. 4. Illustration of Depth 0 and soil angle

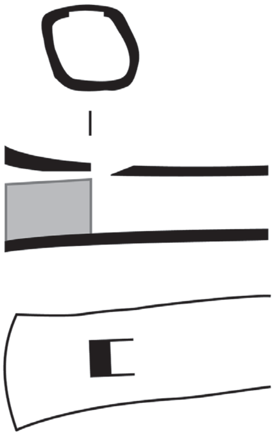


Fig. 5. Depiction of the sound-geometry used for the handcrafting experiments

4.2 Sound Results

The diagrams in Fig. 8 represent the results of basic acoustical analysis of the control flute and its 5 replicas. They obviously show that each individual is different from the control flute.

4.3 Statistics and Discussion

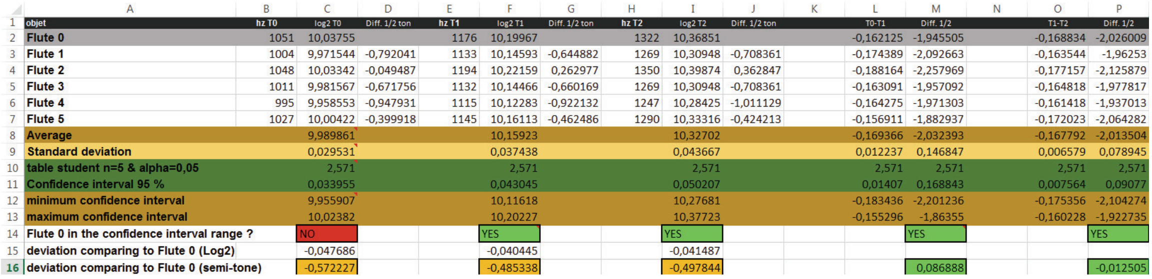
The table (Table 1) represents statistical analysis made on the recorded frequencies. In order to compare them properly, we had to translate them from Hertz to logarithmic scale (base 2 logarithm).

This table shows heterogeneous frequencies and intervals deviations comparing the sample to the control flute, as well as between each individual from the sample itself. Even if the frequency deviations are mostly non-significant regarding statistics (T0 is the only one being significant), the sound estimation they produce is not satisfying for the ear (about one quarter-tone). However, intervals deviations are really small in comparison (about 1/20$^{\text{th}}$ of a tone), which is extremely accurate.

The following facts should also be considered regarding those results:

1. The lower end of the flute was one of the most variable areas and it was then difficult to reproduce an exact geometry in a changing trabecular bone. This could explain T0 deviation.
2. The small sample size is probably involved in those statistical results: a larger sample (20 to 30 replicas) should help us to get better results and thus assess if whether or not this incredibly accurate estimation of intervals is exact. It should also explain the difference between a satisfying intervals reproduction and an unsatisfying frequencies reproduction.
3. The human blow should be ruled out and replaced by a mechanical blower in order to ensure the accuracy of the sound-capture.

Table 1. Statistical analysis of frequencies emitted by F0 to F5 while playing successively T0, T1 and T2. Differences are expressed in semi-tones ("−1" equals "1 semi-tone lower"). The right columns show intervals deviations (T0–T1 and T1–T2).



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	objet	hz T0	log2 T0	Diff. 1/2 ton	hz T1	log2 T1	Diff. 1/2 ton	hz T2	log2 T2	Diff. 1/2 ton		T0-T1	Diff. 1/2		T1-T2	Diff. 1/2
2	Flute 0	1051	10,03755		1176	10,19967		1322	10,36851			-0,162125	-1,945505		-0,168834	-2,026009
3	Flute 1	1004	9,971544	-0,792041	1133	10,14593	-0,644882	1269	10,30948	-0,708361		-0,174389	-2,092663		-0,163544	-1,96253
4	Flute 2	1048	10,03342	-0,049487	1194	10,22159	0,262977	1350	10,39874	0,362847		-0,188164	-2,257969		-0,177157	-2,125879
5	Flute 3	1011	9,981567	-0,671756	1132	10,14466	-0,660169	1269	10,30948	-0,708361		-0,163091	-1,957092		-0,164818	-1,977817
6	Flute 4	995	9,958553	-0,947931	1115	10,12283	-0,922132	1247	10,28425	-1,011129		-0,164275	-1,971303		-0,161418	-1,937013
7	Flute 5	1027	10,00422	-0,399918	1145	10,16113	-0,462486	1290	10,33316	-0,424213		-0,156911	-1,882937		-0,172023	-2,064282
8	Average		9,989861			10,15923			10,32702			-0,169366	-2,032393		-0,167792	-2,013504
9	Standard deviation		0,029531			0,037438			0,043667			0,012237	0,146847		0,006579	0,078945
10	table student n=5 & alpha=0,05		2,571			2,571			2,571			2,571	2,571		2,571	2,571
11	Confidence interval 95 %		0,033955			0,043045			0,050207			0,01407	0,168843		0,007564	0,09077
12	minimum confidence interval		9,955907			10,11618			10,27681			-0,183436	-2,201236		-0,175356	-2,104274
13	maximum confidence interval		10,02382			10,20227			10,37723			-0,155296	-1,86355		-0,160228	-1,922735
14	Flute 0 in the confidence interval range ?		NO			YES			YES				YES			YES
15	deviation comparing to Flute 0 (Log2)		-0,047686			-0,040445			-0,041487							
16	deviation comparing to Flute 0 (semi-tone)		-0,572227			-0,485338			-0,497844				0,086888			-0,012505

5 3D Experiments

5.1 CT-scanning

There exist several possibilities in matter of 3D image acquisition, but CT-scanning was the only viable option because of the very nature of flutes: inner shapes are drastically important and their acoustical properties are extremely sensitive. We needed then a technology that would be able to capture high resolution images both inside and outside of the objects. μ-tomography, also known as μCT, was then the perfect tool. This technology uses X-rays in order to recreate high resolution 3D internal views of an object by compiling the acquired images and is mainly used in medical imaging and industries.

5.2 3D Replication Process and Technical Specifications

- **Objectives:** those experiments aim to question the sound-replication capability of 3D technologies in order to define whether or not they may allow us to pass beyond the sound-morphology limitation endured by handcrafting process. They also aim to assess their own limitations and potential as a sound-reconstruction method.
- **Technologies used:**
  1. **μCT-scanning:** the machine is an X-ray microfocus CT system General Electric (formerly Phoenix) v—tome—x 240D from CRT Morlaix, a resources center dedicated to metrology (http://www.crt-morlaix.com/). In the set-up, the sample is placed on a rotating table, and the X-ray source and detector are stationary.
  2. **3D wire and resin printing:** the machines are a MakerBotReplicator2 from IUT Le Creusot, and a Stratasys Mojo from ENS Rennes. The resin model was printed on a 3D Objet by a contractor.
- **Scanned object:** we chose to scan the control flute used in the sound-morphology principle (the one made from a goat's tibia) in order to compare the 3D results to the handmade replica. The flute was scanned in three parts in order to get a precision of less than 50μ. The reassembly was processed

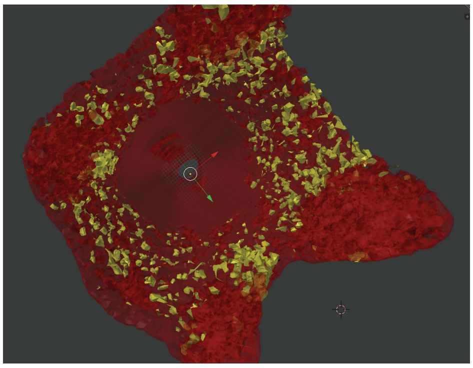


Fig. 6. Disconnected objects (yellow) in the area of the trabecular bone. (Color figure online)

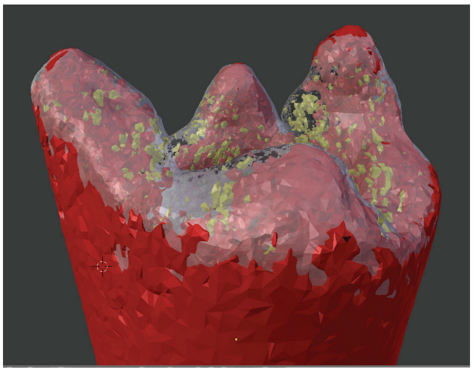


Fig. 7. 3D sculpted patch (transparent gray) on Blender (based on the geometry of the cloud).

with the software Autodesk Meshmixer. Also, as the trabecular bone renders through µ-CT scanning as a cloud of 600+ tiny objects, it cannot be directly printed (Fig. 6). We chose to explore two possibilities: simply removing the objects in one case, and integrating them as a 3D sculpted "patch" in the other (Fig. 7). We used Meshlab and Blender in order to get ready-to-print 3D models.

- **Replicas:** F1 refers to the handmade replica. F2 refers to the 3D orange wire replica (with 3D sculpted "patch", no post-printing treatments). F3 refers to the 3D white wire replica (without the trabecular bone, acetone bath and ultrasounds post-printing treatment). F4 refers to the 3D white resin replica (better printing resolution, with 3D sculpted "patch", no post-printing treatments).
- **Printings:** printed between January and May 2016.
- **Sound capture and analysis:** same context than for the handcrafting process.

5.3 Sound Results

The diagrams in Fig. 9 represent the results of basic acoustical analysis of the control flute and its four replicas.

5.4 Analysis and Discussion

The following tables represent sound-comparisons between the control flute and its replicas using the recorded frequencies translated from Hertz to base 2 logarithm (Table 2).

As we expected, this table shows that 3D printed replicas are globally closest to the original than the handmade one. This is due to the absence of the bone's morphological variability that would occur from using several bones. However, they are not identical between each other (Fig. 10).

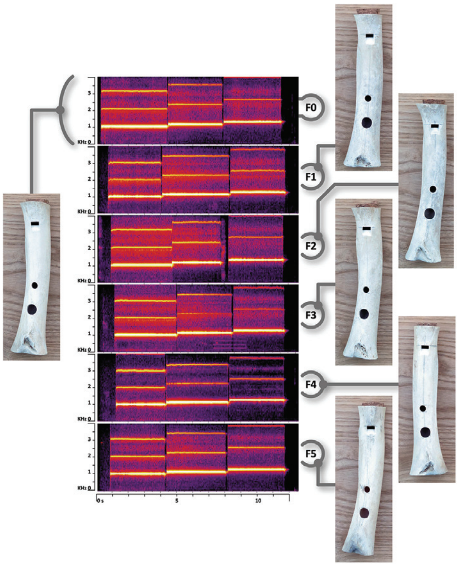


Fig. 8. Diagrams analysis of F0 to F5

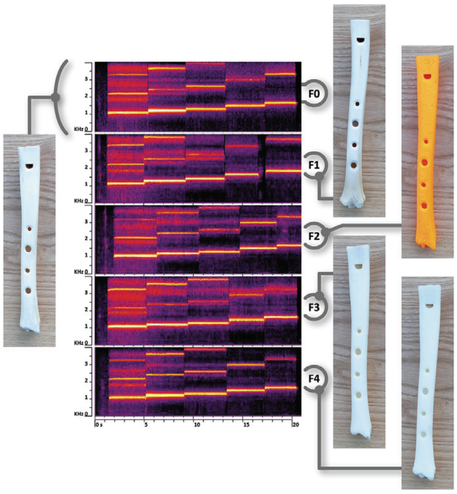
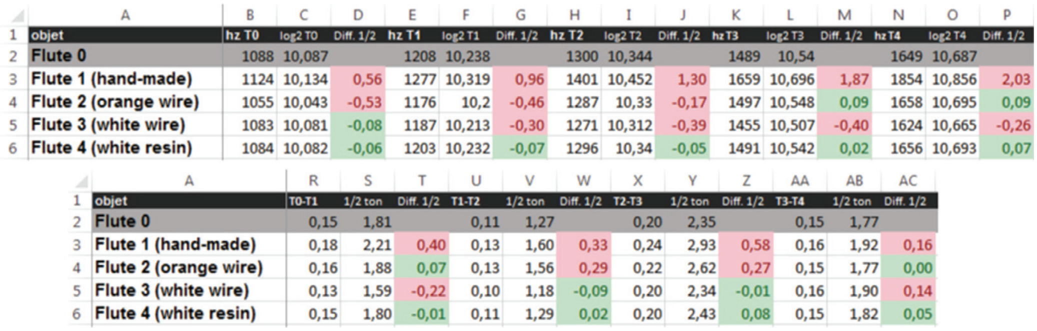


Fig. 9. Diagrams analysis of F0 to F4

Table 2. Comparison between frequencies (top)/intervals (bottom) emitted by F0 to F4 while playing successively T0, T1, T2, T3 and T4. Green cells indicate a sound-reproduction precision of 1/20th of a tone or less.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	objet	hz T0	log2 T0	Diff. 1/2	hz T1	log2 T1	Diff. 1/2	hz T2	log2 T2	Diff. 1/2	hz T3	log2 T3	Diff. 1/2	hz T4	log2 T4	Diff. 1/2
2	Flute 0	1088	10,087		1208	10,238		1300	10,344		1489	10,54		1649	10,687	
3	Flute 1 (hand-made)	1124	10,134	0,56	1277	10,319	0,96	1401	10,452	1,30	1659	10,696	1,87	1854	10,856	2,03
4	Flute 2 (orange wire)	1055	10,043	-0,53	1176	10,2	-0,46	1287	10,33	-0,17	1497	10,548	0,09	1658	10,695	0,09
5	Flute 3 (white wire)	1083	10,081	-0,08	1187	10,213	-0,30	1271	10,312	-0,39	1455	10,507	-0,40	1624	10,665	-0,26
6	Flute 4 (white resin)	1084	10,082	-0,06	1203	10,232	-0,07	1296	10,34	-0,05	1491	10,542	0,02	1656	10,693	0,07

	A	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1	objet	T0-T1	1/2 ton	Diff. 1/2	T1-T2	1/2 ton	Diff. 1/2	T2-T3	1/2 ton	Diff. 1/2	T3-T4	1/2 ton	Diff. 1/2
2	Flute 0	0,15	1,81		0,11	1,27		0,20	2,35		0,15	1,77	
3	Flute 1 (hand-made)	0,18	2,21	0,40	0,13	1,60	0,33	0,24	2,93	0,58	0,16	1,92	0,16
4	Flute 2 (orange wire)	0,16	1,88	0,07	0,13	1,56	0,29	0,22	2,62	0,27	0,15	1,77	0,00
5	Flute 3 (white wire)	0,13	1,59	-0,22	0,10	1,18	-0,09	0,20	2,34	-0,01	0,16	1,90	0,14
6	Flute 4 (white resin)	0,15	1,80	-0,01	0,11	1,29	0,02	0,20	2,43	0,08	0,15	1,82	0,05

In Fig. 10, both orange and white wire flutes present a significant but different deviation regarding their emitted frequencies, whereas the resin flute is the most accurate of them all. Indeed, it reaches the sounds of the original with a precision of less than 1/20th of a tone.

As it appears, acoustical phenomenons related to 3D printed replicas seem to be quite intricate. The following facts should thus be considered regarding those results:

1. 3D wire-printing is processed by fusing a plastic filament which is then deposited by layers, and finally cools down and solidifies. The cooling process comes with a shrinking phenomenon which extent depends on the wire itself as well as on the cooling context (hygrometry and temperature) [17]. Furthermore, these deformations may occur in an irregular way. In other words, 3D wire-printing has a morphological variability of its own.

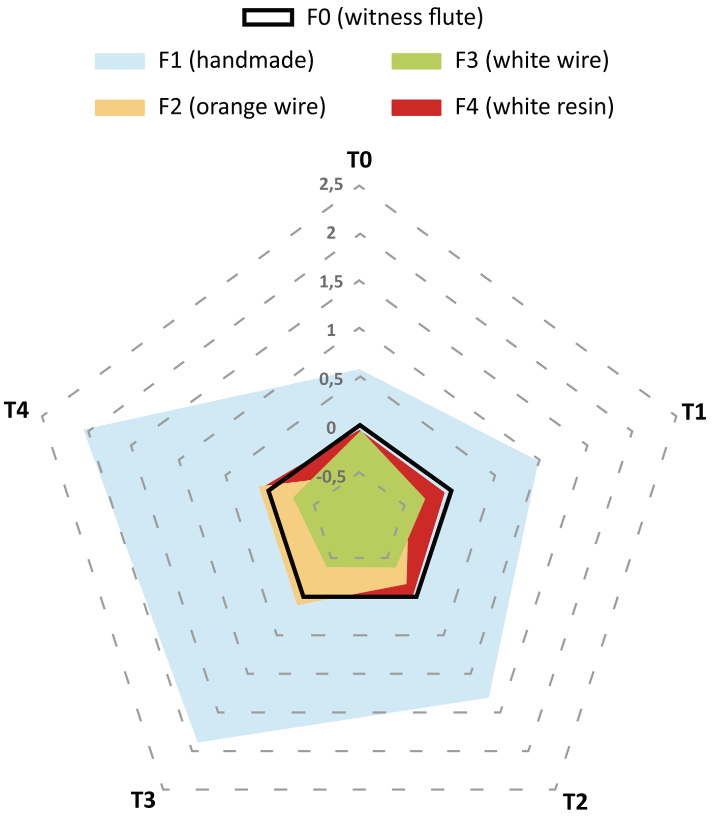


Fig. 10. Diagram representing the sound proximity of each replica comparing to the control flute, for each finger hole (numeric scale in semi-tones). The 0 line represents the control flute. The colored areas represent the replicas' sounds. The more the colored area fills the 0 line, the closest the replica is to the control flute.

2. 3D resin-printing on the other hand does not work the same and thus does not have the same sources of error [19]: it uses a laser impact which solidifies a gelatinous resin. This technology is more accurate than 3D wire-printing and gives different physical results (smoother state of surface, solid 3D printings). That explains why this replica is much more accurate than the other ones.
3. Once again, human blow should be replaced by a mechanical blower.

6 Conclusion

Handcrafting and 3D replication processes illustrate the acoustical complexity of bone flutes, as well as they raise most important epistemological and methodological issues. Succinctly, these results advise of the dangers of sound-interpretations regarding ancient flutes when dealing with replicas. They demonstrate the complexity of the acoustical phenomenons related to naturally hollow raw-materials. They also demonstrate that 3D imagery is not as precise and trustworthy as we would think it would be. However, the use of statistics and of high-precision 3D printers seems to offer a promising track to continue this research. Although there is still much work to do in order to reach a better understanding of this situation, at least we now know that archaeological bone flutes sounds should always be interpreted with caution. In any case, this research will try and go deeper in the epistemological and methodological issues.

Acknowledgments. This project was partially funded by the french CNRS ImagIn IRMA project.

References

1. Avanzini, F., Canazza, S., De Poli, G., Fantozzi, C., Pretto, N., Roda, A., Menegazzi, A.: Archaeology and virtual acoustics - a pan flute from ancient Egypt. In: Proceedings of the 12th International Conference on Sound and Music Computing, pp. 31–36 (2015)
2. Barreau, J.B., Gaugne, R., Bernard, Y., Le Cloirec, G., Gouranton, V.: The West Digital Conservatory of Archaelogical Heritage Project, DH, Marseille, France. Digital Heritage International Congress, pp. 1–8, November 2013
3. Bellia, A.: The virtual reconstruction of an ancient musical instrument: the aulos of selinus. In: Proceedings of Digital Heritage, vol. 1, pp. 55–58 (2015)
4. Borman, T., Stoel, B.: Review of the uses of computed tomography for analyzing instruments of the violin family with a focus on the future. J. Violin Soc. Am. VSA Papers **22**(1), 1–12 (2009)
5. Brade, C.: Die mittelalterlichen Kernspaltflöten Mittel-und Nordeuropas, Ein Beitrage zur Überlieferung prähistorischer und zur Typologie mittelalterlicher Kernspaltflöten, Band. 14, Neumünster, Wachholtz (1975)
6. Clodor-Tissot, T.: Fiche témoins sonores du Néolithique et des Âges des métaux, Industrie de l'os préhistorique. Instruments sonores du Néolithique à l'aube de l'Antiquité, XII (2009)
7. Causs, R., Mille, B., Piechaud, R.: Restitution sonore numérique des cornua de Pompei, Sound Making: Handcraft of Musical Instruments in Antiquity, video recordings of the third IFAO conference, IRCAM (2016). http://medias.ircam.fr/x27292e
8. Garca, B.C., Alcolea, M., Mazo, C.: Experimental study of the aerophone of Isturitz: Manufacture, use-wear analysis and acoustic tests. Quaternary International (2015). doi:10.1016/j.quaint.2015.11.033
9. Garca, B.C.: Methodology for the reconstruction of prehistoric aerophones made of hard animal material, Actas das IV Jornadas de Jovens em Investigação Arqueológica, pp. 411–416 (2011)
10. Gattoni, F., Melgara, C., Sicola, C., Uslenghi, C.M.: Unusual application of computerized tomography: the study of musical instruments. Radiol. Med. **97**(3), 170–173 (1999)
11. Hagel, S.: The meroë pipes: a musical jigsaw puzzle, sound making: handcraft of musical instruments in antiquity, video recordings of the third IFAO conference, IRCAM (2016). http://medias.ircam.fr/x9e8e19
12. Hawkins, J., Jacobson, J., Franklin, J.: Greco-Roman Music in Context; Bringing Sound and Music to Virtual Pompeii, World Conference on E-Learning in Corporate, Government, Health Care, and Higher Education (2011)
13. Koumartzis, N., Tzetzis, D., Kyratsis, P., Kotsakis, R.G.: A new music instrument from ancient times: modern reconstruction of the greek lyre of hermes using 3D laser scanning, advanced computer aided design and audio analysis. J. New Music Res. **44**(4), 324–346 (2015)

14. Kunej, D., Turk, I.: New perspectives on the beginnings of music: archaeological and musicological analysis of a middle Paleolithic bone “flute”. In: The Origins of Music, pp. 235–268 (2000)
15. Laycock, S., Bell, G., Mortimore, D., Greco, M., Corps, N., Finkle, I.: Combining x-ray Micro-CT technology and 3D printing for the digital preservation and study of a 19th century cantonese chess piece with intricate internal structure. ACM J. Comput. Cultl. Heritage **5**(4), 1–7 (2012)
16. Münzel, S.C., Seeberger, F., Hein, W.: The Geissenklösterle flute: discovery, experiments, reconstruction, Studien zur Musikarchäologie. Archäologie früher Klangerzeugung und Tonordnung, Rahden, Verlag M. Leidorf, pp. 107–118 (2002)
17. Pierce, L.B.: Hacking the Digital Print: Alternative image capture and printmaking processes with a special section on 3D printing, New Riders, Voices That Matter, p. 262 (2015)
18. Safa, E.: Handcrafting in archaeomusicological research: record of a one-year apprenticeship alongside a traditional-flute-maker and its application to sound archaeology, Sound Making: Handcraft of Musical Instruments in Antiquity, video recordings of the third IFAO conference, IRCAM (2016). http://medias.ircam.fr/xcf6cf9
19. Sculpteo: 3D Printing Material: PolyJet Resin, Sculpteos website (2016). https://www.sculpteo.com/en/materials/polyjet-resin-material/
20. Sirr, S.A., Waddle, J.R.: CT analysis of bowed stringed instruments. Radiology **203**(3), 801–805 (1997)
21. Sodini, N., Dreossi, D., Chen, R., Fioravanti, M., Giordano, A., Herrestal, P., Zanini, F.: Non-invasive microstructural analysis of bowed stringed instruments with synchrotron radiation X-ray microtomography. J. Cultl. Heritage **13**(3), S44–S49 (2012)
22. Tuniz, C., Bernardini, F., Turk, I., Dimkaroski, L., Mancini, L., Dreossi, D.: Did Neanderthals play music? X-ray computed micro-tomography of the Divje Dabe ’flute’. Archaeometry **54**(3), 581–590 (2012)
23. Turk, I., Blackwell, B.A., Turk, J., Pflaum, M.: Results of computer tomography of the oldest suspected flute from Divje bab I (Slovenia) and its chronological position within global palaeoclimatic and palaeoenvironmental change during last glacial. Anthropologie **110**(3), 293–317 (2006)
24. Tzevelekos, P., Georgaki, A., Kouroupetroglou, G.: HERON: a zournas digital virtual musical instrument. In: Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts, pp. 352–359, September 2015

D RecPhyloXML - a format for reconciled gene trees

This section presents the draft of an unpublished article about a new format for reconciled gene trees (an object that I manipulated extensively during my work). As of yet, most of the different methods that reconcile a species tree and a gene tree have their own output format, which renders complex the creation of flexible analysis pipelines that include reconciliation because many format conversion script must be written (when conversion is possible, as some formats do not allow the inclusion of the data contained in others, resulting in a loss of information). In order to lower the programming burden when manipulating reconciled gene tree (either as output or input of software), I reached out to the community of reconciliation software programmer in order to agree on a common format. Most have answered positively to our call and have agreed to participate to the publication (that should soon be submitted to a scientific journal) and the project. This should be accompanied by the integration of our format to their software, the coding of conversion script to and from existing format, and the design of tools for the manipulation of the format, including a reconciliation visualization tool.

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Subject Section

RecPhyloXML - a format for reconciled gene trees

Wandrille Duchemin [1,2,*], Guillaume Gence [1], Mukul Bansal, Bastien Boussau, François Chevenet, Christophe Dessimoz, David Dylus, David Posada, Celine Scornavacca, Gergely Szollosi, Eric Tannier [1,2], Vincent Daubin [1], *et al...*

[1]Univ Lyon, LBBE (UMR CNRS 5558), Villeurbanne, F-69622, France and
[2]INRIA Grenoble Rhône-Alpes, Montbonnot, F-38334, France.

[*]To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: A reconciliation is an annotation of the nodes of a gene tree with evolutionary events and a mapping onto a species tree. Many algorithms and software produce or use reconciliations but often using exclusive reconciliation formats regarding the type of events considered or whether the species tree is dated or not.

Results: Here we propose a format that aims to promote an integrative albeit flexible specification of phylogenetic reconciliations. This format, named recPhyloXML, is accompanied by several tools such as a reconciled tree visualizer and conversion utilities.

Availability: http://phylariane.univ-lyon1.fr/recphyloxml/

Contact: wandrille.duchemin@univ-lyon1.fr

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The relationship between the history of genomes or species and the history of their constituent genes is often described through a reconciliation. A reconciliation consists of an association between the nodes of a gene tree and the nodes or branches of a species tree, along with different evolutionary events undergone by the gene.

For comprehensive reviews on the subject of reconciliations and their inference, see for example Nakhleh (2013) or Szöllősi *et al.* (2015).

Reconciliations can be used to better understand the history of a specific gene family, but also to study the relationship between several families. They can also be used to infer genome-wide parameters such as rates of gene duplication, loss, or lateral gene transfers (Szöllősi *et al.*, 2013a; Sjöstrand *et al.*, 2014), or population parameters such as divergence time and ancestral population size(Dutheil *et al.*, 2009) Furthermore, reconciliation based metrics can be used as a criterion to construct better gene trees (Durand *et al.*, 2006; Wu *et al.*, 2013; Szöllősi *et al.*, 2013a; Scornavacca *et al.*, 2013; Sjöstrand *et al.*, 2014) or better species tree (Boussau *et al.*, 2013; Nakhleh, 2013).

There are many algorithms and software to infer reconciliations (Nakhleh, 2013; Szöllősi *et al.*, 2015), and while they share many features, each has some unique characteristics.

Some methods work according to a parsimony principle (see for instance (Durand *et al.*, 2006; Bansal *et al.*, 2012; Jacox *et al.*, 2016)) while others rely on a likelihood approach (Åkerborg and Sennblad, 2009; Szöllősi *et al.*, 2013a; Sjöstrand *et al.*, 2014). Reconciliation methods may differ in the type of events they consider. Some methods also require a dated species tree (a species tree where the relative timing of internal speciations is known) while others do not.

The fact that the reconciliation programs (or rather each program family) use different formats to represent reconciliations makes it difficult to compare or use together reconciliation inferred from different software, which can hamper proper comparison and validation studies. This also means that any post-analysis or visualization software will either have limited scope (it will only be able to take as input the reconciliations of specific softwares) or view its release date greatly delayed in order to write a reader for each and every format.

In this paper, we aim to propose a generic reconciliation format encompassing the specificities of different reconciliation programs. This will make reconciliation based analysis more accessible to scientists without the need to develop or use multiple format conversion scripts.

Some events included in reconciliations occur in the species tree, such as speciations or extant representatives of different gene families (*i.e.* the gene tree leaves). Other events included by reconciliations occur along the species tree branches, such as gene duplication (D), gene loss (L), lateral gene transfer (T) or incomplete lineage sorting (ILS) (Than *et al.*, 2008; Rasmussen and Kellis, 2012; Mallo *et al.*, 2014).

Reconciliations can be carried out with dated or undated species trees In a dated species tree, the relative order of speciations is known and it should be possible to include in the reconciliation information about the relative time at which the different events occurred.

(Szöllősi *et al.*, 2013b) introduced a model that considers the existence of extinct or unsampled lineages (*i.e.* branches absent from the gene tree) from which lateral gene transfers might originate. In practice, this means that some transfers, when represented on the species tree can appear to travel to the future (but never in the opposite direction) because they have evolved for a certain time in an unsampled lineage (outside the species tree). Any transfer, even an instantaneous one between lineages of the species tree, can be written in terms of speciation to a dead lineage and transfer from that dead lineage. Thus the format proposed here represents lateral gene transfer as two separate events: leaving the species tree (speciation out), and going back in the species tree (transfer reception).

The notion of evolution in unsampled lineages also implies that a bifurcation in the gene tree can occur in such a lineage. The children of the bifurcation can undergo transfers back to the sampled lineages. The unseen bifurcation might be a duplication, a speciation or a transfer between two unsampled lineage. Existing models are yet unable to discriminate these events. This idea is reflected in our format thanks to a specific way to specify a bifurcation in an unsampled lineage.

There have been previous attempts to develop formats able to represent evolutionary events along a phylogeny. The PhyloXML format (Han and Zmasek, 2009) is able to depict various annotations along a tree. It already has some way of representing evolutionary events along a phylogeny, but with some limitations. For example PhyloXML lacks a mean to specify the species associated with the different events and only include a rudimentary representation of transfers.

Adapting the already existing tags for evolutionary event in PhyloXML would have meant a near complete overhaul, so we decided to create a new format (recPhyloXML) with entirely new tags, ensuring no confusion with PhyloXML.

1.1 state of the art

Existing reconciliation formats can be broadly categorized in two groups.

The first group describes reconciliation events as labels in a newick or NHX tree, in place of the nodal support (e.g., bootstrap) information, or in a devoted NHX comment field. Programs like ALE (Szöllősi *et al.*, 2013a), NOTUNG (Durand *et al.*, 2006; Stolzer *et al.*, 2012), or PrIME (Åkerborg and Sennblad, 2009; Sjöstrand *et al.*, 2014) adhere to this group. The Newick-based reconciliation formats have the advantage of representing the phylogeny. However the reconciliation information often takes the space of other measures like bootstrap values (as in (Szöllősi *et al.*, 2013a)). The NHX-based format solves this by allocating a specific space for the reconciliation. A common problem with NHX and newick-based formats is that some characters are forbidden in the leaf names and annotations[1], while sometimes species or gene annotations contain these characters (whereas they rarely contain whole XML tags).

The second group represents reconciliations as lists of gene tree nodes mapping to species tree nodes, making references to an implicit or external gene tree (meaning that the gene tree structure might not be included in the reconciliation). Examples of such output formats are used by ranger-DTL (Bansal *et al.*, 2012), ecceTERA (Jacox *et al.*, 2016) or the simulation software Simphy (Mallo *et al.*, 2016).

2 Format presentation

recPhyloXML and recGeneTreeXML are two XML grammars inherited from PhyloXML and designed to describe reconciled gene trees.

They both rely on an XML structure composed of tags imbricated in one-another. A specific tag may have different attributes which can be obligatory or facultative.

In this section we briefly detail the structure of the PhyloXML used in our format. We then expand on the tags that are specific to reconciliation.

Fundamental implementation

The recGeneTreeXML grammar allows you to add a new tag `<eventsRec>` in phyloXML `<clade>`. This tag describes the different evolutionary events associated to this clade. To distinguish phyloXML trees from reconciled gene trees inferred by a reconciliation process, the root tag `<phyloxml>` is replaced by `<recGeneTree>`.

The recPhyloXML grammar allows you to store and share one or more reconciled genes trees and the associated species tree. Each reconciled gene tree have to be described using the recGeneTreeXML grammar while the species tree has to be described using the phyloXML grammar.

Common PhyloXML elements

A reconciled gene tree is delimited by the tag `<phylogeny rooted="true"></phylogeny>`. Note that a reconciled gene tree is always rooted. Each clade is then recursively inscribed in a `<clade></clade>` tag. This clade tag possesses a facultative attribute to describe branch length. The name or identifier of the node is given in the `<name></name>` tag. Further information can be included such as support value (`<confidence></confidence>`) or description (`<description></description>`)

recGeneTreeXML

recGeneTreeXML enriches the phyloXML vocabulary by adding the complex tag `<eventsRec>` that must be included inside a `<clade>` tag.

The `<eventsRec>` tag contains the sequence of evolutionary events that occur along a gene tree branch.

Each type of evolutionary event is represented by a specific tag. These can be of two types, according to whether they concern a branch or a node of the gene tree:

- **Non terminal events:** `<speciationLoss>` , `<transferBack>` and `<speciationOutLoss>`. These tags can be used as many times as necessary and in any order. These events do not cause any bifurcation in the gene tree.
- **Terminal events:** `<speciation>`, `<speciationOut>`, `<bifurcationOut>`, `<duplication>` and `<leaf>`. There is exactly one of these tag at the end of the sequence of events contained in the `<eventsRec>` tag.

These terminal events cause either a bifurcation in the gene tree (`<speciation>`, `<speciationOut>`, `<bifurcationOut>`, `<duplication>`) or the end of a lineage (`<leaf>`).

Aside from the `<bifurcationOut>` and `<transferBack>` tags, all tags have an obligatory `speciesLocation` attribute that specifies in which species the event takes place. For `<bifurcationOut>`, the event always take place in an unsampled / extinct lineage. `<transferBack>` events have instead a `destinationSpecies` attribute that specifies the species that receive the transfer. All event tags also have a facultative `confidence` attribute that is intended to store a support value for this event. Additionally, all event tags have a facultative `timeSlice` attribute

[1] These forbidden characters are : `,:();` in newick. In NHX, `[]` are added to this list.

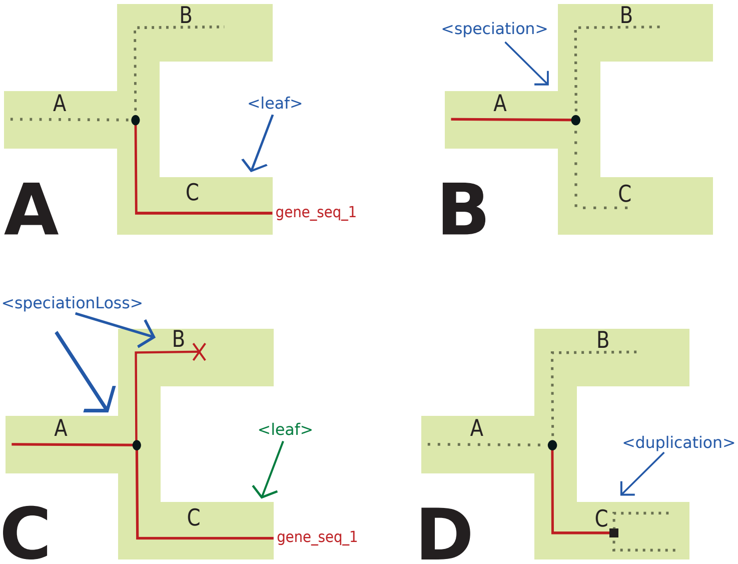


Fig. 1. A. Representation of the `<leaf>` tag. B. Representation of the `<speciation>` tag. C. Representation of the `<speciationLoss>` tag. D. Representation of the `<duplication>` tag. The species tree is figured in green. The part of the gene tree the event occurs in is represented in plain red. Additional parts of the gene tree are represented as dotted black lines.

that can, in models where the species tree is dated and subdivided for instance (as shown in (Doyon *et al.*, 2010), provide information on the timing of the event. Finally, the `<leaf>` tag has a facultative `geneName` attribute that can specify to which extant gene it corresponds.

`<leaf>` tag:

The `<leaf>` tag indicates that the branch ends on a gene tree leaf; see Figure 1 A.

Associated recGeneTreeXML code:

```
<clade>
      <name>gene_seq_1</name>
      <eventsRec>
        <leaf speciesLocation="C"></leaf>
      </eventsRec>
</clade>
```

<speciation> tag:

The `<speciation>` tag describes a gene lineage undergoing a bifurcation due to a speciation; see Figure 1 B.

Associated recGeneTreeXML code:

```
<clade>
  <eventsRec>
    <speciation speciesLocation="A"></speciation>
  </eventsRec>
</clade>
```

`<speciationLoss>` tag:

The `<speciationLoss>` tag describes an event similar to `<speciation>`, with the exception that a gene copy is lost in one of the two descendants resulting from the speciation; see Figure 1 C.

Associated recGeneTreeXML code:

```
<!--Example with end tag <leaf> -->
<clade>
  <name>gene_seq_1</name>
  <eventsRec>
    <speciationLoss speciesLocation="A">
                            </speciationLoss>
    <leaf speciesLocation="C"></leaf>
  </eventsRec>
</clade>
```

`<duplication>` tag:

The `<duplication>` tag represents a gene duplication inside a species tree branch; see Figure 1 D.

Associated recGeneTreeXML code:

```
<clade>
  <eventsRec>
    <duplication speciesLocation="C">
                            </duplication>
  </eventsRec>
</clade>
```

`<speciationOut>` tag:

The `<speciationOut>` tag represents an event analogous to a speciation, but where one of the resulting gene copies occurs in an unsampled/extinct species; see Figure 3 A.

Associated recGeneTreeXML code:

```
<clade>
  <eventsRec>
    <speciationOut speciesLocation="B">
                            </speciationOut>
  </eventsRec>
</clade>
```

<transferBack> tag:
The <transferBack> tag represents an horizontal gene transfer toward a branch of the species tree; see Figure 3 B.

Associated recGeneTreeXML code:

```
<!--Example with end tag <leaf> -->
<clade>
  <eventsRec>
    <transferBack destinationSpecies="E">
                            </transferBack>
    <leaf speciesLocation="E"></leaf>
  </eventsRec>
</clade>
```

<speciationOutLoss> tag:

The <speciationOutLoss> represents a particular case where after a speciation to a lineage absent from the species tree (SpeciationOut), the gene copy that remained inside the species tree is lost; see Figure 3 C.

Associated recGeneTreeXML code:

```
<!--Example with end tag <duplication> -->
<clade>
  <eventsRec>
    <speciationOutLoss speciesLocation="B">
                            </speciationOutLoss>
    <transferBack destinationSpecies="E">
                            </transferBack>
    <duplication speciesLocation="E">
                            </duplication>
  </eventsRec>
</clade>
```

<bifurcationOut> tag:
The <bifurcationOut> tag represents a bifurcation in the species tree that would happen while the gene evolves along an unsampled/extinct species (*ie.* one that is not represented in the species tree, see the <speciationOut> and <transferBack> tags above); see Figure 3 D.

Associated recGeneTreeXML code:

```
<clade>
  <eventsRec>
    <bifurcationOut></bifurcationOut>
  </eventsRec>
</clade>
```

Note on the lateral gene transfer representation
A lateral gene transfer is represented in two steps: one that specifies the species where the transfer originates, the other specifies the species receiving the transfer. This representation follows a model implicating unsampled/extinct lineages that are absent from the species tree (Szöllősi *et al.*, 2013b). In this model, a gene evolving in a given species undergoes a speciation toward a species absent from the species tree; thus, a gene copy exits the species tree. The gene copy situated outside of the species tree will then be transferred back to another species (the gene copy comes back to the species tree). The copy that remains outside the species tree is considered lost as it belongs to a species that is not represented in the species tree. These two successive steps are respectively represented by the <speciationOut> and <transferBack> tags.

recPhyloXML

recPhyloXML facilitates the exchange of several gene family that were reconciled to a same species tree. Its structure is fairly simple. A <recPhylo> root tag contains the following sequence:

- 0 .. 1 species tree. phyloXML format, but contains in the <spTree> tag rather than the <phyloxml> tag.
- 1 .. n gene family tree in recGeneTreeXML format, each defined in a separate <recGeneTree> tag.

```
<!--skeleton of a recphylo object with a species
           tree and two reconciled gene trees -->
<recphylo>
  <spTree>
    ...
  <!--phyloxml species tree -->
  </sptree>
  <recGeneTree>
    ...
    <!-- first reconciled gene tree -->
  </recGeneTree>
  <recGeneTree>
    ...
    <!-- second reconciled gene tree -->
  </recGeneTree>
</recphylo>
```

3 Availability

A detailed description of the recPhyloXML format, as well as a .xsd file[2], is available at **http://phylariane.univ-lyon1.fr/recphyloxml/**. This website also presents a tool that can generate a visual representation of any reconciled tree in the recPhyloXML format, as shown in Figure 3. The generated file is a *.svg* file, allowing easy further manipulation, like changing the color scheme for instance.

The recPhyloXML format has already been implemented as an output option in the reconciliation software ALE (Szöllősi *et al.*, 2013a) and both as input and output options in the adjacency history computing software DeCoSTAR (Duchemin *et al.*, 2017).

Furthermore, scripts have been developed to convert the reconciliations produced by ecceTERA(Jacox *et al.*, 2016), NOTUNG(Durand *et al.*, 2006) and PrIME(Åkerborg and Sennblad, 2009) in recPhyloXML , as well as additional scripts to convert a recPhyloXML reconciled tree in the newick format, or count the different events represented in a recphyloXML file.

[2] This is a file formally describing the format, used by many XML tools.

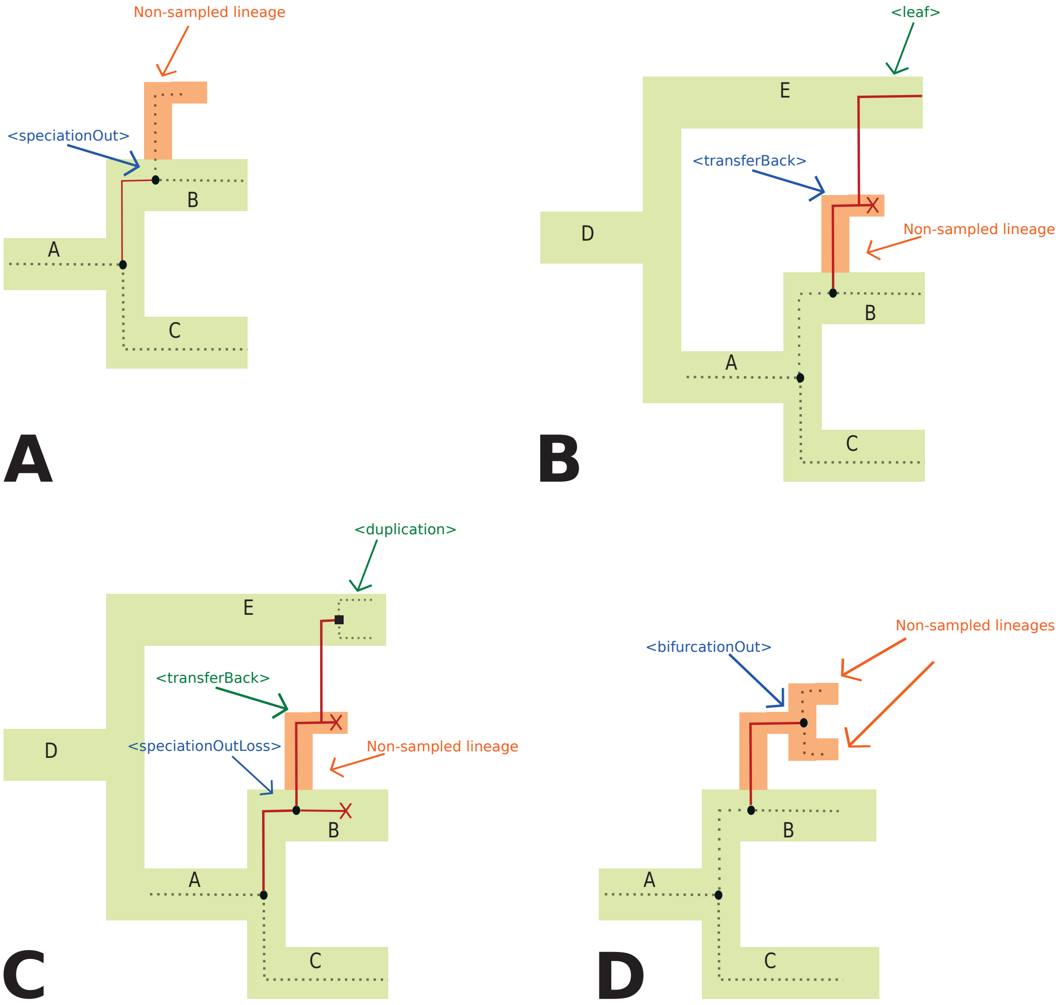


Fig. 2. A. Representation of the <speciationOut> tag. B. Representation of the <transferBack> tag. C. Representation of the <speciationOutLoss> tag. D. Representation of the <bifurcationOut> tag. The species tree is figured in green. Dead / unsampled lineages are represented in orange. The part of the gene tree the event occurs in is represented in plain red. Additional parts of the gene tree are represented as dotted black lines.

APIs have been written to import and export in recPhyloXML for the C++ library Bio++ (Gueguen *et al.*, 2013), for the python libraries ETE3 (Huerta-Cepas *et al.*, 2016) and for Biopython (Cock *et al.*, 2009) (right now, we distribute all these scripts and API by e-mail, on demand. Later they shall be downloadable from the website).

4 Conclusion

With the growing number of available reconciliation models and software, it becomes crucial to be able to exchange and compare their results. RecPhyloXML is a format that can accommodate many reconciliation features (dated / undated ; with or without lateral gene transfers). It relies on an XML structure, a standard format for nested data and that already has multiple API libraries in various programming languages. We provide a detailed description of the recPhyloXML format on a website, along with a tool to visualize it.

Funding

This work is funded by the Agence Nationale pour la Recherche, Ancestrome project ANR-10-BINF- 01-01.

References

Åkerborg, Ö. and Sennblad, B. (2009). Simultaneous Bayesian gene tree reconstruction and reconciliation analysis. *Proceedings of the ...*, **106**(14), 5714–5719.

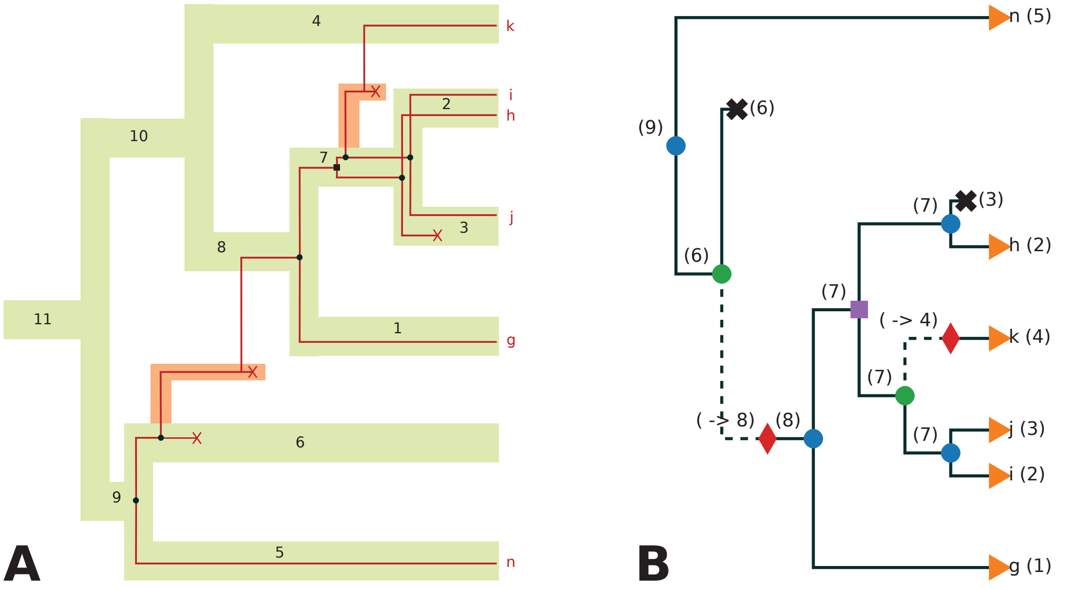


Fig. 3. A. example of a reconciled tree in a species tree (following the conventions described in Fig . B. Example of representation of the same reconciled gene tree by the visualizer available at http://phylariane.univ-lyon1.fr/recphyloxml/. Triangles represent leaves. Blue dots are speciations, green dots are speciations toward an unsampled/dead lineage. Red diamonds represent a transfer reception. Purple squares are duplications. Black crosses symbolize a gene loss. In parenthesis are indicated the id of the species an event takes place in, or the destination of a transfer (then marked with an arrow : −>). Dotted lines represent evolution along an unsampled/dead lineage.

Bansal, M. S., Alm, E. J., and Kellis, M. (2012). Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss. *Bioinformatics*, **28**(12), 283–291.

Boussau, B., Szöllősi, G. J., Duret, L., Gouy, M., Tannier, E., and Daubin, V. (2013). Genome-scale coestimation of species and gene trees. *Genome research*, **23**(2), 323–30.

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., and De Hoon, M. J. L. (2009). Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**(11), 1422–1423.

Doyon, J.-p., Scornavacca, C., Ranwez, V., and Berry, V. (2010). An efficient algorithm for gene / species trees parsimonious reconciliation with losses , duplications , and transfers Concept de l âĽ™ arbre de la vie. (October).

Duchemin, W., Anselmetti, Y., Patterson, M., Ponty, Y., Bérard, S., Chauve, C., Scornavacca, C., Daubin, V., and Tannier, E. (2017). Decostar: Reconstructing the ancestral organization of genes or genomes using reconciled phylogenies. *Genome Biology and Evolution*. to appear.

Durand, D., Halldorsson, B. V., and Vernot, B. (2006). A hybrid micro-macroevolutionary approach to gene tree reconstruction. *Journal of Computational Biology*, **2**(13), 320–335.

Dutheil, J. Y., Ganapathy, G., Hobolth, A., Mailund, T., Uyenoyama, M. K., and Schierup, M. H. (2009). Ancestral population genomics: The coalescent hidden markov model approach. *Genetics*, **183**(1), 259–274.

Gueguen, L., Gaillard, S., Boussau, B., Gouy, M., Groussin, M., Rochette, N. C., Bigot, T., Fournier, D., Pouyet, F., Cahais, V., Bernard, A., Scornavacca, C., Nabholz, B., Haudry, A., Dachary, L., Galtier, N., Belkhir, K., and Dutheil, J. Y. (2013). Bio++: Efficient Extensible Libraries and Tools for Computational Molecular Evolution. *Molecular Biology and Evolution*, **30**(8), 1745–1750.

Han, M. V. and Zmasek, C. M. (2009). phyloXML: XML for evolutionary biology and comparative genomics. *BMC bioinformatics*, **10**, 356.

Huerta-Cepas, J., Serra, F., and Bork, P. (2016). ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data. *Molecular Biology and Evolution*, **33**(6), 1635–1638.

Jacox, E., Chauve, C., Szöllősi, G. J., Ponty, Y., and Scornavacca, C. (2016). ecceTERA : Comprehensive gene tree-species tree reconciliation using parsimony. *Bioinformatics (Oxford, England)*, pages 1–3.

Mallo, D., De Oliveira Martins, L., and Posada, D. (2014). Unsorted homology within locus and species trees. *Systematic Biology*, **63**(6), 988–992.

Mallo, D., de Oliveira Martins, L., and Posada, D. (2016). SimPhy: Phylogenomic Simulation of Gene, Locus and Species Trees. *Systematic biology*, **65**(2), 334–344.

Nakhleh, L. (2013). Computational approaches to species phylogeny inference and gene tree reconciliation. *Trends in Ecology and Evolution*, **28**(12), 719–728.

Rasmussen, M. D. and Kellis, M. (2012). Unified modeling of gene duplication, loss, and coalescence using a locus tree. *Genome research*, **22**(4), 755–65.

Scornavacca, C., Paprotny, W., Berry, V., and Ranwez, V. (2013). Representing a Set of Reconciliations in a Compact Way. *Journal of Bioinformatics and Computational Biology*, **11**(02), 1250025.

Sjöstrand, J., Tofigh, A., Daubin, V., Arvestad, L., Sennblad, B., and Lagergren, J. (2014). A bayesian method for analyzing lateral gene transfer. *Systematic Biology*, **63**(3), 409–420.

Stolzer, M., Lai, H., Xu, M., Sathaye, D., Vernot, B., and Durand, D. (2012). Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees. *Bioinformatics*, **28**(18), 409–415.

Szöllősi, G. J., Rosikiewicz, W., Boussau, B., Tannier, E., and Daubin, V. (2013a). Efficient exploration of the space of reconciled gene trees. *Systematic biology*, **62**(6), 901–12.

Szöllősi, G. J., Tannier, E., Lartillot, N., and Daubin, V. (2013b). Lateral gene transfer from the dead. *Systematic Biology*, **62**(3), 386–397.

Szöllősi, G. J., Tannier, E., Daubin, V., and Boussau, B. (2015). The inference of gene trees with species trees. *Systematic Biology*, **64**(1), e42–e62.

Than, C., Ruths, D., and Nakhleh, L. (2008). Phylonet: a software package for analyzing and reconstructing reticulate evolutionary relationships. *BMC Bioinformatics*, **9**(1), 322.

Wu, Y.-C., Rasmussen, M. D., Bansal, M. S., and Kellis, M. (2013). TreeFix: Statistically Informed Gene Tree Error Correction Using Species Trees. *Systematic Biology*, **62**(1), 110–120.