JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 24, Number 6, 2017

Pp. 616–634
DOI: 10.1089/cmb.2017.0065

New Genome Similarity Measures based on Conserved Gene Adjacencies

DANIEL DOERR[1], LUIS ANTONIO B. KOWADA[2], ELOI ARAUJO[3,4], SHACHI DESHPANDE[1,5], SIMONE DANTAS[2], BERNARD M.E. MORET[1], and JENS STOYE[2,4]

ABSTRACT

Many important questions in molecular biology, evolution, and biomedicine can be addressed by comparative genomic approaches. One of the basic tasks when comparing genomes is the definition of measures of similarity (or dissimilarity) between two genomes, for example, to elucidate the phylogenetic relationships between species. The power of different genome comparison methods varies with the underlying formal model of a genome. The simplest models impose the strong restriction that each genome under study must contain the same genes, each in exactly one copy. More realistic models allow several copies of a gene in a genome. One speaks of gene families, and comparative genomic methods that allow this kind of input are called *gene family-based*. The most powerful—but also most complex—models avoid this preprocessing of the input data and instead integrate the family assignment within the comparative analysis. Such methods are called *gene family-free*. In this article, we study an intermediate approach between family-based and family-free genomic similarity measures. Introducing this simpler model, called *gene connections*, we focus on the combinatorial aspects of gene family-free genome comparison. While in most cases, the computational costs to the general family-free case are the same, we also find an instance where the gene connections model has lower complexity. Within the gene connections model, we define three variants of genomic similarity measures that have different expression powers. We give polynomial-time algorithms for two of them, while we show NP-hardness for the third, most powerful one. We also generalize the measures and algorithms to make them more robust against recent local disruptions in gene order. Our theoretical findings are supported by experimental results, proving the applicability and performance of our newly defined similarity measures.

Keywords: family-free genome comparison, gene connections, genome rearrangements, genome similarity measure, conserved adjacencies.

[1]École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.
[2]Universidade Federal Fluminense, Niterói, Brazil.
[3]Universidade Federal de Mato Grosso do Sul, Campo Grande, Brazil.
[4]Faculty of Technology and Center for Biotechnology, Bielefeld University, Bielefeld, Germany.
[5]Department of Computer Science and Engineering, IIT Bombay, Mumbai, India.

1. INTRODUCTION

Many important questions in molecular biology, evolution, and biomedicine can be addressed by comparative genomic approaches. One of the basic tasks when comparing genomes is the definition of measures of similarity between two genomes. Direct applications of such measures are the computation of phylogenetic trees or the reconstruction of ancestral genomes, but also more indirect tasks such as the prediction of orthologous gene pairs (derived from the same ancestor gene through speciation) or the transfer of gene function across species profit immensely from accurate genome comparison methods.

Indeed, over the past 40-or-so years, many methods have been proposed to quantify the similarity of single genes, mostly based on pairwise or multiple sequence alignments. However, in many situations, similarity measures based on whole genomes are more meaningful than gene-based measures, because they give a more representative picture and are more robust against side effects such as horizontal gene transfer. Therefore, in this article, we develop and analyze methods for whole-genome comparison, based on the physical structure (gene order) of the genomes.

The simplest picture of a genome is one where, in a set of genomes under study, orthologous genes have been identified beforehand, and only groups of orthologous genes (also known as *gene families*) are considered that have exactly one member in each genome. In this model, a variety of genomic similarity (or distance) measures have been studied and are relatively easy to compute (Sankoff, 1992; Hannenhalli and Pevzner, 1999; Yancopoulos et al., 2005; Bergeron et al., 2009). However, the singleton gene family is a great oversimplification compared to what we find in nature. Therefore, more general models have been devised where several genes from the same family can exist in one genome. The computation of genomic similarities in these cases is generally much more difficult, though. In fact, many problem variants are NP-hard (Bryant, 2000; Chen et al., 2005; Angibaud et al., 2008; Bulteau and Jiang, 2012; Shao et al., 2015).

Another biological inaccuracy arises from the fact that a gene family assignment is not always without dispute, because orthology is usually not known but just predicted, and most prediction methods require some arbitrary threshold, deciding when two genes belong to the same family and when not. Therefore, *gene family-free* measures have recently been proposed, based on pairwise similarities between genes (Doerr et al., 2012, 2014; Braga et al., 2013; Martinez et al., 2015). While the resulting similarity measures are very promising, their computation is usually not easier than for the family-based models and therefore NP-hard as well (Doerr et al., 2012; Martinez et al., 2015).

In this article, we study an intermediate approach between family-based and family-free genomic similarity measures, *gene connections*. We do this to focus on the combinatorial aspects of gene family-free genome comparison. Our data structure is slightly less complex than in the family-free approach, where arbitrary (real-valued) similarities between genes are considered. This intermediate status allows us to achieve results comparable to those for family-free methods. Whereas in most cases the general family-free model can be recovered by using arbitrary similarities between genes, we also found one case with lower computational complexity for the gene connections model.

The article is structured as follows. We first define three new genome similarity measures based on conserved gene adjacencies (Section 2), followed by some pointers to related literature (Section 3). Each of the three following sections is then devoted to one of the similarity measures. We show that the first problem can be computed in polynomial time, but is biologically quite simplistic. The second one, while avoiding some of the weaknesses of the first, is NP-hard to compute and can therefore not be applied for genomes of realistic size. The third measure, finally, provides a compromise between biological relevance and computational complexity. To demonstrate how our similarity measures can be used in practice, we adapt a method for inferring phylogenetic trees in Section 7. In Section 8, we present experimental results, using a large data set of plant (rosid) genomes. The last section concludes the article.

The implemented algorithms used in this work as well as the studied data set are available for download at http://bibiserv.techfak.uni-bielefeld.de/newdist.

2. BASIC DEFINITIONS

An *alphabet* is a finite set of *characters*. A *string* over an alphabet $\mathcal{A}$ is a sequence of characters from $\mathcal{A}$. Given a string S, $S[i]$ refers to the ith character of S and $|S|$ is the *length* of S, that is, the number of characters in S. In a *signed string* S, each character is labeled with a sign, denoted $sgn_S(i)$ for the character at index position i. A sign is either positive (+) or negative (−). In comparative genomics, for example, the

signs may indicate the orientations of genes on their genomic sequences, which themselves are represented as strings. Therefore, in this article, we use the term *gene* as a synonym for ''signed character'' and the term *genome* as a synonym for ''signed string.''

Definition 1 (gene connection graph). *Given two genomes S and T, a* gene connection graph $G(S, T)$ *of S and T is a bipartite graph with one vertex for each gene of S and one vertex for each gene of T. An edge between two vertices, one from S and one from T, indicates that there is some* connection *between the two genes represented by these vertices.*

The term *connection* in the above definition is not very specific. Depending on the data set and context, connections may be defined based on gene homology, sequence similarity, functional relatedness, or any other similarity measure between genes.

For ease of notation, we let $S[i]$ denote both the ith gene of genome S and the vertex of G representing this gene. Similar for $T[j]$. The set of edges of a graph G is denoted by $E(G)$. The size of a graph G is the number of its edges, $|G| = |E(G)|$. Furthermore, we define a *connection* function t that returns for an index position i of S the list $t(i)$ of index positions in T that are connected to $S[i]$ by an edge in $G(S, T)$. That is, $t(i) = [j|(i, j) \in E(G(S, T))$ for $1 \leq j \leq |T|]$. The function $s(j)$ for an index position of T is defined analogously.

Commonly, a pair of adjacent index positions (i, i') with $i' = i + 1$ in a string is called an *adjacency*. Note that this definition of adjacency only considers direct neighborhood of genes ($i' = i + 1$), while all our following uses of this term refer to an extended definition given by Zhu et al. (2009), who introduced *generalized gene adjacencies* as follows:

Definition 2 (adjacency). *Given an integer $\theta \geq 1$, a pair of index positions (i, i') with $i' \leq i + \theta$ in a string is a (θ-) adjacency.*

In other words, two genes of the same genome form a θ-adjacency if the number of genes between them is less than θ. In the following, we will frequently differentiate between *simple adjacencies* ($\theta = 1$) and *generalized adjacencies* ($\theta \geq 1$).

As mentioned in the Introduction, in this article, we are interested in defining measures of similarity to compare pairs of genomes. A simple approach is based on their number of *conserved adjacencies*. Although below we will study different variants of similarities, they all use the following basic notion of conserved adjacency.

Definition 3 (conserved adjacency). *Given two genomes S and T and a gene connection graph $G(S, T)$, a pair of adjacencies (i, i') in S and (j, j') in T is called a conserved adjacency, denoted $(i, i'||j, j')$, if one of the following two holds:*

(a) $(i, j) \in E(G(S, T))$, $(i', j') \in E(G(S, T))$, $sgn_S(i) = sgn_T(j)$ and $sgn_S(i') = sgn_T(j')$; or
(b) $(i, j') \in E(G(S, T))$, $(i', j) \in E(G(S, T))$, $sgn_S(i) \neq sgn_T(j')$ and $sgn_S(i') \neq sgn_T(j)$.

For an illustration of these definitions, see Figure 1.

We further denote two conserved adjacencies as *conflicting* if their intervals in either genome are overlapping.

Definition 4 (conflicting conserved adjacencies). *Two conserved adjacencies $(i, i'||j, j')$ and $(k, k'||l, l')$ are* conflicting *if (1) $(i, i'||j, j') \neq (k, k'||l, l')$ and (2) $[i, i' - 1] \cap [k, k' - 1] \neq \emptyset$ or $[j, j' - 1] \cap [l, l' - 1] \neq \emptyset$.*

Subsequently, a set of conserved adjacencies is denoted as *nonconflicting* if the above-defined property does not hold between any two of its members.

In the example of Figure 1, (3, 4||4, 6) and (4, 6||5, 7) are the only conflicting conserved adjacencies. All other pairs are nonconflicting.

The different similarity measures that we consider in this work can be derived from the following three problem statements:

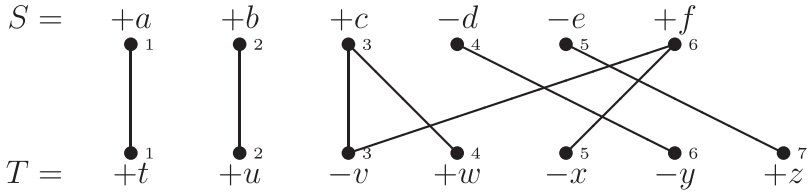

FIG. 1. Genomic data structure. Gene connection graph of two genomes $S = (+a, +b, +c, -d, -e, +f)$ (top row) and $T = (+t, +u, -v, +w, -x, -y, +z)$ (bottom row). Conserved 2-adjacencies are (1, 2||1, 2), (2, 3||2, 4), (3, 4||4, 6), and (5, 6||5, 7). Note that (2, 3||2, 3) and (4, 5||6, 7) are no conserved adjacencies because the signs do not match the definition.

Problem 1 (total adjacency model). *Given two genomes S and T and a gene connection graph $G(S, T)$, discover all pairs of index positions (i, i') in S and (j, j') in T that form a conserved adjacency. In other words, compute* $adj(S, T) = \{(i, i', j, j') | 1 \leq i < i' \leq |S|, 1 \leq j < j' \leq |T| \text{ and } (i, i' || j, j')\}$.

An immediate similarity measure corresponds to the size of the computed adjacency set. Alternatively, one may weight adjacencies according to the similarity of their encompassing genes, as proposed by Doerr et al. (2012), and then compute the sum of their weights. In Section 7, we will further explain how sets of conserved adjacencies between pairs of genomes can be used in inferring phylogenies.

Because a gene connection graph $G(S, T)$ is not limited to one-to-one connections between genes of genomes S and T, some conserved adjacencies found by solving Problem 1 may biologically not be very plausible. Therefore, we define a second problem, motivated by the one used by Doerr et al. (2012) and Braga et al. (2013), which asks for one-to-one correspondences between genes of S and T in its solutions:

Problem 2 (gene matching model). *Given two genomes S and T, a gene connection graph $G(S, T)$, and a real-valued parameter $\alpha \in 0, 1]$, find a bipartite matching M in $G(S, T)$ such that the induced sequences S^M and T^M maximize the objective function*

$$\mathcal{F}_\alpha(M) = \alpha \cdot |adj(S^M, T^M)| + (1 - \alpha) \cdot |M| .$$

(The induced sequences S^M and T^M are the subsequences of S and T, respectively, that contain those characters incident to edges of M.)

As we will see later in this article, solving Problem 2 is NP-hard even for simple adjacencies. Therefore, we define a third, intermediate problem, which is more efficient to solve in practice, while producing one-to-one correspondences between gene extremities. The aim is to find the largest subset of nonconflicting conserved adjacencies found in a pair of genomes:

Problem 3 (adjacency matching model). *Given two genomes S and T and a gene connection graph $G(S, T)$, find a maximum cardinality set of nonconflicting conserved adjacencies $C \subseteq adj(S, T)$.*

3. RELATED WORK

As mentioned above, the *gene connection graph* input format that we propose here is an intermediate between gene families and the family-free model. Indeed, we do not require the gene connection graph to be transitive, which is the main difference to the *gene family graph*, where vertices are assigned to genes and edges are drawn between genes from different genomes whenever they belong to the same family, thus forming bipartite cliques. [This graph has not been introduced under this name in the literature, but is implicitly mentioned already by Sankoff (1999) and later more explicitly by Doerr et al. (2012).] On the other end, the *gene similarity graph* (Braga et al., 2013) is a weighted version of the gene connection graph, increasing the expression power by its ability to represent different strengths of gene connections.

The only previous use of such an intermediate model in comparative genomics that we are aware of is in the form of *indeterminate strings* by Doerr et al. (2014).

Definition 5 (indeterminate string, signed indeterminate string). *Given an alphabet $\mathcal{A}$, a string S over the power set $\mathcal{P}(\mathcal{A}) \setminus \{\emptyset\}$ is called an* indeterminate string *over $\mathcal{A}$. In other words, for $1 \leq i \leq n$, $\emptyset \neq S[i] \subseteq \mathcal{A}$. In a* signed indeterminate string *S, any index position i has a sign $sgn_S(i)$, which therefore is the same for all characters at that position.*

Given two genomes S and T and a gene connection graph $G(S, T)$, it is easy to create a pair of signed indeterminate strings S' and T' over an alphabet $\mathcal{A}'$ that contain the same set of conserved adjacencies as S and T: For any edge $e = (S[i], T[j])$ of $G(S, T)$, create one symbol $e' \in \mathcal{A}'$ and let $e' \in S'[i]$ and $e' \in T'[j]$. The signs are just transferred from S and T to S' and T', respectively: $sgn_{S'}[i] = sgn_S[i]$ for all i, $1 \leq i \leq |S|$, and $sgn_{T'}[j] = sgn_T[j]$ for all j, $1 \leq j \leq |T|$.

Conversely, given two indeterminate strings S' and T', we can easily create sequences S and T and the corresponding gene connection graph with the same set of conserved adjacencies. Let $\mathcal{A} = \{1, 2, \ldots, |S'|, 1', 2', \ldots, |T'|'\}$ set $S = sgn_{S'[1]}1, \ldots, sgn_{S'[|S'|]}|S'|$, $T = sgn_{T'[1]}1', \ldots, sgn_{T'[|T'|]}|T'|'$ and create in $G(S, T)$ an edge $e = (S[i], T[j])$ whenever $S'[i] \cap T'[j] \neq \emptyset$.

Clearly, all the information about conserved adjacencies between these two representations is identical, while sometimes the graph representation and sometimes the representation as signed indeterminate string are more concise.

Indeterminate strings by Doerr et al. (2014) were used to identify regions of common gene content (*gene clusters*) in two genomes, which is important in functional genomics. Here our focus is on conserved adjacencies (which can be seen as small clusters of just two genes) for defining whole-genome similarities. Similar measures are known for singleton gene families as the *breakpoint distance* (Blanchette et al., 1999; Tannier et al., 2009) and have been extended to gene families by Sankoff (1999); Bryant (2000); Angibaud et al. (2008) and were defined for the family-free model by Doerr et al. (2012).

4. AN OPTIMAL SOLUTION FOR PROBLEM 1

To solve Problem 1, we construct a list L of edges of $G(S, T)$ using connection function $t(i)$ for $1 \leq i \leq |S|$. In doing so, we assume that the elements of $t(i)$, $1 \leq i \leq |S|$, are sorted in increasing order. If this is not given as input, it can always be achieved by applying counting sort to all lists $t(i)$ in overall $O(|S|+|T|+|G(S,T)|)$ time, which is proportional to the input size.

We present with Algorithm 1 a solution to Problem 1 for simple adjacencies and subsequently extend this approach for the generalized case. Our algorithm is a simple, linear time procedure that uses three pointers e, e', e'' into list L. These pointers simultaneously traverse L while reporting any pair of adjacent parallel edges (e, e') or crossing edges (e, e'').

Algorithm 1

Input: genomes S and T, gene connection graph $G(S, T)$

Create a list L of all edges $(i, j) \in E(G(S, T))$ ordered by primary index i and secondary index j
Let $e' = (i', j')$ and $e'' = (i'', j'')$ point to the second element of L
for each element $e = (i, j)$ of L in sorted order **do**
  if $sgn_S(i) = sgn_T(j)$ **then**
    while $i' < i+1$ **or** $(i' = i+1$ **and** $j' < j+1)$ **do**
      advance $e' = (i', j')$ by one step in L
    end while
    if $(i', j') = (i+1, j+1)$ **and** $sgn_S(i') = sgn_T(j')$ **then**
      report the conserved adjacency $(i, i' || j, j')$
    end if
  else
    while $i'' < i+1$ **or** $(i'' = i+1$ **and** $j'' < j-1)$ **do**
      advance $e'' = (i'', j'')$ by one step in L
    end while
    if $(i'', j'') = (i+1, j-1)$ **and** $sgn_S(i'') \neq sgn_T(j'')$ **then**
      report the conserved adjacency $(i, i'' || j'', j)$
    end if
  end if
end for

4.1. Correctness

Given a pair $(i, j) \in L$, there are overall four cases for the signs of index i in S and index j in T, each with two subcases for the signs of index $i+1$ in S and index $j+1$ or index $j-1$ in T, listed in the following.

(1) If $sgn_S(i) = +$ and $sgn_T(j) = +$, then we have a conserved adjacency $(i, i+1 || j, j+1)$ if and only if $(i+1, j+1) \in L$ and either $sgn_S(i+1) = +$ and $sgn_T(j+1) = +$ or $sgn_S(i+1) = -$ and $sgn_T(j+1) = -$.
(2) If $sgn_S(i) = +$ and $sgn_T(j) = -$, then we have a conserved adjacency $(i, i+1 || j-1, j)$ if and only if $(i+1, j-1) \in L$ and either $sgn_S(i+1) = +$ and $sgn_T(j-1) = -$ or $sgn_S(i+1) = -$ and $sgn_T(j-1) = +$.
(3) If $sgn_S(i) = -$ and $sgn_T(j) = +$, then we have a conserved adjacency $(i, i+1 || j-1, j)$ if and only if $(i+1, j-1) \in L$ and either $sgn_S(i+1) = -$ and $sgn_T(j-1) = +$ or $sgn_S(i+1) = +$ and $sgn_T(j-1) = -$.
(4) If $sgn_S(i) = -$ and $sgn_T(j) = -$, then we have a conserved adjacency $(i, i+1 || j, j+1)$ if and only if $(i+1, j+1) \in L$ and either $sgn_S(i+1) = -$ and $sgn_T(j+1) = -$ or $sgn_S(i+1) = +$ and $sgn_T(j+1) = +$.

Clearly, cases 1 and 4 and cases 2 and 3 can be summarized to the two cases given in Algorithm 1.

4.2. Runtime analysis

The list L has length $|G(S,T)|$ and can be constructed and sorted in linear time $O(|S|+|T|+|G(S,T)|)$, as discussed above. Each of the three edge pointers e, e', and e'' traverses L once from the beginning to the end, so that the **for** loop in lines 3–19 takes $O(|L|)$ time. Therefore, the overall running time is $O(|S|+|T|+|G(S,T)|)$.

4.3. Space analysis

The algorithm needs space only for the two input strings S and T, the list L, and some constant-space variables. Therefore, the space usage is of order $O(|S|+|T|+|G(S,T)|)$.

4.4. Extension to generalized adjacencies

Algorithm 1′ solves Problem 1 for generalized adjacencies. Following the same strategy as Algorithm 1, the extension requires next to the main pointer e additional 2θ pointers into list L that are denoted e'_t and e''_t, $1 \le t \le \theta$. While it traverses through each element (i,j) in the list using pointer e, each pointer e'_t, $1 \le t \le \theta$, is subsequently increased to point to the smallest element larger than or equal to $(i+t, j+1)$ in L. A copy $\hat{e}$ of pointer e'_t is then used to find candidates $(i+t, j+1), \ldots, (i+t, j+\theta)$. Likewise, pointers e''_t, $1 \le t \le \theta$, are incremented to the smallest element larger than or equal to $(i+t, j-\theta)$, whereupon copy $\hat{e}$ of e''_t is used to find candidates $(i+t, j-\theta), \ldots, (i+t, j-1)$.

Algorithm 1′

Input: genomes S and T, gene connection graph $G(S,T)$, gap threshold θ

1: Create a list L of all edges $(i,j) \in E(G(S,T))$ ordered by primary index i and secondary index j
2: Let $e'_t=(i'_t,j'_t)$ and $e''_t=(i'_t,j'_t)$, $1 \le t \le \theta$, point to the second element of L
3: **for each** element $e=(i,j)$ of L in sorted order **do**
4: **if** $sgn_S(i)=sgn_T(j)$ **then**
5: **for each** $e'_t=(i'_t,j'_t)$, $1 \le t \le \theta$ **do**
6: **while** $i'_t < i+t$ **or** $(i'_t=i+t$ **and** $j'_t < j+1)$ **do**
7: advance $e'_t=(i'_t,j'_t)$ by one step in L
8: **end while**
9: let $\hat{e}=(\hat{\imath},\hat{\jmath}) \leftarrow e'_t$
10: **while** $\hat{\imath}=i+t$ **and** $\hat{\jmath} \le j+\theta$ **do**
11: **if** $sgn_S(\hat{\imath})=sgn_T(\hat{\jmath})$ **then**
12: report the conserved adjacency $(i,\hat{\imath}||j,\hat{\jmath})$
13: **end if**
14: advance $\hat{e}=(\hat{\imath},\hat{\jmath})$ by one step in L
15: **end while**
16: **end for**
17: **else**
18: **for each** $e''_t=(i''_t,j''_t)$, $1 \le t \le \theta$ **do**
19: **while** $i''_t < i+t$ **or** $(i''_t=i+t$ **and** $j''_t < j-\theta)$ **do**
20: advance $e''_t=(i''_t,j''_t)$ by one step in L
21: **end while**
22: let $\hat{e}=(\hat{\imath},\hat{\jmath}) \leftarrow e''_t$
23: **while** $\hat{\imath}=i+t$ **and** $\hat{\jmath} < j-1$ **do**
24: **if** $sgn_S(\hat{\imath}) \neq sgn_T(\hat{\jmath})$ **then**
25: report the conserved adjacency $(i,\hat{\imath}||\hat{\jmath},j)$
26: **end if**
27: advance $\hat{e}=(\hat{\imath},\hat{\jmath})$ by one step in L
28: **end while**
29: **end for**
30: **end if**
31: **end for**

All pointers e, e'_t, and e''_t, $1 \leq t \leq \theta$ are continuously increased, and thus, each traversing L once. Any instance of pointer $\hat{e}$ visits at the most θ elements in each iteration, thus leading to an overall running time of $O(|S|+|T|+\theta^2|G(S,T)|)$. The running time is asymptotically optimal in the sense of worst case analysis, since there can be just as many θ-adjacencies in graph $G(S,T)$. Algorithm $1'$ requires $O(\theta+|S|+|T|+\theta^2|G(S,T)|)$ space.

5. COMPLEXITY OF PROBLEM 2

While one may hope that the intermediate status of the gene connection graph between the gene family graph and the gene similarity graph generally allows more efficient algorithms than for the more complex gene similarity graph, this is not the case for the gene matching model.

Only for $\alpha=0$, we have $\mathcal{F}_\alpha(M)=|M|$ and therefore Problem 2 reduces to computing a maximum bipartite matching, which is possible in polynomial time (Hopcroft and Karp, 1973). However, this case is not very interesting because it completely ignores conserved adjacencies and just compares the gene content of the two genomes. All interesting cases are more difficult to solve, as the following theorem shows[1]:

Theorem 1. *Problem 2 is NP-hard for* $0 < \alpha \leq 1$.

Proof. We will focus on simple adjacencies ($\theta=1$), as this is sufficient to prove Theorem 1. Inspired by the proof of Bryant (2000) for the family-based case, we provide a P-reduction from VERTEX COVER: Given a graph $\mathcal{G}=(V,E)$ and an integer λ, does there exist a subset $V' \subseteq V$ such that $|V'|=\lambda$ and each edge in E is adjacent to at least one vertex in V'?

Our reduction transforms an instance of VERTEX COVER into an instance of the decision version of Problem 2: Given strings S and T, a gene connection graph $G(S,T)$, a real value α, $0 < \alpha \leq 1$, and a real value $F \geq 0$, does there exist a bipartite matching M in $G(S,T)$ such that $\mathcal{F}_\alpha(M) \geq F$?

Let $\mathcal{G}=(V,E)$ and λ be an instance of VERTEX COVER with $V=\{v_1, v_2, \ldots, v_n\}$ and $E=\{e_1, e_2, \ldots, e_m\}$. Then, we construct an alphabet $\mathcal{A}$ of size $2n+4m+2$ given by

$$\mathcal{A}=V \cup \{v'_i | v_i \in V\} \cup E \cup \{e_i{}' | e_i \in E\} \cup \{x_i, x_i{}' | 1 \leq i \leq m+1\}.$$

The two genomes S and T are constructed as follows:

$$S=v_1 v'_1 v_2 v'_2 \ldots v_n v'_n x_1 x'_1 e_1 e'_1 x_2 x'_2 e_2 e'_2 x_3 x'_3 \ldots x_m x'_m e_m e'_m x_{m+1} x'_{m+1},$$

and

$$T=x_{m+1} x'_{m=1} x_m x'_m \ldots x_2 x'_2 x_1 x'_1 v_n \mathcal{E}_n v'_n v_{n-1} \mathcal{E}_{n-1} v'_{n=1} \ldots v_1 \mathcal{E}_\infty v'_1,$$

where $\mathcal{E}_i$ is a string of the symbol pairs $e_j e_j{}'$ for the edges e_j that are adjacent to v_i. The gene connection graph $G(S,T)$ has an edge for each pair of identical symbols $S[i]$ and $T[j]$. The parameter α may be chosen arbitrarily within the range $0 < \alpha \leq 1$.

First, we show that among the matchings maximizing the value $\mathcal{F}_\alpha$ for this problem, there is always at least one that is a maximal matching. Let M be a nonmaximal matching in $G(S,T)$ maximizing $\mathcal{F}_\alpha$ and consider an edge $\ell \notin M$ that may be added to M, forming a new matching $M'=M \cup \{\ell\}$. Clearly, ℓ can dismiss at most two adjacencies of M in M', so $|adj(M')| \geq |adj(M)|-2$. However, in our construction, where the symbols of $\mathcal{A}$ (except the e_i and $e_i{}'$) are in reverse order in S related to T, and furthermore each e_i and each e'_i is between x_i and x_{i+1} in S, any new edge ℓ added to M can dismiss at most one adjacency: If ℓ is adjacent to a symbol a and the symbol a' is adjacent to another edge $\ell' \in M$ (or vice versa) then $|adj(M')|=|adj(M)|+1$. Moreover, if two partner edges $\ell, \ell' \notin M$ are added to M and thus $M'=M \cup \{\ell, \ell'\}$, then $|adj(M')| \geq |adj(M)|$ and $|M'|=|M|+2$. Therefore, $\mathcal{F}_\alpha(M') > \mathcal{F}_\alpha(M)$ for $\alpha < 1$ and $\mathcal{F}_\alpha(M') \geq \mathcal{F}_\alpha(M)$ for $\alpha=1$.

[1]A weaker result, namely the NP-hardness of Problem 2 for values of α between 0 and 1/3, can be found in Doerr (2015).

Next, we show that there is a vertex cover of size λ for a graph $\mathcal{G}$ if and only if Problem 2 has a solution with $F = \alpha(2m+1+(n-\lambda))+(1-\alpha)(2n+4m+2)$. Note that by construction of S, T, and $G(S,T)$, conserved adjacencies in a maximal matching are only possible between pairs of the same symbol of $\mathcal{A}$, that is, $v_i v_i'$, $e_i e_i'$, or $x_i x_i'$. Therefore, we can simplify the notation and represent a conserved adjacency $(i, i'||j, j')$ by the pair of elements in S, $S[i]S[i']$. Clearly, any maximal matching of $G(S,T)$ has $|S| = 2n+4m+2$ edges. Moreover, any maximal matching realizes at least the $2m+1$ conserved adjacencies $e_i e_i'$ and $x_i x_i'$. The other possible conserved adjacencies are the $v_i v_i'$. If there exists a solution with value $F = \alpha(2m+1+(n-\lambda))+(1-\alpha)|S|$, then there are at least $n-\lambda$ conserved adjacencies involving $v_i v_i'$. These adjacencies are possible if the respective edges of $\mathcal{G}$ are covered by λ vertices. If we do not have a solution for F, then $\mathcal{G}$ does not have a vertex cover of size λ. ■

Solving Problem 2 for simple adjacencies, we make use of a method described by Doerr (2015), which was originally developed for solving the gene family-free variant of Problem 2. In doing so, it constructs an *integer linear program* (ILP) similar to program FFAdj-Int described by Doerr et al. (2012). It includes a preprocessing algorithm that identifies small components in gene similarity graphs that are part of an optimal solution. This approach enables the computation of optimal solutions for small- and medium-sized gene similarity graphs. However, as the method is specifically tailored for gene family-free analysis, it does not perform very efficiently on gene connection graphs, as we will see in Section 8. We refer to this ILP and its preprocessing step as Algorithm 2.

We further believe it will be difficult to develop a practical algorithm solving Problem 2 for generalized adjacencies.

6. EXACT SOLUTIONS AND COMPLEXITY OF PROBLEM 3

We present a polynomial time algorithm solving Problem 3 for simple adjacencies, before showing the hardness of the general case. Our algorithm makes use of the following graph structure:

Definition 6 (adjacencies graph). *Given two genomes S and T and a set C of conserved adjacencies between S and T, $C = \{(i_1, i_1'||j_1, j_1'), \ldots, (i_n, i_n'||j_n, j_n')\}$, the* adjacencies graph $A_C(S,T)$ *is a bipartite graph with one vertex for each gene adjacency (i, i') of S and (j, j') of T, respectively. The edges correspond to the conserved adjacencies in C.*

Pseudocode of our algorithm solving Problem 3 for simple adjacencies is shown in Algorithm 3.

Algorithm 3

Input: genomes S and T, gene connection graph $G(S,T)$
1: Let C be the set of conserved adjacencies reported by Algorithm 1 applied to S, T and $G(S,T)$
2: Construct the adjacencies graph $A_C(S,T)$
3: Compute a maximum bipartite matching M on $A_C(S,T)$
4: return M

Clearly its running time is dominated by the time to compute a maximum matching in line 3, which in unweighted bipartite graphs with n vertices and m edges is possible in $O(m\sqrt{n})$ time (Hopcroft and Karp, 1973). In our case, $n = |S|+|T|-2$ and $m \leq n^2$, therefore Algorithm 3 takes overall $O((|S|+|T|)^{5/2})$ time.

6.1. Extension to generalized adjacencies

Other than for the first two problems, the properties of Problem 3 change drastically when generalized adjacencies are considered. Because a θ-adjacency corresponds to an interval of up to $\theta+1$ consecutive genes, the intervals of two θ-adjacencies for $\theta \geq 2$ can overlap on more than two genes, or even be contained in one another. In fact, we will show below that Problem 3 becomes NP-hard in the general case. First, however, we propose an exact solution to the problem; Algorithm 3′ follows the same strategy as its counterpart for simple adjacencies. While for the latter it was possible to find a maximum subset of nonconflicting θ-adjacencies using a maximum matching approach, for the general case we need to resort to an ILP, described in Algorithm 3′.1. It makes use of two types of binary variables, $\mathbf{a}(i,j)$ for each edge (i,j)

in gene connection graph $G(S, T)$, and $\mathbf{b}(i, i', j, j')$ for each θ-adjacency $(i, i'||j, j')$ in the generalized adjacency set C_θ. We say a binary variable is *saturated* if it is assigned value 1. While maximizing the number of saturated $\mathbf{b}(.)$ variables (which represents the output of the program), our ILP imposes matching constraints `(C.01)` for the set of edges in selected θ-adjacencies. Further constraints `(C.02)` ensure that for each θ-adjacency $(i, i'||j, j')$, (1) both edges between its corresponding genes are saturated and (2) no saturated edge is incident to a gene in interval $[i+1, i'-1]$ of genome S (i.e., a possibly empty interval corresponding to all genes between i and i') and interval $[j+1, j'-1]$ of genome T, respectively.

Algorithm 3′

Input: genomes S and T, gene connection graph $G(S, T)$, gap threshold θ

1: Let C_θ be the set of conserved adjacencies reported by Algorithm 1′ applied to S, T, and $G(S, T)$
2: Compute a maximum cardinality set of nonconflicting conserved θ-adjacencies $C_\theta^\star \subseteq C_\theta$ using the ILP given in Algorithm
3: return $C_\theta^\star$

Algorithm 3′.1 ILP solving Step 2 in Algorithm 3′

Objective:
maximize $\sum_{(i, i'||j, j'') \in C_\theta} \mathbf{b}(i, i', j, j')$
Constraints:
(C.01) **for each** $i \leftarrow 1$ **to** $|S|$, $\sum_{j \in t(i)} \mathbf{a}(i, j) \leq 1$
for each $j \leftarrow 1$ **to** $|T|$, $\sum_{i \in s(j)} \mathbf{a}(i, j) \leq 1$
(C.02) **for each** $(i, i'||j, j') \in C_\theta$
if $sgn_S(i) = sgn_S(i')$ **then**
$2 \cdot \mathbf{b}(i, i', j, j') - \mathbf{a}(i, j) - \mathbf{a}(i', j') \leq 0$
otherwise
$2 \cdot \mathbf{b}(i, i', j, j') - \mathbf{a}(i, j') - \mathbf{a}(i', j) \leq 0$
end if
for each $\hat{\imath} \leftarrow [i+1, i'-1]$ **and each** $\hat{\jmath}$ **in** $t(\hat{\imath})$
$\mathbf{b}(i, i', j, j') + \mathbf{a}(\hat{\imath}, \hat{\jmath}) \leq 1$
for each $\hat{\jmath} \leftarrow [j+1, j'-1]$ **and each** $\hat{\imath}$ **in** $s(\hat{\jmath})$
$\mathbf{b}(i, i', j, j') + \mathbf{a}(\hat{\imath}, \hat{\jmath}) \leq 1$
end for
Domains:
(D.01) **for each** $(i, j) \in E(G(S, T))$, $\mathbf{a}(i, j) \in \{0, 1\}$
(D.02) **for each** $(i, i'||j, j') \in C_\theta$, $\mathbf{b}(i, i', j, j') \in \{0, 1\}$

Hardness of Problem 3 for generalized adjacencies. Here we show that the following decision version of Problem 3 is NP-complete:

Problem 4 (S2-AMM). *Let S and T be genomes such that $sgn_S(i) = sgn_T(j)$ for each i, j. Given the gene connection graph $G(S, T)$ and an integer k, determine whether there is a set of nonconflicting conserved adjacencies C for $\theta = 2$ such that $|C| \geq k$.*

Clearly, hardness of Problem 4 implies that solving Problem 3 is NP-hard even if in the genomes all genes have the same sign or if $\theta = 2$.

In what follows, we denote the literals of a Boolean variable V by v and $\neg v$ and we say that the first is *positive* and the second is *negative*, and one is the *opposite* of the other. In a Boolean formula $(\mathcal{V}, \mathcal{C})$, where $\mathcal{V} = \{V_1, \ldots, V_n\}$ is a set of variables and $\mathcal{C}$ is a set of clauses, an *interpretation of* $\mathcal{V}$ is a set of literals $\mathcal{X} = \{x_1, \ldots, x_n\}$ where $x_i \in \{v_i, \neg v_i\}$. We say that $\mathcal{X}$ *satisfies* $\mathcal{C}$ if every clause in $\mathcal{C}$ has a literal in $\mathcal{X}$. The Boolean satisfiability problem (SAT) is to decide if there exists an interpretation for a given Boolean formula.

Like SAT, a restricted version with three variables per clause and where each variable appears at most three times and each literal at most twice in a formula is NP-complete (see Papadimitriou, 2003, p. 183). So, as a direct corollary of this last version, we have that the following problem is also NP-complete.

Problem 5 (2L-SAT). *Given a Boolean formula $(\mathcal{V}, \mathcal{C})$ where each literal appears in at most two clauses, determine whether there is an interpretation of $\mathcal{V}$ that satisfies $\mathcal{C}$.*

This problem, finally, can be reduced to S2-AMM, as shown in the proof of the following theorem.

Theorem 2. *S2-AMM is NP-complete.*

Proof. We denote the concatenation of n copies of a string x by x^n and the concatenation of $x_1, \ldots, x_n$ by $(x_i)_{i=1}^n$.

Given a Boolean formula

$$(\mathcal{V} = \{V^1, \ldots, V^n\}, \mathcal{C} = \{C_1, \ldots, C_m\}),$$

as an instance of 2 L-SAT, we consider an alphabet

$$\mathcal{A} = \mathcal{V} \cup \mathcal{C} \cup \{p_i | i = 0, 1, \ldots, 4(n+m-1)+1\} \cup \{v_1^i, v_2^i, \bar{v}^i, \neg v_1^i, \neg v_2^i | i = 1, \ldots, n\} \cup \{q_i | i = 1, \ldots, 2(n-1)\},$$

and define the following genomes over $\mathcal{A}$ that have only positive genes:

$$S = p_0 \left(V^i P_i\right)_{i=1}^n (C_i P_{n+i})_{i=1}^{m-1} C_m p_{4(n+m-1)+1},$$

where $P_i = \left(p_{4(i-1)+j}\right)_{j=1}^4$ and

$$T = \left(v_1^i v_2^i \bar{v}^i \neg v_1^i \neg v_2^i q_{2i-1} q_{2i}\right)_{i=1}^{n-1} v_1^n v_2^n \bar{v}^n \neg v_1^n \neg v_2^n.$$

The edges in E_S and E_C we describe next are used as connections that define with S and T a gene connection graph $G(S, T)$.

First, we have

$$E_S = \{(p_j, v_1^i), (p_{j+1}, \bar{v}^i), (p_j, \bar{v}^i), (p_{j+1}, \neg v_2^i) | i = 1, \ldots, n \text{ and } j = 4(i-1)\}.$$

The conserved 2-adjacencies induced by edges in E_S are $(p_j, p_{j+1} || v_1^i, \bar{v}^i)$ and $(p_j, p_{j+1} || \bar{v}^i, \neg v_2^i)$ for each i and $j = 4(i-1)$. We say that they *cover* variable V^i: the first covers the literal v^i and the second covers the literal $\neg v^i$. Notice that two different conserved adjacencies covering variables V and V' are nonconflicting if and only if $V \neq V'$.

Second, we describe E_C. For each literal v appearing in clause C_j, let $r = 4(n+j-1)$ and consider all of the cases: (*a*) if v is positive and there is no $k < j$, such that v appears in C_k, add edges (p_r, v_1) and (p_{r+1}, v_2); (*b*) if v is positive and there is $k < j$, such that v appears in C_k, add edges (p_r, v_2) and $(p_{r+1}, \bar{v})$; (*c*) if v is negative and there is no $k < j$, such that v appears in C_k, add edges $(p_r, \bar{v})$ and $(p_{r+1}, \neg v_1)$; (*d*) if v is negative and there is $k < j$, such that v appears in C_k, add edges $(p_r, \neg v_1)$ and $(p_{r+1}, \neg v_2)$. So, for each literal in clause C_j, we have one conserved 2-adjacency $(p_r, p_{r+1} || v_1, v_2)$, $(p_r, p_{r+1} || v_2, \bar{v})$, $(p_r, p_{r+1} || \bar{v}, \neg v_1)$, or $(p_r, p_{r+1} || \neg v_1, \neg v_2)$ depending on the case; and there is no other conserved 2-adjacency induced by edges in E_C. We say that a conserved adjacency induced by the edges added above *covers* clause C_j and, in cases (*a*) and (*b*), we also say that it covers literal v and, in cases (*c*) and (*d*), it covers literal $\neg v$. Notice that if two conserved adjacencies cover different clauses, they are nonconflicting.

Notice that there is no 2-adjacency in S induced by one edge in E_S and one edge in E_C, which implies that there is no conserved 2-adjacency induced by an edge in E_S and an edge in E_C.

Figure 2 shows an example of our construction of a gene connection graph from an instance of 2L-SAT.

Clearly, S, T, and $G(S, T)$ can be obtained from $(\mathcal{V}, \mathcal{C})$ in polynomial time. Now we are going to show that $2L\text{-}SAT(\mathcal{V}, \mathcal{C}) = S2\text{-}AMM(G, n+m)$ to complete the proof, that is, we are going to show that $2L\text{-}SAT(\mathcal{V}, \mathcal{C}) = \text{true}$ if and only if $S2\text{-}AMM(G, n+m) = \texttt{true}$.

Suppose that $2L\text{-}SAT(\mathcal{V}, \mathcal{C}) = \texttt{true}$. Then, there is an interpretation $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ of $\mathcal{V}$ where x_i is a literal of V^i and $\mathcal{X}$ satisfies $\mathcal{C}$. Let $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$ be sets of conserved adjacencies in $G(S, T)$, where a_i covers the variable V^i and the literal opposite to x_i for $i = 1, \ldots, n$, and b_j covers C_j and a literal $x \in \mathcal{X}$ of C_j (whose existence is guaranteed by the hypothesis) for each $j = 1, \ldots, m$. Let $a \neq b \in A \cup B$. We are going to show that a and b are nonconflicting. If a and b are both in A (B), then a and b are nonconflicting conserved adjacencies because a and b cover different variables (clauses). Suppose

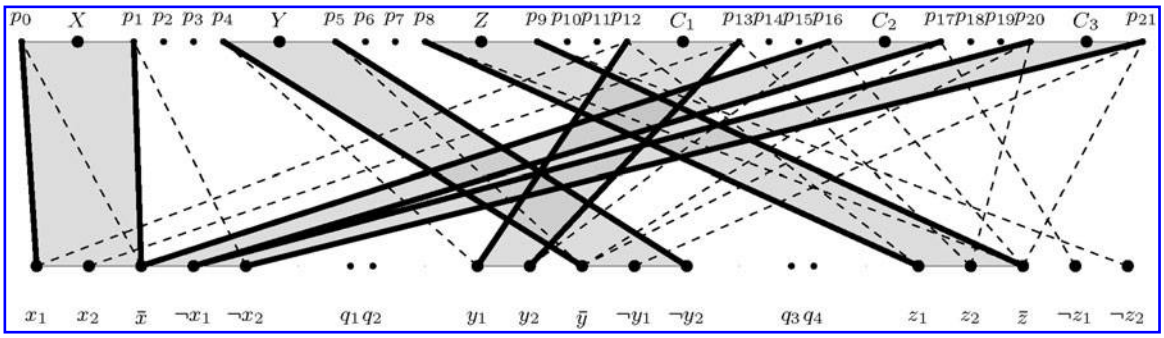

FIG. 2. An example of the reduction scheme. We construct a gene connection graph from the instance of 2L-SAT $(\mathcal{V}=\{X,Y,Z\},\mathcal{C}=\{C_1,C_2,C_3\})$, where $C_1=\{x,y,z\}$, $C_2=\{\neg x,y,\neg z\}$, and $C_3=\{\neg x,\neg y,z\}$. Since 2L-SAT $(\mathcal{V},\mathcal{C})=$true and $|\mathcal{V}|=3$ and $|\mathcal{C}|=3$, we can find $3+3=6$ nonconflicting conserved 2-adjacencies that are marked as shaded polygons.

that $a\in A$ and $b\in B$. Since a covers a variable and b covers a clause, a and b cover disjoint intervals in S. If a and b cover literals from different variables, they also cover disjoint literals in T, which implies that a and b are nonconflicting. So, we assume that a and b cover the literals from the same variable. However, in this case, by construction, a and b cover opposite literals, which implies that even in this case a and b are nonconflicting. Since $A\cup B$ is a set of nonconflicting conserved adjacencies and $|A\cup B|=n+m$, we have that $S2-AMM(G,n+m)=\texttt{true}$.

Suppose that $S2-AMM(G,n+m)=\texttt{true}$. Let R be a set of $n+m$ nonconflicting conserved adjacencies. Since $|V|=n$, $|C|=m$, $|R|=n+m$, two nonconflicting conserved adjacencies do not cover the same variable and the same clause, and since all conserved adjacencies cover a variable or a clause, it follows that all variables and clauses are covered by conserved adjacencies from R and each is covered only once. Let $\mathcal{X}$ be a set of literals such that $x\in\mathcal{X}$ if the opposite to x is covered by a conserved adjacency that covers a variable in $\mathcal{V}$. Since all variables are covered by conserved adjacencies from R and they are covered only once, it follows that $\mathcal{X}$ is an interpretation of $\mathcal{V}$. Let $C\in\mathcal{C}$. Since all clauses are covered by conserved adjacencies from R, it follows that there is an $a\in R$, such that a covers C. Let l be a literal covered by a and V its corresponding variable. Let $l'\in\mathcal{X}$ such that l' is a literal of V. There is $b\in R$ such that b covers V and the opposite to l'. Since a,b are nonconflicting, we have that l and the opposite to l' are different. Since l and l' are both literals from V, it follows that $l=l'$. Since there is a literal $l\in C$ such that $l\in\mathcal{X}$ for each $C\in\mathcal{C}$, we have that $\mathcal{X}$ satisfies $\mathcal{C}$. Therefore, $2L-SAT(\mathcal{V},\mathcal{C})=\texttt{true}$. ■

7. INFERRING PHYLOGENIES

We now describe an approach for reconstructing evolutionary trees based on solutions to the three problems discussed above. To this end, we adapt a gene family-based approach for reconstructing phylogenies with gene order data described by Lin et al. (2013). The aim is to obtain a *maximum likelihood* (ML) tree subject to a simple Markov model explaining commonalities and differences in the gene order sequences of our genomic data set. The actual inference is made using RAxML (Stamatakis, 2006), a popular tool for reconstructing phylogenies with ML.

In constructing the model, Lin et al. decompose gene order sequences into two binary characteristics, the presence of adjacencies and occurrence of gene families in each genome of the data set. Hence, the genomic data set is represented by a 0/1 matrix, in which each row corresponds to a distinct genome. There are two types of columns arranged in separate sections of the matrix. The first part is associated with adjacencies, the remainder with gene content. Henceforth, we will refer to *adjacency columns* and *gene content columns*, respectively. As Lin et al. rely on gene family assignments, each gene is represented by its gene family identifier rather than by its index position. Thus, each adjacency column corresponds to an adjacency between any two genes of certain relative orientation associated with a particular pair of gene families.

A column in the 0/1 matrix is *informative* if its entries are neither all 1s nor all 0s. For constructing an ML tree, RAxML relies entirely on informative columns, and hence, noninformative columns will be discarded. Clearly, for a data set with n gene families, there are up to n gene content columns and up to n^2 adjacency columns, through which each row will have up to n 1s in the gene content and adjacency columns, respectively.

In the following we describe how sets of conserved adjacencies in the gene connections model can be used to construct 0/1 matrices similar to Lin et al.'s. We show how adjacency sets translate into adjacency columns and how gene connections give rise to gene content columns. We relate between genes and adjacencies across all genomes in the data set using the connected components of the graphs defined in the following.

Definition 7 (joint gene connection graph; joint adjacencies graph). *The joint gene connection graph $G(\mathcal{S})=(V,E)$ of a set of m genome sequences $\mathcal{S}=\{S_1,\ldots,S_m\}$ is an m-partite graph, where each position i, $1\leq i\leq |S_k|$, for each genome S_k, $1\leq k\leq m$, is associated with a vertex $(k,i)\in V$. Any two vertices $(k,i),(l,j)\in V$, $k\neq l$, share an edge in E if their corresponding genes $S_k[i]$ and $S_l[j]$ have some connection.*

Similarly, for a given set of m genome sequences $\mathcal{S}=\{S_1,\ldots,S_m\}$ and any (sub-) set of pairwise conserved adjacency sets $\mathcal{C}=\{C_{k,l}|1\leq k<l\leq m\}$, the joint adjacencies graph $A_{\mathcal{C}}(\mathcal{S})=(V,E)$ *is an m-partite graph where each adjacency (i,i') in a genome S_k, $1\leq k\leq m$, is associated with a vertex $(k,i,j)\in V$. Edge set E corresponds to conserved adjacencies $(i,i'||j,j')\in C_{k,l}$, where $C_{k,l}\in\mathcal{C}$ and $1\leq k<l\leq m$.*

Gene content columns are generated from connected components of the joint gene connection graph over all genomes of the data set, or from its subgraph, as for Problem 2 or Problem 3. Each connected component then corresponds to one or more columns in the 0/1 matrix, according to the maximum number of its genes associated with the same genome. These columns are filled from left to right with as many 1s as there are genes associated with each genome within the connected component. The outcome of Problem 2 is a gene matching that directly translates into a subgraph of the joint gene connection graph suitable for generating gene content columns.

The adjacency matching that is outcome of Problem 3 requires more elaborate treatment to generate gene content columns of a 0/1 matrix. Clearly, genes that are part of matched adjacencies induce a subgraph of the joint gene connection graph whose connected components can again be encoded into gene content columns. Yet, if gene orders are sufficiently perturbed, many genes will not participate in conserved adjacencies. Rather than ignoring possibly large parts of the joint gene connection graph in constructing gene content columns, we define additional columns from connected components of a further subgraph induced by the set of genes that are not contained in matched adjacencies.

For Problem 1, adjacency columns are generated from connected components of the joint adjacencies graph $A_{\mathcal{C}}(\mathcal{S})$ over the set of pairwise conserved adjacencies $\mathcal{C}=\{adj(S_k,S_l)|\{S_k,S_l\}\subseteq\mathcal{S}\}$. Conversely, for Problem 2 and Problem 3, only the subsets of conserved adjacencies are considered that are also part of their corresponding matchings.

We then follow the approach by Lin et al. of adjusting the transition probabilities of the two-state time reversible model that is subject to RAxML's ML tree inference.

7.1. Extension to general gene family-free model

We proceed to outline an approach for the general gene family-free model that we will later use (next to the approach described above) for reconstructing the phylogeny of the rosid data set. The general family-free model allows for arbitrary similarities between genes that are given as strictly positive edge weights in a gene similarity graph. In addition, we define a weighted adjacencies graph whose edge weights correspond to adjacency scores as described by Doerr et al. (2012). We then rely on a set of user-defined threshold values. For each threshold value, we prune the joint adjacencies graph and the joint gene similarity graph, respectively, by removing edges that fall below the threshold. The connected components of each of these graphs give rise to columns in the 0/1 matrix, as described above. Thus, with increasing threshold value, connected components split up into smaller components. The sequence of threshold values leads to a series of pruned graphs. This results in a redundant encoding of gene order information in the 0/1 matrix where connected components with high weights are associated with more columns than faint ones.

Table 1. Biological Data

Species	*Version*	*No. of genes*	*No. of scaffolds*	*References*
Arabidopsis thaliana	TAIR10	27,416	7	Lamesh et al., 2011
Brassica rapa	FPSc v1.3	40,492	669	Goodstein et al., 2012
Boechera stricta	v1.2	27,416	854	Goodstein et al., 2012
Citrus clementina	v1.0	24,533	94	Wu et al., 2014
Capsella rubella	v1.0	26,521	123	Slotte et al., 2013
Eucalyptus grandis	v1.1	36,376	1315	Bartholomé et al., 2015
Eutrema salsugineum	v1.0	26,351	61	Yang et al., 2013
Fragaria vesca	v1.1	32,831	8	Shulacv et al., 2011
Glycine max	Wm82.a2	56,044	147	Schmutz et al., 2010
Gossypium raimondii	v2.1	37,505	133	Paterson et al., 2012
Linum usitatissimum	v1.0	43,471	1028	Wang et al., 2012
Medicago truncatula	Mt4.0v1	50,894	1033	Young et al., 2011
Prunus persica	v1.0	27,864	59	Verde et al., 2013
Populas trichocarpa	v3.0	41,335	379	Du et al., 2015
Phaseolus vulgaris	v1.0	27,197	91	Schmutz et al., 2014
Ricinus communis	v0.1	31,221	4962	Chan et al., 2010
Theobroma cacao	v1.1	29,452	99	Motamayor et al., 2012
Vitis vinifera	Genoscope.12X	26,346	33	Jaillon et al., 2007

The genomic data set of 18 rosid species used in subsequently described experiments.

8. EXPERIMENTAL RESULTS

8.1. Genomic data set

We study genomes of 18 rosid species listed in Table 1. Rosids are a prominent subclass of flowering plants to which also many agricultural crops belong. The genomic sequences of the studied species were obtained from *Phytozyme* (Goodstein et al., 2012)[2], an online resource of the Joint Genome Institute providing databases and tools for comparative genomics analyses of plant genomes. Most of the studied plant genomes are partially assembled, comprising up to 5000 scaffolds covering one or more annotated protein coding genes. While the smallest genome in our data set contains roughly 24,500 genes, the largest spans with 56,000 genes more than twice as many. Rosids, just like many other plants, met their evolutionary fate through multiple events of whole-genome duplication, followed by periods of fractionation, in which many duplicated genes were lost again.

8.2. Construction of gene connection and gene family graphs

Next to the genomic sequences and gene annotations, Phytozyme also provides gene family information in the form of co-orthologous clusters computed by InParanoid (Sonnhammer and Östlund, 2015). InParanoid follows a seed-based strategy by identifying pairs of orthologous genes (the ''seeds'') through reciprocal best pairwise local alignment hits computed by the BLASTP program (Camacho et al., 2008). These are subsequently used to recruit inparalogs, eventually forming groups of co-orthologous genes.

We ran BLASTP on all genes of our data set using an e-value threshold of 10^{-5} and otherwise default parameter settings. We then constructed gene connection graphs for all 153 genome pairs by establishing edges between vertices whose corresponding genes share reciprocal BLASTP hits. We refer to these graphs as *BLASTP GC graphs*. Similarly, we constructed pairwise gene family graphs using InParanoid's homology assignment, which we refer to as *InParanoid GF graphs*.

Unsurprisingly, the BLASTP GC graphs are much larger in size than the InParanoid GF graphs. We observed average sizes of 150,000 edges for the former, whereas the latter graphs had on average of only one-fifth of this size. Moreover, only 4% of edges in InParanoid GF graphs were not contained in their BLASTP GC counterparts. Lacking ground truth of homologies in our data set, we take a conservative stance by assuming that InParanoid's homology assignment can be considered true, or, in other words, that it contains

[2]The described experiments were performed on data sets of Phytozyme v10.3.

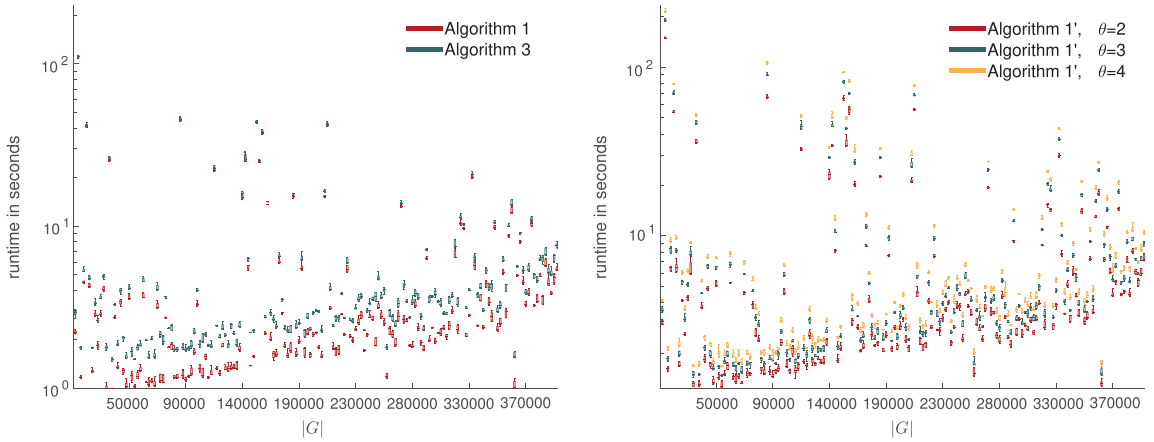

FIG. 3. Runtime benchmarks. Left: Runtimes of Algorithms 1 and 3 for all 153 BLASTP GC graphs of the studied data set. Right: Runtimes of Algorithm 1′ for $\theta = 2, 3, 4$.

only a negligible number of false positives. However, we conclude from a previous study by Lechner et al. (2014), in which InParanoid (as well as all other gene family prediction tools in that study) exhibited a poor recall, that the homology assignment may be incomplete. That being said, we regard the edges of BLASTP GC graphs with suspicion. In doing so, we assume many of them leading to false-positive homology assignments. We perform subsequent analysis to outline a possible procedure of identifying additional potential homologies that are supported by conservation in gene order in BLASTP GC graphs.

8.3. *Implementation*

We implemented Algorithms 1, 1′, 3, and 3′ in Python. For Algorithm 2 we used the implementation of Doerr (2015). In Algorithm 3, the maximum cardinality matching was computed using an implementation of Hopcroft and Karp's algorithm (Hopcroft and Karp, 1973) provided by the Python-based NetworkX[3] library. The ILPs of Algorithms 2 and 3′ were run using CPLEX,[4] a solver for various types of linear and quadratic programs. All computations were performed on a Linux machine using a single 2.3 GHz CPU.

8.4. *Running times*

The runtimes of Algorithms 1 and 3 are shown in Figure 3 (left). The runtime analysis was repeated five times and is visualized by whisker plots. For each of the 153 BLASTP GC graphs in our data set, the computation was finished in less than 50 CPU seconds. Moreover, our evaluation reveals that the enumeration of the set of conserved adjacencies in our data set requires often more time than the subsequent computation of the maximum matching for Algorithm 3. The plot on the right side of Figure 3 shows that the runtimes of Algorithm 1′ for $\theta = 2, 3, 4$ increase only moderately for higher values of θ.

Comparing our methods to the gene family-free approach, an implementation of a heuristic method described by Doerr et al. (2012) failed to return a result for the gene family-free variant of Problem 2 on the BLASTP GC graph of *R. communis* and *V. vinifera* within 36 hours of computation. Surprisingly, running Algorithm 2 with $\alpha = 0.1$ just as long, we were able to obtain a suboptimal solution of which CPLEX reported an optimality gap of only 1.89%. Nevertheless, as a reference for comparison with our various models, it would be even more informative to have optimal solutions of these problems. We leave it as an open problem whether it is possible to improve our ILPs to achieve this.

Furthermore, we were able to compute exact results for Problem 3 and $\theta = 2$ with Algorithm 3′ for all 153 but 19 BLASTP GC graphs and for all but 16 InParanoid GC graphs, limiting computation time to 2 hours per graph instance.

[3]http://networkx.github.io
[4]www.ibm.com/software/integration/optimization/cplex-optimizer

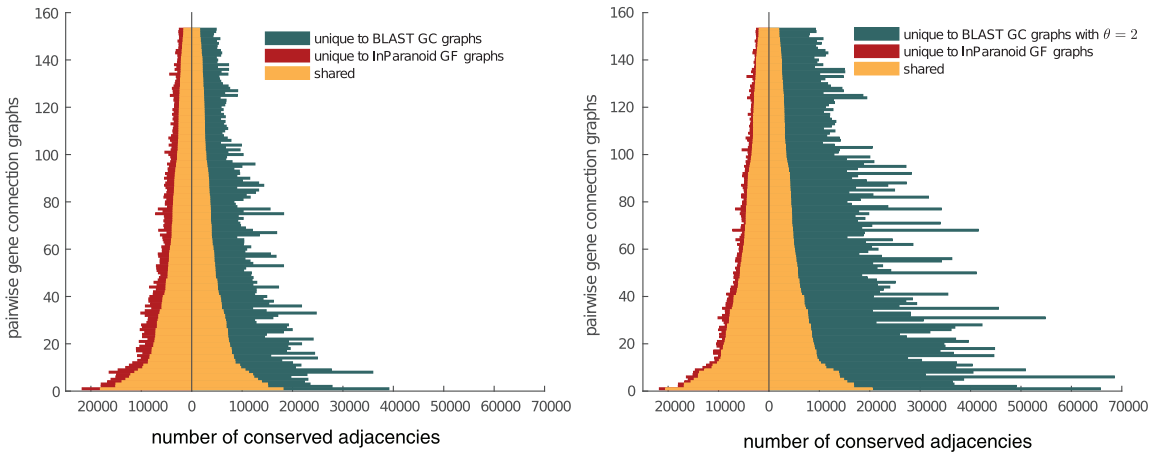

FIG. 4. Comparison of adjacencies. Overlap of conserved adjacencies between BLASTP GC and InParanoid GF graphs.

8.5. *Gene connection versus gene family graphs*

The overlap between sets of conserved adjacencies identified in BLASTP GC graphs and in InParanoid GF graphs is visualized in Figure 4 for simple and 2-adjacencies, respectively. Overall, 70% of the conserved simple adjacencies of the InParanoid GF graphs were also found in the BLASTP GC graphs, whereas we find in the latter 90% more conserved adjacencies than in the former. Investigating the high number of InParanoid adjacencies that are missing in BLASTP GC graphs, we discovered that many generalized adjacencies of the former span genes that are connected (and therefore breaking the surrounding adjacency) in their BLASTP GC counterparts. However, the mean number of connected intervening genes was only 1.4. In fact, the overlap of 2-adjacencies in BLASTP GC graphs with 1-adjacencies of InParanoid GF graph was at 83%.

Finally, Figure 5 visualizes the number of nonconflicting conserved adjacencies in BLASTP GC and InParanoid GF graphs computed for $\theta = 1$ using Algorithm 3 (left plot) and computed for $\theta = 2$ using Algorithm 3′ (right plot). For the former, we observed on average 42% more nonconflicting conserved adjacencies in BLASTP GC graphs when compared to their InParanoid GF counterparts, whereas for the latter, this number dropped to 32%. Nevertheless, from $\theta = 1$ to $\theta = 2$, the absolute number of nonconflicting conserved adjacencies increases on average by 27% for BLASTP GC graphs and by 37% for InParanoid GF graphs, respectively.

8.6. *Reconstructing the rosid phylogeny*

We inferred the phylogeny of our rosid data set using the approach described in Section 7. To this end, we constructed a 0/1 matrix whose adjacency columns are based on an adjacency matching solving Problem

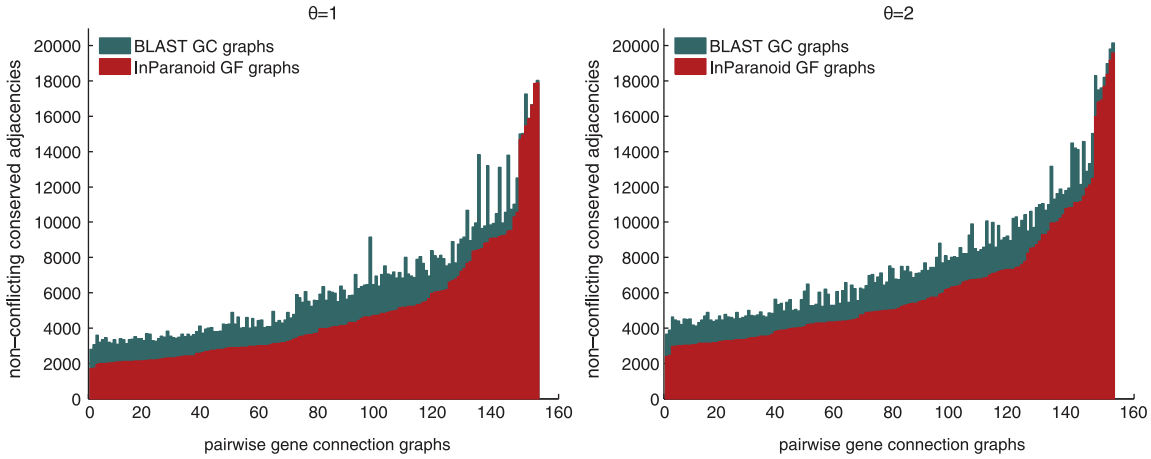

FIG. 5. Comparison of nonconflicting adjacencies. Numbers of nonconflicting conserved adjacencies in BLASTP GC and InParanoid GF graphs for $\theta = 1$ (left) and $\theta = 2$ (right).

3 for $\theta = 1$. The gene content columns of the matrix were generated from connected components of the joint gene connection graph of our data set. This matrix was then input to RAxML, which computed the ML tree shown in Figure 6. We configured RAxML to calculate bootstrap values with 500 replicates. The bootstrap percentages are shown along the branches.

We used RAxML to infer a further ML tree from our data set, this time using a 0/1 matrix that was generated with the general family-free approach. In doing so, we applied a similarity measure between genes called *Relative reciprocal BLAST score* (Pesquita et al., 2008). Just as in our previous attempt, we used adjacencies from a matching solving Problem 3 with $\theta = 1$. These were then scored as described by Doerr et al. (2012). We used threshold values 0, 0.2, 0.6, 0.8, 0.95, and 0.999 to successively prune both the joint (weighted) adjacency graph and the joint gene similarity graph and encode the connected components for each threshold value in the 0/1 matrix. The tree computed by RAxML is shown in Figure 7.

In both phylogenies, the Brassicales (*A. thaliana*, *B. stricta*, *B. rapa*, *C. rubella*, and *E. salsugineum*), Fabales (*M. truncatula*, *P. vulgaris*, and *G. max*), and Rosales (*F. vesca* and *P. persica*) cluster not only together but also their internal branching is consistent with the phylogenetic tree provided by Phytozyme.

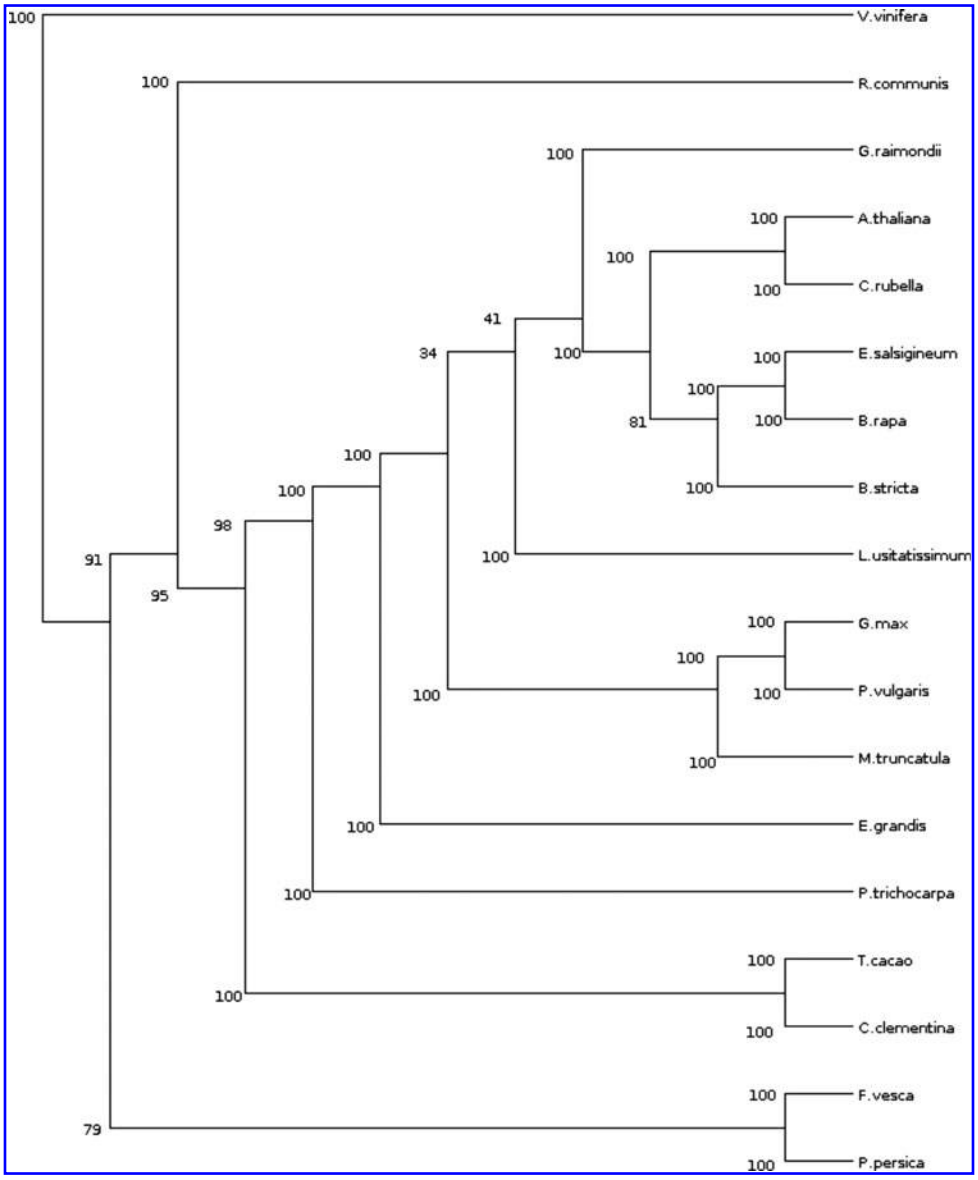

FIG. 6. Phylogeny of 18 rosid species based on matching data. The phylogenetic tree was computed by RAxML from the outcome of the adjacency matching problem for $\theta = 1$.

However, in Figure 6, the three Malpighiales (*M. trichocarpa*, *L. usitatissimum*, and *R. communis*) and two Malvales (*G. raimondii* and *T. cacao*) do not group together at all. The phylogeny of Figure 7 is a more accurate reconstruction: Two out of three Malpighiales are clustered together and also the Malvales are closer together. Moreover, the bootstrap values are better than in Figure 6 in the sense that none is below 75.

Still, in both approaches we do not see a consistent branching between different genera. This might be attributed to the whole-genome duplication events that occurred at the base of Brassicales and Fabales, but the actual cause remains unclear and demands further investigation.

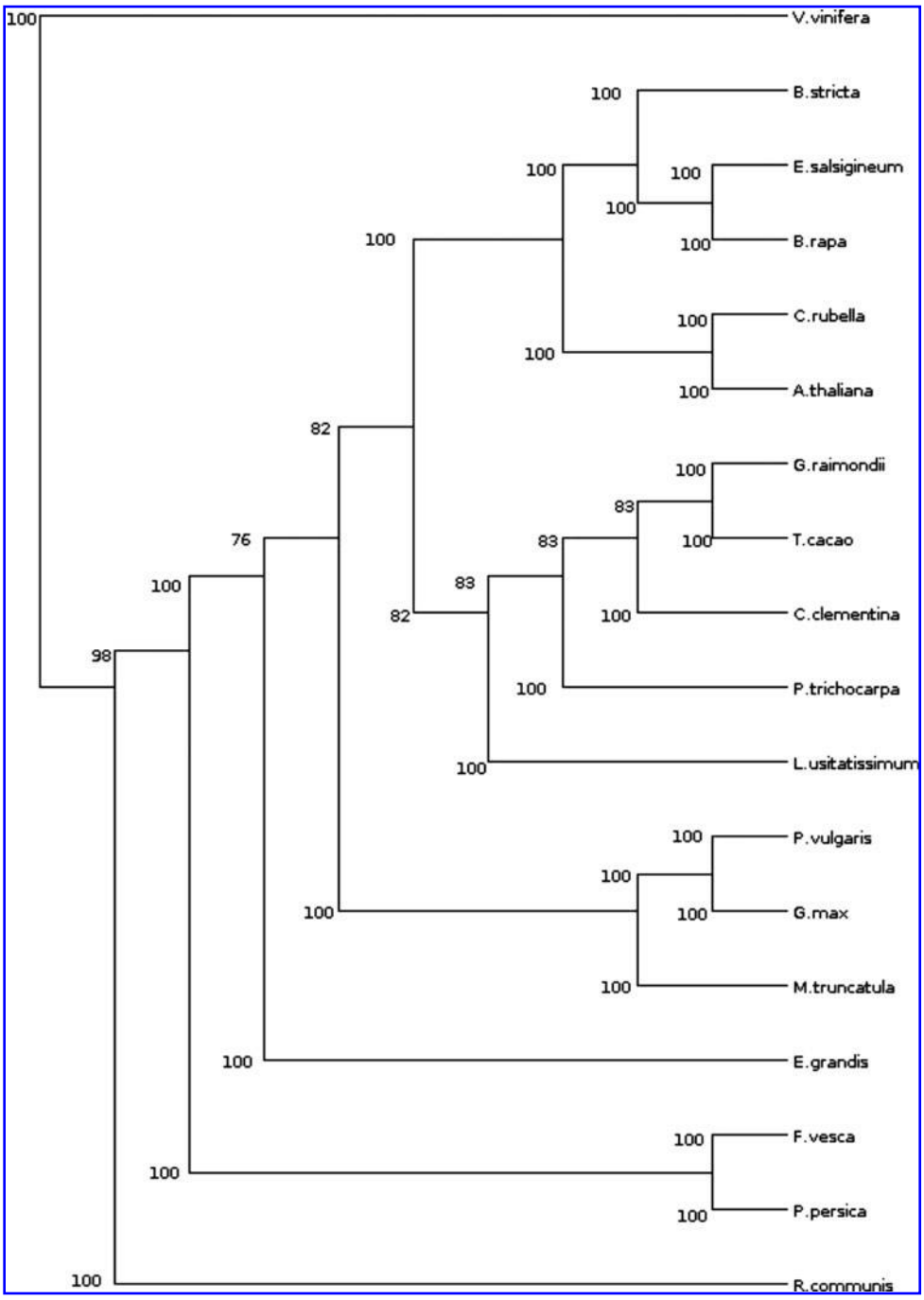

FIG. 7. Phylogeny of 18 rosid species using the family-free approach. The phylogenetic tree was computed by RAxML using the general family-free approach on the outcome of the adjacency matching problem for $\theta = 1$.

9. CONCLUSION

We have presented new similarity measures for complete genomes, thereby defining gene connections as an intermediate model of genome similarity representations, between gene families and the gene family-free approach. Our theoretical results with some problem variants being polynomial and others being NP-hard show that we are very close to the hardness border of similarity computations between genomes with unrestricted gene content. On the practical side we could show that the computation of genomic similarities and reconstruction of phylogenies in the gene connection model gives meaningful results and is possible in reasonable time, if the measures and algorithms are designed carefully.

A few questions remain open, though. Since it is always difficult to choose optimal values for parameters such as the gap threshold θ, it will be worthwhile to examine whether parameter estimation methods for generalized adjacencies as the ones developed by Yang and Sankoff (2010) can be adapted to the gene connection model.

Various model extensions can also be envisaged. The adjacency matching model (Problem 3) removes inconsistencies from the output of the total adjacencies model (Problem 1) by computing a maximum matching on it. It could be tested whether other criteria to remove genes from the genomes and thus derive consistent sets of conserved adjacencies yield even better results. Moreover, the resulting reduced genomes with conserved adjacencies could also be used to predict orthologies between the involved genes, not just to compute genomic similarities.

An alternative objective function for our problem formulations, instead of counting (generalized) gene adjacencies, could be a variant of the *summed adjacency disruption number* (Delgado et al., 2010) that also allows to distinguish between small and larger distortions in gene order.

Finally, Algorithm 3 can easily be generalized for weighted gene similarities (instead of gene connections). It remains to be evaluated if such a more fine-grained measure in the spirit of a family-free analysis has advantages compared to the unit-cost measures studied in this article.

ACKNOWLEDGMENTS

L.A.B.K. and S.D. are partially supported by Brazilian research agencies FAPERJ and CNPq. E.A. is funded by a Brazilian research agency CNPq grant Ciência sem Fronteiras Postdoctoral Scholarship 234234/2014-8. This work was performed while J.S. was on sabbatical at UFF in Niterói, Brazil, funded by Brazilian research agency CAPES grant Ciência sem Fronteiras Special Visiting Researcher.

AUTHOR DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Angibaud, S., Fertin, G., Rusu, I., et al. 2008. Efficient tools for computing the number of breakpoints and the number of adjacencies between two genomes with duplicate genes. *J. Comp. Biol.* 15, 1093–1115.

Bergeron, A., Mixtacki, J., and Stoye, J. 2009. A new linear time algorithm to compute the genomic distance via the double cut and join distance. *Theor. Comput. Sci.* 410, 5300–5316.

Blanchette, M., Kunisawa, T., and Sankoff, D. 1999. Gene order breakpoint evidence in animal mitochondrial phylogeny. *J. Mol. Evol.* 49, 193–203.

Braga, M.D.V., Chauve, C., Doerr, D., et al. 2013. The potential of family-free genome comparison, 63–81. *In* Chauve, C., El-Mabrouk, N., and Tannier, E., eds., *Models and Algorithms for Genome Evolution*, volume 19 of *Computational Biology Series*. Springer Verlag, Berlin.

Bryant, D. 2000. The complexity of calculating exemplar distances, 207–211. *In* Sankoff, D., and Nadeau, J.H., eds., *Comparative Genomics*, volume 1 of *Computational Biology Series*. Kluwer Academic Publishers, London.

Bulteau, L., and Jiang, M. 2012. Inapproximability of (1,2)-exemplar distance. *IEEE/ACM Trans. Comput. Biol. Bioinf.* 10, 1384–1390.

Camacho, C., Coulouris, G., Avagyan, V., et al. (2008). BLAST+: Architecture and applications. *BMC Bioinfo*. 10, 421.

Chen, X., Zheng, J., Fu, Z., et al. 2005. Assignment of orthologous genes via genome rearrangement. *IEEE/ACM Trans. Comput. Biol. Bioinf.* 2, 302–315.

Delgado, J., Lynce, I., and Manquinho, V. 2010. Computing the summed adjacency disruption number between two genomes with duplicate genes. *J. Comp. Biol.* 17, 1243–1265.

Doerr, D. 2015. Gene family-free genome comparison [Ph.D. thesis], Faculty of Technology, Bielefeld University, Germany. Available https://pub.uni-bielefeld.de/publication/2902049 Last viewed:5/29/17.

Doerr, D., Stoye, J., Böcker, S., et al. 2014. Identifying gene clusters by discovering common intervals in indeterminate strings. *BMC Bioinformatics* 15(Suppl. 6), S2.

Doerr, D., Thévenin, A., and Stoye, J. 2012. Gene family assignment-free comparative genomics. *BMC Bioinformatics* 13(Suppl. 19), S3.

Goodstein, D.M., Shu, S., Howson, R., et al. 2012. Phytozome: A comparative platform for green plant genomics. *Nucleic Acids Res.* 40(Database issue), D1178–D1186.

Hannenhalli, S., and Pevzner, P.A. 1999. Transforming cabbage into turnip: Polynomial algorithm for sorting signed permutations by reversals. *J. ACM* 46, 1–27.

Hopcroft, J.E., and Karp, R.M. 1973. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput*. 2, 225–231.

Lechner, M., Hernandez-Rosales, M., Doerr, D., et al. 2014. Orthology detection combining clustering and synteny for very large datasets. *PLoS One* 9, e10515.

Lin, Y., Hu, F., Tang, J., et al. 2013. Maximum likelihood phylogenetic reconstruction from high-resolution whole-genome data and a tree of 68 eukaryotes. In *Proceedings of PSB 2013*, 285–296. Publisher: World Scientific.

Martinez, F.V., Feijão, P., Braga, M.D.V., et al. 2015. On the family-free DCJ distance and similarity. *Algorithms Mol. Biol.* 10, 13.

Papadimitriou, C.H. 2003. *Computational Complexity*. John Wiley and Sons Ltd., Chichester, UK.

Pesquita, C., Faria, D., Bastos H., et al. 2008. Metrics for GO based protein semantic similarity: A systematic evaluation. *BMC Bioinformatics* 9(Suppl 5), S4.

Sankoff, D. 1992. Edit distance for genome comparison based on non-local operations, 121–135. *In* Apostolico, A., Crochemore, M., Galil, Z., and Manber, U., eds., *Proceedings of CPM 1992*, volume 644 of *LNCS*. Springer Verlag, Berlin. .

Sankoff, D. 1999. Genome rearrangement with gene families. *Bioinformatics* 15, 909–917.

Shao, M., Lin, Y., and Moret, B.M.E. 2015. An exact algorithm to compute the double-cut-and-join distance for genomes with duplicate genes. *J. Comp. Biol.* 22, 425–435.

Sonnhammer, E.L.L., and Östlund, G. 2015. Inparanoid 8: Orthology analysis between 273 proteomes, mostly eukaryotic. *Nucleic Acids Res.* 43(Database issue), D234–D239.

Stamatakis, A. 2006. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics* 22, 2688–2690.

Tannier, E., Zheng, C., and Sankoff, D. 2009. Multichromosomal median and halving problems under different genomic distances. *BMC Bioinformatics* 10, 120.

Yancopoulos, S., Attie, O., and Friedberg, R. 2005. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics* 21, 3340–3346.

Yang, Z., and Sankoff, D. 2010. Natural parameter values for generalized gene adjacencies. *J. Comp. Biol.* 17, 1113–1128.

Zhu, Q., Adam, Z., Choi, V, et al. 2009. Generalized gene adjacencies, graph bandwidth, and clusters in yeast evolution. *IEEE/ACM Trans. Comput. Biol. Bioinf.* 6, 213–220.

Address correspondence to:
Prof. Jens Stoye
Universität Bielefeld
Technische Fakultät
Genominformatik, Universitätsstr. 25
33615 Bielefeld
Germany

E-mail: jens.stoye@uni-bielefeld.de