Evolution of gene neighborhoods within reconciled phylogenies

Séverine Bérard¹,², Coralie Gallien¹, Bastien Boussau³,⁴, Gergely J. Szöllősi³, Vincent Daubin³ and Eric Tannier¹,⁵,*

¹Univ Montpellier2, UMR AMAP, Montpellier F-34000, ²LIRMM, CNRS, Univ Montpellier2, Montpellier F-34392, France, ³LBBE, UMR CNRS 5558, Université de Lyon 1, Villeurbanne F-69622, France, ⁴Department of Integrative Biology, UC Berkeley 4163A Valley Life Sciences Bldg Berkeley, CA 94720-3140, USA and ⁵INRIA Rhône-Alpes, Montbonnot F-38322, France

ABSTRACT

Motivation: Most models of genome evolution integrating gene duplications, losses and chromosomal rearrangements are computationally intractable, even when comparing only two genomes. This prevents large-scale studies that consider different types of genome structural variations.

Results: We define an ‘adjacency phylogenetic tree’ that describes the evolution of an adjacency, a neighborhood relation between two genes, by speciation, duplication or loss of one or both genes, and rearrangement. We describe an algorithm that, given a species tree and a set of gene trees where the leaves are connected by adjacencies, computes an adjacency forest that minimizes the number of gains and breakages of adjacencies (caused by rearrangements) and runs in polynomial time. We use this algorithm to reconstruct contiguous regions of mammalian and plant ancestral genomes in a few minutes for a dozen species and several thousand genes. We show that this method yields reduced conflict between ancestral adjacencies. We detect duplications involving several genes and compare different modes of evolution between phyla and lineages.

Availability: C++ implementation using BIO++ package, available upon request to Séverine Bérard.

Contact: Severine.Berard@cirad.fr, Eric.Tannier@inria.fr

Supplementary information: Supplementary materials available at Bioinformatics online.

1 INTRODUCTION

A phylogenetic tree describes the kin relationships between a set of homologous objects. Non-homologous objects may have other types of relationships, such as interactions, functional relationships, co-expression or neighborhood between genes. Studying the pattern of descent of these relationships can be used to define homology between them, reconstruct ancestral relationships and build phylogenetic trees.

The evolution of gene proximity or interaction has been the subject of numerous recent studies. It is for example a way to assess co-evolution between genes, even if often co-evolution is detected by searching for similarities in gene trees, but without modeling explicitly the relation that make the genes co-evolve (Rodionov et al., 2008; Tuller et al., 2009).

Closer to our study, Pousses et al. (2009) and (Dokukowski and Tiunov, 2009) or (Ma et al., 2008) propose methods to reconstruct ancestral protein–protein interactions or gene neighborhoods based on a model of evolution allowing gene duplications. They, however, assume that the chronology of duplications is known, which often is not the case. (Paro et al., 2011) define a general problem of network evolution without this assumption and give a heuristic solution for the comparison of two species. Our model considers the more specific problem of gene neighborhoods on chromosomes, but generalizes (Paro et al., 2011)’s method in that it handles an arbitrary number of species and provides an exact solution to a less constrained problem.

Several methods are aimed at building ancestral chromosomes (which can be seen as relationships between genes). Most of these methods, however, ignore duplications and losses and are limited to gene families which have exactly one representative in each studied species (Alekseyev and Pevzner, 2009; Chauve and Tannier, 2008). The number of such gene families becomes smaller and smaller as the number of species grows. Some methods take as input gene trees allowing duplications and losses (Lajoie et al., 2011), but do not model these events and treat them as noise that is removed for the construction of chromosomes by traveling salesman-like optimization methods. (Chauve et al., 2011). (Muffato et al., 2011) or (Zheng and Sankoff, 2011) model duplications only in the context of whole genome duplications.

Here, we propose a method that takes a species tree and a set of gene trees as inputs, and models the gain and breakage of gene adjacencies along a pair of trees, taking duplications and losses into account. We consider two genes to be ‘adjacent’ if they are on the same chromosome in the same genome and no other gene is located among lineages.

We assume that adjacencies evolve independently from each other, so we do not model the rearrangement explicitly (inversions, translocations etc.), but model their effect on adjacencies, which thus can undergo gains and breakages.

Doing this, we solve a problem that fits in the methodological program started by Sankoff and El-Mabrouk, 2000, which mixes rearrangements and reconciliations of phylogenetic trees (a reconciliation is an annotation of gene tree nodes by duplication or speciation events, according to a species tree).

Algorithmically, the dynamic programming principle we use generalizes the Sankoff–Fitch (Fitch, 1971; Sankoff, 1975) algorithmically intractable, even when comparing only two genomes. This prevents large-scale studies that consider different types of genome structural variations.

Results: We define an ‘adjacency phylogenetic tree’ that describes the evolution of an adjacency, a neighborhood relation between two genes, by speciation, duplication or loss of one or both genes, and rearrangement. We describe an algorithm that, given a species tree and a set of gene trees where the leaves are connected by adjacencies, computes an adjacency forest that minimizes the number of gains and breakages of adjacencies (caused by rearrangements) and runs in polynomial time. We use this algorithm to reconstruct contiguous regions of mammalian and plant ancestral genomes in a few minutes for a dozen species and several thousand genes. We show that this method yields reduced conflict between ancestral adjacencies. We detect duplications involving several genes and compare different modes of evolution between phyla and lineages.

Availability: C++ implementation using BIO++ package, available upon request to Séverine Bérard.

Contact: Severine.Berard@cirad.fr, Eric.Tannier@inria.fr

Supplementary information: Supplementary materials available at Bioinformatics online.

1 INTRODUCTION

A phylogenetic tree describes the kin relationships between a set of homologous objects. Non-homologous objects may have other types of relationships, such as interactions, functional relationships, co-expression or neighborhood between genes. Studying the pattern of descent of these relationships can be used to define homology between them, reconstruct ancestral relationships and build phylogenetic trees.

The evolution of gene proximity or interaction has been the subject of numerous recent studies. It is for example a way to assess co-evolution between genes, even if often co-evolution is detected by searching for similarities in gene trees, but without modeling explicitly the relation that make the genes co-evolve (Rodionov et al., 2008; Tuller et al., 2009).

Closer to our study, Pousses et al. (2009) and (Dokukowski and Tiunov, 2009) or (Ma et al., 2008) propose methods to reconstruct ancestral protein–protein interactions or gene neighborhoods based on a model of evolution allowing gene duplications. They, however, assume that the chronology of duplications is known, which often is not the case. (Paro et al., 2011) define a general problem of network evolution without this assumption and give a heuristic solution for the comparison of two species. Our model considers the more specific problem of gene neighborhoods on chromosomes, but generalizes (Paro et al., 2011)’s method in that it handles an arbitrary number of species and provides an exact solution to a less constrained problem.

Several methods are aimed at building ancestral chromosomes (which can be seen as relationships between genes). Most of these methods, however, ignore duplications and losses and are limited to gene families which have exactly one representative in each studied species (Alekseyev and Pevzner, 2009; Chauve and Tannier, 2008). The number of such gene families becomes smaller and smaller as the number of species grows. Some methods take as input gene trees allowing duplications and losses (Lajoie et al., 2011), but do not model these events and treat them as noise that is removed for the construction of chromosomes by traveling salesman-like optimization methods. (Chauve et al., 2011). (Muffato et al., 2011) or (Zheng and Sankoff, 2011) model duplications only in the context of whole genome duplications.

Here, we propose a method that takes a species tree and a set of gene trees as inputs, and models the gain and breakage of gene adjacencies along a pair of trees, taking duplications and losses into account. We consider two genes to be ‘adjacent’ if they are on the same chromosome in the same genome and no other gene is located among lineages.

We assume that adjacencies evolve independently from each other, so we do not model the rearrangement explicitly (inversions, translocations etc.), but model their effect on adjacencies, which thus can undergo gains and breakages.

Doing this, we solve a problem that fits in the methodological program started by Sankoff and El-Mabrouk, 2000, which mixes rearrangements and reconciliations of phylogenetic trees (a reconciliation is an annotation of gene tree nodes by duplication or speciation events, according to a species tree).

Algorithmically, the dynamic programming principle we use generalizes the Sankoff–Fitch (Fitch, 1971; Sankoff, 1975)
Evolution of gene neighborhoods

Fig. 1. Examples of a species tree (left), two gene trees (middle) and an adjacency tree (right). Blue dots are speciation nodes. Leaves are extant (species, genes, adjacencies), except the one labeled by a red cross (gene loss) or a red flash (breakage). Green squares are (gene or adjacency) duplication nodes. Gene labels refer to the species they belong to. Every node of the adjacency tree is labeled by a couple of nodes from gene trees

\[ E(\mathcal{N}) = \text{Extant} \text{ for all leaves } \mathcal{N}. \text{ All } \mathcal{S}(\mathcal{N}) \text{ are distinct species and if } \mathcal{N} \text{ is an internal node, } \mathcal{S}(\mathcal{N}) \text{ defines an ancestral species.} \]

(2) A gene tree \( T_G \) describes the evolution of a family of homologous genes along a species tree \( T_S \). All gene trees here are 'reconciled' with the 'LCA (Last Common Ancestor) reconciliation' (Goodman et al. 1979) where all gene losses are represented by leaves, which means every node \( \mathcal{N} \) verifies:

- If \( \mathcal{N} \) is a leaf, then \( E(\mathcal{N}) \in \{\text{Extant, GLos}\} \) and if \( \mathcal{N} \) is an internal node, then \( E(\mathcal{N}) \in \{\text{Spec, GDup}\} \).
- If \( E(\mathcal{N}) = \text{Extant} \), then there is a gene \( G(\mathcal{N}) \) that belongs to \( S(\mathcal{N}) \) and all such genes are distinct.
- If \( E(\mathcal{N}) = \text{GDup} \) then the children \( N1 \) and \( N2 \) of \( \mathcal{N} \) are such that \( S(N1) = S(N2) = S(\mathcal{N}) \).
- If \( E(\mathcal{N}) = \text{Spec} \) then the children \( N1 \) and \( N2 \) of \( \mathcal{N} \) are such that there are two edges \( AA1 \) and \( AA2 \) of \( T_S \) such that \( P(A1) = P(A2) = A \) and \( S(N1) = S(S(A)) \) and \( S(N2) = S(S(A)) \).
- Let \( L \) be the set of leaves of \( T_G(N) \); Let \( S(L) \) be the set of all extant species which are descendants of some \( S(f) \) of \( L \); Let now \( N_0 \) be the lowest node in \( T_S \) such that \( S(L) \subseteq \{\text{Extant, GLos}\} \); Then, \( S(N_0) = S(\mathcal{N}) \).

(3) An adjacency tree \( T_A \) describes the descent pattern of adjacencies. As adjacencies are pairs of genes, they follow the evolution of genes: if an adjacency \( AB \) descents from an adjacency \( CD \), then \( A \) descents from \( C \) and \( B \) from \( D \). So adjacency trees are defined given a set of reconciled gene trees \( T_S \) and have to follow their LCA reconciliations. Formally, every node \( \mathcal{N} \) of an adjacency tree verifies:

- If \( \mathcal{N} \) is a leaf, then \( E(\mathcal{N}) \in \{\text{Extant, GLos, Alos, Break}\} \) and if \( \mathcal{N} \) is an internal node, then \( E(\mathcal{N}) \in \{\text{Spec, GDup, ADup}\} \).
- If \( E(\mathcal{N}) \neq \text{Break} \) then there is a couple \( A(\mathcal{N}) = XY \) of gene tree nodes \( X \) and \( Y \) (possibly from different gene trees) such that \( S(X) = S(Y) \).
- If \( E(\mathcal{N}) = \text{Extant} \), then \( G(X)G(Y) \) is an adjacency.
- If \( E(\mathcal{N}) = \text{GLos} \), then \( X = \text{GLos} \) or \( Y = \text{GLos} \) (and not both).
- If \( E(\mathcal{N}) = \text{Alos} \), then \( X = \text{GLos} \).
- If \( E(\mathcal{N}) = \text{Spec} \), then \( E(X) = E(Y) = \text{GLos} \). In addition, \( \mathcal{N} \) has two children \( N1 \) and \( N2 \) and either \( E(N1) = \text{Break} \) (respectively, \( E(N2) = \text{Break} \)) or \( E(A(N1)) \) (respectively, \( E(A(N2)) \)) is a couple of children of \( X \) and \( Y \).

[Even if 'Extant' is not an evolutionary event, it is included because it annotates some tree leaves.]
nodes

Gain
each adjacency

A

adjacency forest of minimum cost. We give a polynomial algorithm

to compute an

where the adjacency can have been gained earlier in the evolution.

Gain

where

Gain

duplication propagates two. It is thus possible to catch the places

independent duplications propagate only one adjacency, and the joint
discriminated because the two do not have the same effect: the

solutions. The duplication of an adjacency has the same cost as

genes, and as long as one gene has its own events and belongs

we make the hypothesis of independent evolution of couples of
duplication and loss events in the objective function. This is because

that there is no need that in ancestral genomes, genes participate to

the problem to other kinds of relationships. The drawback of this is

adjacencies. We have not required this property in the input of the

In extant genomes, one gene can participate in at most two

3.2 The linearity of genomes

3.1 The cost of a duplication or loss event

• If $E(N) = ADup$, then $E(X) = E(Y) = GDup$. In addition, $N$ has two children $N_1$ and $N_2$ either $E(N_1) = Break$ (respectively, $E(N_2) = Break$) or $A(N_1)$ (respectively, $A(N_2)$) is a couple of genes of $X$ and $Y$.

• If $E(N) = GDup$, then $E(X) = GDup$ or $E(Y) = GDup$ (suppose it is $Y$). In addition, $N$ has only one child $N_1$ and either $E(N_1) = Break$ or $A(N_1)$ is a couple of genes composed of $X$ and one child of $Y$.

An ‘adjacency forest’ is a set of adjacency trees, such that for two

nodes $N_1$ and $N_2$ in this forest, $A(N_1) \neq A(N_2)$, and such that for each adjacency $A$ from any species, there exists a leaf $L$ in the forest, which verifies $A(L) = A$.

The cost of an adjacency tree $T_A$, is

$C(T_A) = |\{v \in V(T_A) \mid E(v) = Break\}| + C(Break) + Gain(T_A),$

where $Gain(T_A)$ is computed in this way: if the root $R$ of $T_A$ is such that $A(R) = XY$ and either

• $P(X) = P(Y)$ or

• $X$ is the root of a gene tree, and either $Y$ is also a root, or $S(P(Y)) = S(T_Y)$

then $Gain(T_A) = 0$, else $Gain(T_A) = C(Gain)$. The cases where $Gain(T_A) = 0$ are those arising from tandem duplications or those where the adjacency can have been gained earlier in the evolution.

The cost of an adjacency forest is the sum of the costs of all

adjacency trees.

The problem we address is to take as input a species tree, a set of
gene trees and a set of extant adjacencies, and to compute an
adjacency forest of minimum cost. We give a polynomial algorithm
which gives one optimal solution.

3 PROPERTIES

3.1 The cost of a duplication or loss event

The optimization focuses only on breaks and gains of adjacencies.
The dynamic programming technique we use does not allow to count
duplication and loss events in the objective function. This is because
we make the hypothesis of independent evolution of couples of
genes, and as long as one gene has its own events and belongs
to several couples, this independence is broken.

Nevertheless, duplication events have an importance for the
solutions. The duplication of an adjacency has the same cost as
the independent duplication of two genes, but the events can still
be discriminated because the two do not have the same effect: the
independent duplications propagate only one adjacency, and the joint
duplication propagates two. It is thus possible to catch the places
where a joint duplication is advantageous in terms of gains and
breaks.

3.2 The linearity of genomes

In extant genomes, one gene can participate in at most two
adjacencies. We have not required this property in the input of the
program because it is not used, and in this way we could easily adapt
the problem to other kinds of relationships. The drawback of this is that
there is no need that in ancestral genomes, genes participate to
at most two adjacencies.
Evolution of gene neighborhoods

2a. If \( G_1 \neq G_2 \), then there are two nodes \( N_1 \in G_1, N_2 \in G_2 \) such that \( S(N_1) = S(N_2) \) and \( A \) and \( C \) are descendants of \( N_1 \), while \( B \) and \( D \) are descendants of \( N_2 \).

2b. If \( G_1 = G_2 \), then the lowest common ancestor of \( A \) and \( B \) is the same node as the lowest common ancestor of \( C \) and \( D \) (it is necessarily a duplication node).

This relation between adjacencies satisfying all conditions is an equivalence relation (reflexive, symmetric and transitive). Equivalence classes are treated independently. This is justified by the following lemma, whose proof stands in the Supplementary Material.

**Lemma 1.** If there is a set of adjacencies which contains adjacencies \( AB \) and \( CD \), then \( AB \) and \( CD \) are in the same equivalence class.

In other words, if \( AB \) and \( CD \) are not in the same class they cannot be homologous. The converse is not true however. Solutions for one class may consist of several adjacency trees.

This clustering allows to divide the problem into equivalence classes, which concern one or two gene trees. If in an equivalence class, adjacencies have extremities in the same tree, by definition of the classes, there is a common ancestor to all pairs of extremities of adjacencies. By removing this vertex, we get two trees rooted at its children, and all adjacencies have one extremity in each of these two trees.

So we may restrict ourselves to this case where we have exactly two gene trees and all adjacencies are between these two trees. Moreover, we may suppose that each tree is rooted at the lowest common ancestor of all genes involved in adjacencies of the chosen class, because we may simply consider the subtrees rooted at this vertex. This yields that the two roots are necessarily assigned to the same species.

### 4.2 Recurrence formulas

Formally, we have two gene trees \( T_G^1 \) and \( T_G^2 \), extant adjacencies have one extremity in each tree, and if \( R_1 \) and \( R_2 \) are the respective roots of \( T_G^1 \) and \( T_G^2 \), then \( S(R_1) = S(R_2) \).

For a pair of nodes \((v^1, v^2) \in V(T_G^1) \times V(T_G^2) \) such that \( S(v^1) = S(v^2) \), we compute two values, \( c_1(v^1, v^2) \) and \( c_0(v^1, v^2) \) by recurrence formulas described in the sequel. Remark that we only consider pairs of nodes annotated with the same species because an adjacency is always linking genes from the same genome. We prove that these numbers have the following properties (proofs are in the [Supplementary Material](#) Appendix 2).

**Theorem 1.**

1. If \( c_1(v^1, v^2) \) is the minimum cost of an adjacency forest \( F \) for the adjacencies between two gene trees \( T_G^1(v^1) \) and \( T_G^2(v^2) \), such that there is a node \( N \) in \( F \) with \( A(N) = v^1v^2 \).

2. \( c_0(v^1, v^2) \) is the minimum cost of an adjacency forest \( F \) for the adjacencies between two gene trees \( T_G^1(v^1) \) and \( T_G^2(v^2) \), such that there is no node \( N \) in \( F \) with \( A(N) = v^1v^2 \).

In consequence, the minimum cost of an adjacency forest will be given by computing the minimum between \( c_1(R_1^1, R_2^1) \) and \( c_0(R_1^1, R_2^1) \), following a case analysis, according to the type of event associated to \( v^1 \) and \( v^2 \). The roles of \( v^1 \) and \( v^2 \) are symmetrical. We note \( ca(v) \) and \( ch(v) \) the two children of a node \( v \).

**Case 1.** \( E(v^1) = Extant \) and \( E(v^2) = Extant \). If \( v^1v^2 \) is an adjacency then \( c_1(v^1, v^2) = 0 \) and \( c_0(v^1, v^2) = \infty \); else \( c_1(v^1, v^2) = \infty \) and \( c_0(v^1, v^2) = 0 \).

**Case 2.** \( E(v^1) = GLoc \) and \( E(v^2) \neq GLoc \). In this case \( c_1(v^1, v^2) = 0 \) and \( c_0(v^1, v^2) = 0 \).

**Case 3.** \( E(v^1) = CLoc \) and \( E(v^2) = GLoc \). In this case \( c_1(v^1, v^2) = 0 \) and \( c_0(v^1, v^2) = 0 \). This case has to be distinguished from the previous one for the backtracking procedure described in the following subsection.

**Case 4.** \( E(v^1) = \{Extant, Spec\} \) and \( E(v^2) = GDup \).

**Case 5.** \( E(v^1) = Spec \) and \( E(v^2) = Spec \).

**Case 6.** \( E(v^1) \in \{Extant, Spec\} \) and \( E(v^2) = GDup \).
follows each cost on the chosen line in the recurrence formulas, creating adjacency trees from root to leaves. A node \( N \) with \( A(N) = v^1v^2 \) is created each time \( c_j(v^1, v^2) \) is chosen. The event labeling this node depends on the events labeling \( v^1 \) and \( v^2 : \textit{Extant} \) for Case 1, \( \textit{GLos} \) for Case 2, \( \textit{ALos} \) for Case 3, \( \textit{GDup} \) for Cases 4 and 6, (\( D1\&D2 \)), \textit{Spec} for Case 5 and \( \textit{ADup} \) for Case 6 (\( D12 \)). A node \( N \) with \( E(N) = \textit{Break} \) is created each time there is a \( C \textit{(Break)} \) in the chosen formula.

Edges between the nodes follow the pattern of descent between adjacencies:

- \textit{Break} nodes are leaves, and their parent are the nodes constructed in the formula where \( C \textit{(Break)} \) occurs;
- In Cases 4 and 6 (\( D1\&D2 \)), there is an edge between \( v^1v^2 \) and one of \( v^1c(a(v^2)), v^1c(b(v^2)), c(a(v^2))^2, c(b(v^2))^2 \), if \( c_1 \) is chosen for either of them.
- In Cases 5 and 6 (\( D12 \)), there is an edge between \( v^1v^2 \) and one or two of \( c(a(v^2)c(b(v^2)), c(b(v^2)c(a(v^2)) \), \( c(a(v^2)c(b(v^2))) \) if \( c_1 \) is chosen for either of them (there can be arbitrary choices for equivalent solutions).

Recurrence formulas imply that the backtracking procedure does not create twice the same node: each formula computes the cost for \( v^1v^2 \) between pairs of nodes where at least one is a descendant of \( v^1 \) or \( v^2 \).

An example of an algorithm input and output is drawn on Figure 4.

4.4 Complexity

The algorithm takes as input a dataset composed by a species tree, several gene trees and a list of adjacencies. It first computes the equivalence classes of adjacencies. Then for each class it constructs two subtrees to compute \( c_0 \) and \( c_1 \) costs on their roots and applies the backtracking procedure. The algorithm outputs the adjacency forest resulting from the union of all adjacency forests built on each class.

Let \( n \) be the number of gene trees and \( k \) be the maximum size of a tree. The algorithm runs in \( O(n^2 \times k^2) \). Indeed, the maximum number of adjacency equivalence classes is bounded by \( O(n^2) \), while for each equivalence class, every couple of node is examined with a constant-time case analysis.

In practice, the number of equivalence classes is much closer to \( n \) than to \( n^2 \) and trees are small compared with the total number of genes. For all datasets we tested, including dozens of species and thousand of genes, the execution time was under 10 min.

5 IMPLEMENTATION AND APPLICATION

We implemented the algorithm using the Biomega platform [Dutheil et al., 2004]. The program, named \( \textit{DeCo} \) (Detection of Co-evolution or DeCoration of trees), takes as input a species tree, a set of genes along with the species they are in, a set of adjacencies and a set of gene trees.

We tested it on four datasets, with costs \( C \textit{(Gain)} = C \textit{(Break)} = 1 \). The first and second datasets are based on 5039 gene trees from the Ensembl database (release 57) restricted to mammalian assembled genomes (11 species) [The first set of trees are those provided in this

\[ v^1, v^2 = \min \]

\[ \text{Cases where the } i^\text{th} \text{ duplication comes first} \]

\[ \text{Cases where the } i^\text{th} \text{ duplication comes first} \]

\[ \text{The parsimony framework practically makes it necessary to work with only assembled genomes, since we would count to many breaks for un assembled} \]
Evolution of gene neighborhoods

Input

\[ C(\text{Gain}) = C(\text{Break}) = 1 \]

possible to envisage branch specific costs, where unassembled genomes ones, preventing the reconstruction of some ancestral adjacencies. It is database, made according to the TreeBeST pipeline \cite{Vilella2009}. The second consists of the trees reconstructed by the PhylDog method \cite{Boussau2014}, with an explicit model of duplication and losses of trees. Both sets of trees were reconciled according to the LCA method \cite{Goodman1979}, which gives gene trees with the properties written in Section 4.

Then, we computed ancestral adjacencies according to the method described here, and compared with the 'pairwise alternative', an implementation of the principles used by \cite{Chauve2008,Muffato2010,Bertrand2014} or \cite{Boussau2014}, in which adjacencies are constructed by comparing couples of species (the method is described in the Supplementary Material, Appendix 3) instead of all genomes together.

We computed the degree of each ancestral gene, that is, the number of adjacencies which has it as an extremity. As shown in Figure 2, most ancestral genes have degree 2, which means the signal of linearity of the ancestral genomes is recovered. We can observe the gain obtained by using PhylDog trees instead of TreeBeST trees (red plain versus blue dotted line), and the gain obtained by using DeCo instead of the pairwise alternative (red plain versus green dashed line). These two gains are nearly equivalent, showing that to get better ancestral genomes, we need good trees as well as good adjacency inference algorithms. Better trees tend to give a better estimate of the ancestral gene content, minimizing the degree 0 (probably wrong) genes, while the adjacency inference algorithm may minimize the number of genes with degree \(>2\), convergent evolution can yield false ancestral adjacencies, which add to the two true ones. Convergent evolution is impossible to handle in a pairwise method.

The third and fourth datasets are constructed from the Ensembl (release 65) and EnsemblPlant (release 12) databases, restricted to some assembled mammalian (11 species, 19,217 gene trees) and angiosperm (9 genomes, 35,182 gene trees, with an average of 9 genes) species. We chose these two clades for a phylogenetic comparison because the estimated divergence times are similar, and there are approximately as many assembled genomes in both. We compared the number of segmental duplications involving more than one gene in these two datasets. In Figure 3, phylogenetic trees of mammals and angiosperms are drawn, in which branch length is the number of pairs of genes duplicated together over the total number of ancestral genes found in the same branch. We find that on average branch lengths are more than three times longer in plants, indicating genome architectures rapidly evolving compared with slow mammalian ones. Angiosperm genomes have been shaped by several whole genome duplications: at the basis of monocots, a triplication at the basis of dicots, plus one event on the Maize and Poplar lineages, and two on the Arabidopsis one. These events probably create a long branch in Poplar, or Glycine, but are not always visible (e.g. in Arabidopsis) due to differentiated losses which blurred the synteny signal. The difference in branch length can partly be due to whole genome duplications. But measuring the average size of the duplicated segments by computing

\[ \#GDup - \#Adup \]

with an average of 10 genes) and angiosperm (9 genomes, 35,182 gene trees, with an average of 9 genes) species. We chose these two clades for a phylogenetic comparison because the estimated divergence times are similar, and there are approximately as many assembled genomes in both. We compared the number of segmental duplications involving more than one gene in these two datasets. In Figure 3, phylogenetic trees of mammals and angiosperms are drawn, in which branch length is the number of pairs of genes duplicated together over the total number of ancestral genes found in the same branch. We find that on average branch lengths are more than three times longer in plants, indicating genome architectures rapidly evolving compared with slow mammalian ones. Angiosperm genomes have been shaped by several whole genome duplications: at the basis of monocots, a triplication at the basis of dicots, plus one event on the Maize and Poplar lineages, and two on the Arabidopsis one. These events probably create a long branch in Poplar, or Glycine, but are not always visible (e.g. in Arabidopsis) due to differentiated losses which blurred the synteny signal. The difference in branch length can partly be due to whole genome duplications. But measuring the average size of the duplicated segments by computing

\[ \#GDup - \#Adup \]

we found no significant difference between the two phyla (\(=1.08\) on average among all branches for both), indicating that the changes in genome architectures following a whole genome duplication are not fully accessible to this method. The long branch at the basis of eutheria would deserve more studies to know to which extent it is artificial and due to the quality of gene trees.

6 PERSPECTIVES

The algorithm can easily be extended to handle other relations than adjacencies (interactions, regulations, co-expression or any functional relation which can evolve by gain or breakage like adjacencies). It can be seen as even more adapted to less constrained relations (without a linear organization). Indeed, if a gene is lost,
Alekseyev, M.A. and Pevzner, P.A. (2009) Breakpoint graphs and ancestral genome reconstructions. *Genome Res.*, **19**, 943–957.

Bertrand, D. et al. (2010) Reconstruction of ancestral genome subject to whole genome duplication, speciation, rearrangement and loss. In *Proceedings of WABI’10, Algorithms in Bioinformatics*, Lecture Notes in Bioinformatics, Springer, Berlin Heidelberg, pp. 78–89.

Bonnass B. et al. (2012) Genome-scale coestimation of species and gene trees, in revision.

Chauve, C. and Tannier, E. (2008) A methodological framework for the reconstruction of contiguous regions of ancestral genomes and its application to mammalian genomes. *PLoS Comput. Biol.*, **4**, e1000234.

Chauve, C. et al. (2010) Yeast ancestral genome reconstructions: the possibilities of computational methods. *II. J. Comput. Biol.*, **17**, 1097–1112.

Doyon, J.-P. et al. (2011) Models, algorithms and programs for phylogeny reconciliation. *Brief. Bioinform.*, **12**, 392–400.

Dufetel, J. et al. (2006) Bis+ a set of c++ libraries for sequence analysis, phylogenetics, molecular evolution and population genetics. *BMC Bioinformatics*, **7**, 188.

Dutkowski, J. and Tiarni, J. (2009) Phylogeny-guided interaction mapping in seven rakerates. *BMC Bioinformatics*, **10**, 393.

Figueiredo, M. and Medini, D. (2011) SCIT, a breakpoint-like distance that simplifies several rearrangement problems. *IEEE/ACM Trans. Comp. Biol. Bioinf.*, **8**, 1316–1329.

Fitch, W. M. (1971) Toward defining the course of evolution: minimum change for a specified tree topology. *Syst. Zool.*, **20**, 406–416.

Goodman, M. et al. (1979) Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globus sequences. *Syst. Zool.*, **28**, 132–163.

Gagneux, M. et al. (2010) Inferring the evolutionary history of gene clusters from phylogenetic and gene order data. *Mol. Biol. Evol.*, **27**, 761–772.

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

Ma, J. et al. (2008) Dupcar: reconstructing contiguous ancestral regions with duplications. *J. Comput. Biol.*, **15**, 1007–1027.

Mullatto, M. et al. (2010) Genomics: a database and a browser to study gene syntenies in modern and ancestral genomes. *Bioinformatics*, **26**, 1119–1121.

Ouangrouna, A. et al. (2011) Reconstructing the architecture of the ancestral amniote genome. *Bioinformatics*, **27**, 2664–2671.

Patio, R. et al. (2011) Parsimonious reconstruction of network evolution. In *Proceedings of WABI’11*, Vol. LNBI 6833, p. 237249.

Pinney, J. W. et al. (2007) Reconstruction of ancestral protein interaction networks for the bzip transcription factors. *Proc. Natl. Acad. Sci. USA*, **104**, 20440–20453.

Rasmussen, M. D. and Kellis, M. (2012) Unified modeling of gene duplication, loss, and coalescence using a locus tree. *Genome Res.*, **22**, 755–765.

Raven, A. et al. (2011) A new, fast algorithm for detecting protein coevolution using maximum compatible cliques. *Algorithmica Med. Biol.*, **6**, 17.

Sankoff, D. (1979) Minimal mutation trees of sequences. *SIAM J. Appl. Math.*, **28**, 35.

Sankoff, D. and El-Mabrouk, N. (2000) Duplication, rearrangement and reconciliation. *In Sankoff, D. and Nadeau, J.H. (eds), Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map alignment and the Evolution of Gene Families, Vol. 1 of Computational Biology*, Kluwer Academic Press, Kluwer Academic publishers Dordrecht/Boston/London, pp. 517–530.

Tang, J. and Wang, L. (2005) Improving genome rearrangement phylogeny using sequence-style parsimony. In *Proceedings 5th IEEE Conference on Bioinformatics and Bioengineering (BIBE 2005)*, pp. 137–144.

Tucker, T. et al. (2010) Reconstructing ancestral gene content by coevolution. *Genome Res.*, **20**, 122–132.

Vilella, A. J. et al. (2009) Ensembl/compare genomes: Complete, duplication-aware phylogenetic trees in vertebrates. *Genome Res.*, **19**, 327–335.

Zheng, C. and Sankoff, D. (2011) On the pathgroups approach to rapid small phylogeny. *BMC Bioinformatics*, **12** (Suppl. 1), S4.