JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 17, Number 9, 2010
© Mary Ann Liebert, Inc.
Pp. 1327–1344
DOI: 10.1089/cmb.2010.0112

Reconstructing Ancestral Genomic Sequences by Co-Evolution: Formal Definitions, Computational Issues, and Biological Examples

*TAMIR TULLER,[1,2] *HADAS BIRIN,[3] MARTIN KUPIEC,[4] and EYTAN RUPPIN[3,5]

ABSTRACT

The inference of ancestral genomes is a fundamental problem in molecular evolution. Due to the statistical nature of this problem, the most likely or the most parsimonious ancestral genomes usually include considerable error rates. In general, these errors cannot be abolished by utilizing more exhaustive computational approaches, by using longer genomic sequences, or by analyzing more taxa. In recent studies, we showed that co-evolution is an important force that can be used for significantly improving the inference of ancestral genome content. In this work we formally define a computational problem for the inference of ancestral genome content by co-evolution. We show that this problem is NP-hard and hard to approximate and present both a Fixed Parameter Tractable (FPT) algorithm, and heuristic approximation algorithms for solving it. The running time of these algorithms on simulated inputs with hundreds of protein families and hundreds of co-evolutionary relations was fast (up to four minutes) and it achieved an approximation ratio of <1.3. We use our approach to study the ancestral genome content of the Fungi. To this end, we implement our approach on a dataset of 33, 931 protein families and 20, 317 co-evolutionary relations. Our algorithm added and removed hundreds of proteins from the ancestral genomes inferred by maximum likelihood (ML) or maximum parsimony (MP) while slightly affecting the likelihood/parsimony score of the results. A biological analysis revealed various pieces of evidence that support the biological plausibility of the new solutions. In addition, we showed that our approach reconstructs missing values at the leaves of the Fungi evolutionary tree better than ML or MP.

Key words: Co-evolution, maximum likelihood, maximum parsimony, reconstruction of ancestral genomes.

[1] Faculty of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel.
[2] Department of Molecular Genetics, Weizmann Institute of Science, Rehovot, Israel.
[3] School of Computer Science, Tel Aviv University, Tel Aviv, Israel.
[4] Department of Molecular Microbiology and Biotechnology, Tel Aviv University, Tel Aviv, Israel.
[5] School of Medicine, Tel Aviv University, Tel Aviv, Israel.

*T.T. and H.B. contributed equally to this work.

1. INTRODUCTION

The problem of reconstructing ancestral genomic sequences is as old as the field of molecular evolution. The first approach for inferring ancestral genomic sequences was suggested by Fitch around 40 years ago (Fitch, 1971). This first algorithm assumed a binary alphabet, and was based on the Maximum Parsimony (MP) criteria, i.e., find the labels to the internal nodes of a tree that minimize the number of mutations along the tree edges. Over the years this basic algorithm was generalized in many ways. Sankoff showed how to efficiently solve versions of the maximum parsimony problem with a non-binary alphabet and with multiple edge weights (Sankoff, 1975). Similar algorithms for inferring ancestral sequences based on maximum likelihood (ML; instead of maximum parsimony) were suggested more than 15 years later (Barry and Hartigan, 1987; Pupko et al., 2000; Elias and Tuller, 2007; Felsenstein, 1993; Krishnan et al., 2004; Pagel, 1999). Recently, similar approaches were used to infer ancestral sequences in phylogenetic networks (Jin et al., 2006).

Dozens of biological studies have dealt with the reconstruction of ancestral genomic sequences and ancestral genomes. For example, reconstruction of ancestral sequences was used for understanding the origins of genes and proteins (Zhang and Rosenberg, 2002; Thornton et al., 2003; Tauberberger et al., 2005; Jermann et al., 1995; Hillis et al., 1994; Gaucher et al.; 2003, Cai et al., 2004; Blanchette et al., 2004; Ouzounis et al., 2006), and for aligning genomic sequences (Hudek and Brown, 2005), for inferring ancestral enzymes and genomes (Yang et al., 1995; Koshi and Goldstein, 1996; Rascola et al., 2007; Ma et al., 2006; Csurӧ̋s and Miklӧ̋s, 2009; Cohen et al., 2008), noncoding genomic regions (Gorbunov et al., 2009), and SNPs (Hacia et al., 1999).

The main problem related to reconstructing ancestral sequences and genomes is that in practice many times the reconstructed sequences contain a large number of errors. A major source of this phenomenon is the existence of multiple local and/or global maxima in the solution space searched by both the maximum likelihood and the maximum parsimony approaches (Chor et al., 2000; Tuller et al., 2009a). Thus, many times our confidence in the most likely or most parsimonious reconstructed ancestral state is not too high. As the above mentioned algorithms assume that different sites and different genes/proteins evolve independently, this problem cannot be solved by adding more samples or more taxa (Li et al., 2008).

Several studies demonstrated that functionally and physically interacting proteins tend to co-evolve (Juan et al., 2008; Sato et al., 2003, 2005; Tuller et al., 2009b; Felder and Tuller, 2008), and that co-evolutionary relations between proteins are quite ubiquitous (Wapinski et al., 2007). Some of these previous investigations used the fact that interacting proteins have correlative evolution in order to successfully predict physical interactions (Juan et al., 2008; Sato et al., 2003, 2005).

Based on these results, we recently suggested a different approach, the Ancestral Co-Evolver (ACE), for improving the accuracy of reconstructed ancestral genomes (Tuller et al., 2009a). Our approach was based on utilizing information embedded in the co-evolution of functionally/physically interacting proteins. We used this approach to study the genome content of the Last Universal Common Ancestor (LUCA). In this work we give a formal description of the ancestral co-evolution problem, we analyze its computational complexity and describe algorithms for solving it; the performances of these algorithms are demonstrated by a simulation study. Additionally, we use the ACE for studying a new biological example, the ancestral genome content of the Fungi; specifically, we show that our approach reconstructs missing values at the leaves of the Fungi evolutionary tree better than ML or MP.

2. DEFINITIONS AND PRELIMINARIES

For simplicity, we assume a binary alphabet. However, all the results here can be easily generalized to models with more than two characters. In this work we assume that in general, if they do not have co-evolutionary relation, neighbor sites in the input sequences evolve independently. Thus, the basic components in the model and algorithms are single characters.

Our goal is to reconstruct the ancestral states for a set of organisms $\mathcal{T}$ of size $|\mathcal{T}| = n$. A *phylogenetic tree* is a rooted binary tree $T = (\mathrm{V}(T), \mathrm{E}(T))$ together with a *leaf labeling* function λ, where $\mathrm{V}(T)$ is the set of vertices and $\mathrm{E}(T)$ the set of edges. In our context, a weight table is attributed to each edge $(u, v) = e \in E(T)$. This *weight table* includes a weight (a positive real number) for each pair of labels of the two vertices $(u, v) = e$.

In this work, we assume that each node in a phylogenetic tree corresponds to a different organism. The leaves in a phylogenetic tree correspond to organisms that exist today ($\mathcal{T}$), while the internal nodes correspond to organisms that have become extinct ($\mathcal{T}'$). Thus, the *leaf labeling* function is a bijection between the leaf set L(T) and the set of organisms that exist today, $\mathcal{T}$

In our binary case, each label is a binary sequence; all the sequences have the same length. In the case of conventional ML/MP, as we assume an i.i.d. case where different characters in a sequence evolve independently, we can describe the algorithm for sequences of lengths one: i.e., either "1" or "0". A *full labeling* of a phylogeny $\hat{\lambda}(T)$ is a labeling of all nodes of the tree such that the labels at the leaves are the same as in λ, i.e., for all $l \in L(T)$ $\lambda(l) = \hat{\lambda}(l)$.

We can name each node after its corresponding organism. Let $O_T(\cdot)$ denote a function that returns the index of the organisms corresponding to each node in T, i.e., for every $v \in V(T)$, $O_T(v)$ is the index of the organism (from $\mathcal{T} \cup \mathcal{T}'$) corresponding to v.

A *co-evolving forest* $F = (S_F = \{T_1, T_2, \ldots\}, E_c(S_F))$ is a set of *phylogenetic trees*, S_F, with *identical* topology that correspond to the same organisms [i.e., each tree has the same $O(\cdot)$], and an additional set of edges, $E_C(S_F)$, that connect pairs of nodes in *different* trees. This set of edges represents the coevolutionary relations between pairs of protein families. Edges in $E_C(S_F)$ must connect pairs of nodes that correspond to the same organism (i.e., $(v, u) \in E_c(S_F), v \in V(T_1), u \in V(T_2) \Rightarrow O_{T_1}(v) = O_{T_2}(u)$; Fig. 1); we call such pairs of nodes *legal co-evolutionary pairs*.

The edges in *Ec* (*SF*) are named *co-evolution edges* while edges that are part of the evolutionary trees are named *tree edges*. For example, Figure 1A includes a *co-evolving forest* with two trees (the *co-evolution edges* are dashed with arrows while the *tree edges* are continuous). As co-evolutionary relations. In this work we assume that new *co-evolutionary edges* do not appear/dissapear during evolution. Namely, we assume that if there is a *co-evolutionary edge* between a *legal co-evolutionary pair* of nodes in two trees than all the *legal co-evolutionary pairs* of nodes in the two trees are connected by *co-evolutionary edge*.

A *full labeling* of a *co-evolving forest* $\hat{\lambda}(S_F)$ is a full labeling, $\{\hat{\lambda}(T_1), \hat{\lambda}(T_2), \ldots\}$, of all the nodes of the trees in S_F. The roots of a *co-evolving forest* are the set of roots of the *phylogenetic trees* in the *co-evolving forest*.

As mentioned, a *co-evolving forest* also includes a weight table for each *co- evolution edge* and each *tree edge*. These weight tables are cost functions that return a real positive number for each pairs of labels at the two ends of the edge. In the case of *tree edges*, these weights reflect the probability of a mutation along the

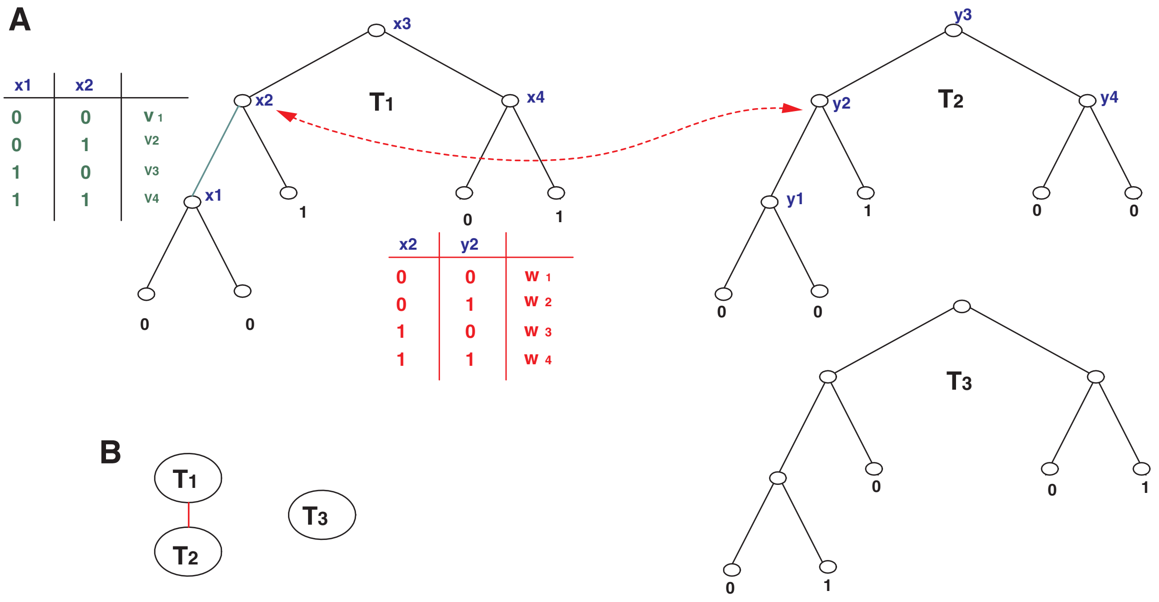
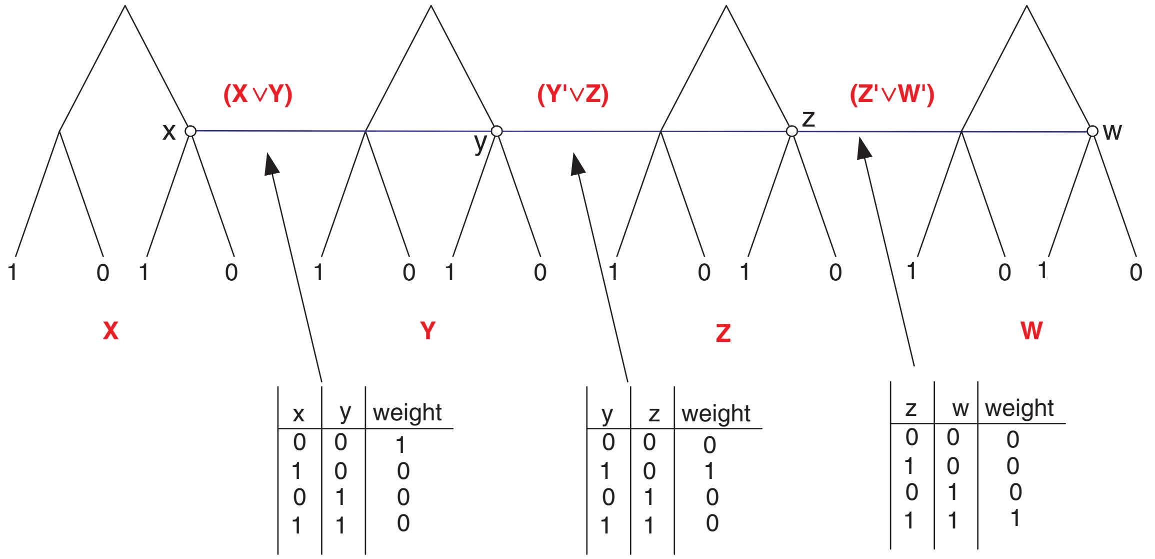


FIG. 1. **(A)** A simple example of a *co-evolving forest* with three trees, and one *co-evolution edge* connecting node x_2 in tree T_1 and node y_2 in tree T_2; the weight table corresponding to this co-evolution edge is in red. The weight table corresponding to the tree edge (x_1, x_2) in T_1 is in green. **(B)** The *co-evolutionary graph* corresponding to the *co-evolving forest* in *A*.



x	y	weight
0	0	1
1	0	0
0	1	0
1	1	0

y	z	weight
0	0	0
1	0	1
0	1	0
1	1	0

z	w	weight
0	0	0
1	0	0
0	1	0
1	1	1

FIG. 2. Reduction from *max–2–sat* to *Ancest–co–evol*.

edge. In the case of *co-evolution edges*, these weights reflect the distribution of mutual occurrence of the labels of the nodes at the ends of the edge.

This leads us to the formal definition of the problem we are concerned with, the *Ancestral co-evolution* problem:

Problem 1. Ancestral co-evolution (*Ancest–co–evol*)
Input: A *co-evolving forest*, $F = (S_F, E_C\ (S_F))$, and a real number, B.
Question: Are there labels for the internal nodes of all the trees in the *co-evolving forest* such that the sum of the corresponding weights along all the tree edges and the *co-evolution* edges is less than B?

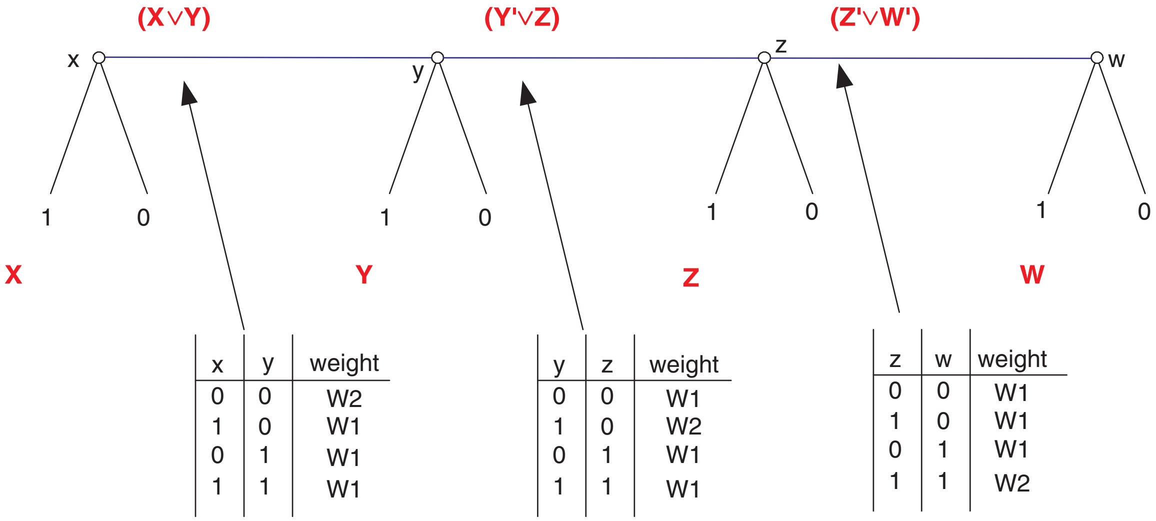
The following example demonstrates the advantages of the ancestral co-evolver compared to the simple *i.i.d* parsimony approach.

Example 1. Consider the co-evolving forest that appears in Figure 1, and assume that all the weight tables of the tree edges are identical to the table that appears in the figure where $[v_1, v_2, v_3, v_4] = [0, 1, 1, 0]$. It is easy to see that there are two MP solutions for the labels of the internal nodes in the phylogenetic tree T_1: either $[x_1, x_2, x_3, x_4] = [0, 0, 0, 0]$ or $[x_1, x_2, x_3, x_4] = [0, 1, 1, 1]$ gives the same score (2). In the case of the tree T_2, it is easy to see that there is one MP solution: all the labels of the internal nodes are "0." Now, suppose that in the weight table corresponding to the co-evolution edge (x_2, y_2), $w1$ is the smallest entry. Thus, by co-evolution study, the solution $[x_1, x_2, x_3, x_4] = [0, 0, 0, 0]$ is more plausible and the ambiguity in the labels of T_1 is solved.

Note that in general it is not necessarily required that the solution of each tree *separately* will be the most parsimonious (see the next sections). The minimal sum of edge weights corresponding to a *co-evolving forest*, *F* (problem 1) is named the *cost* of *F*. A *co-evolutionary graph* is an undirected graph that describes the *co-evolution* edges in the *co-evolving forest*. In such a graph, each node corresponds to a tree in the *co-evolving forest*, and two nodes are connected by an edge if there is at least one co-evolution edge between their corresponding trees. A connected component in the *co-evolving forest* is a sub-set of trees whose corresponding nodes in the co-evolutionary graph induce a connected component (see an example in Fig. 1B).

We finish this section with an observation that will be used in the next sections.

Observation 1 (*Tuller* et al., *2009a*). *The optimization problem of inferring the ancestral states of a phylogenetic tree when the optimization criteria is maximum likelihood (Pupko* et al., *2000) under i.i.d models such as Jukes Cantor (JC) (Jukes and Cantor, 1969), Neyman (1971), or the model of Yang* et al. *(1995) can be formalized as a maximum parsimony problem for non-binary alphabet and with multiple edge weights (Sankoff, 1975).*



x	y	weight
0	0	W2
1	0	W1
0	1	W1
1	1	W1

y	z	weight
0	0	W1
1	0	W2
0	1	W1
1	1	W1

z	w	weight
0	0	W1
1	0	W1
0	1	W1
1	1	W2

FIG. 3. Reduction from *gap–max–2–sat* to *gap–Ancest–co–evol*.

Observation 1 teaches us that the *Ancestral co-evolution* problem without *co-evolution edges* ($|E_C(S_F)| = 0$) can describe a Maximum Likelihood (ML) problem. In this work, indeed the weights of the *tree edges* correspond to the probabilities to gain/lose proteins and thus we describe and solve a generalization of both the ML and the MP problems on trees.

It is important to note that the model that we deal with in this paper is a generalized parsimony and not a full probabilistic model (like Bayesian network or Markov network). This is still true even when the tree edges correspond to probabilities of mutation; in this case, when we consider only the tree edge, we solve the maximum likelihood problem; when we consider also the co-evolutionary edges we can find solutions that are close to the maximum likelihood solution (depending on the weighting of the co-evolution edges; see also Subsection 4.2) for the tree edges alone but these are not maximum likelihood solution(s) for a model that corresponds both to the tree edges and the co-evolutionary edges.

3. HARDNESS ISSUES

In this section, we show that the *Ancestral co-evolution* problem is NP-hard. We will show it by reduction from the *max–2–sat* problem which is known to be NP-hard (Garey and Johnson, 1979) (the reduction appears in Fig. 2).

Problem 2. Maximum 2-Satisfiability (*max–2–sat*)
Input: Set U of variables, collection C of clauses over U such that each clause $c \in C$ has $|c| = 2$, and a positive integer $K \leq |C|$.
Question: Is there a truth assignment for U that simultaneously satisfies at least K of the clauses in C?

Theorem 1. *The Ancest–co–evol problem is NP-hard.*

Proof. By reduction from *max–2–sat*. Given an input $<U, C, K>$ to the *max–2–sat* problem reconstruct the following input to the *Ancest–co–evol* problem (Fig. 3): $|S_F| = |U|$, and each tree in S_F corresponds to one variable in U; each tree has the same structure and leaf labels as described in Figure 3. Each *co-evolution* edge in E_C corresponds to one clause in C, and connects the right-most internal nodes in two trees (Fig. 3). The translation of the four types of possible clauses (true-true, true-false, false-false) to the weight matrix of its corresponding *co-evolution* edge appears in Figure 3. Finally, choose $B = 2 \cdot |U| + |C| - |K|$.

It is easy to see that the optimal parsimony score for each tree in the *Ancest–co–evol* problem (excluding the *co-evolution* edges) is 2; either a solution that labels all the internal nodes with "0" or a solution that labels all the internal nodes with "1" gives this score.

$\Rightarrow$ Suppose the answer to the *max–2–sat* problem is *YES* (i.e., there is a truth assignment for U that simultaneously satisfies at least K of the clauses in C). In this case, we choose the labeling of all the internal nodes of each tree to be identical to the assignment of its corresponding variable in U. Thus, the contribution of all the tree edges to the parsimony score (i.e., excluding the *co-evolution* edges) is $2 \cdot |U|$.

By the construction of the *weight tables* (Fig. 3), the contribution to the parsimony score due to *co-evolution* edges that correspond to one of the K satisfied clauses is 0, and the contribution to the parsimony score due to each of the other *co-evolution* edges is at most 1. Thus, the contribution of all the *co-evolution* edges to the parsimony score is at most $|C| - |K|$, and the answer to the *Ancest–co–evol* problem is *YES.*

$\Leftarrow$ Suppose the answer to the *Ancest–co–evol* problem is *YES* (i.e., the parsimony score along all the edges is no more than $2 \cdot |U| + |C| - |K|$). As mentioned earlier, for optimal parsimony score, in each tree all the internal nodes should be labeled with the same label as that of the internal node that is connected by *co-evolution* edges. Thus, the contribution of all the *tree edges* to the parsimony score is $2 \cdot |U|$, and the contribution of all *co-evolution* edges to the parsimony score is at most $|C| - |K|$. Thus, the contribution to the parsimony score for K of the *co-evolution* edges is 0. By the reconstruction of the *co-evolution* edges, they correspond to K clauses in C that are be satisfied when the assignment to each variable in U is identical to the labeling of the internal nodes of its corresponding tree. Thus, the answer to the *max–2–sat* problem is *YES*. ■

In the rest of this section, we will show that *Ancest–co–evol* is hard to approximate; i.e., we will show that the following gap problem is NP-hard:

Problem 3. Gap ancestral co-evolution (*gap–Ancest–co–evol*$[B_1, B_2]$)

Input: A *co-evolving forest*, $F = (S_F, E_C(S_F))$, and two real number, B_1 and B_2.

Output: If there are labels for the internal nodes of all the trees in the *co-evolving forest* such that the sum of the corresponding weights along all the tree edges and the *co-evolution* edges is less than B_1 then answer "Yes"; if there are no labels for the internal nodes of all the trees in the *co-evolving forest* such that the sum of the corresponding weights along all the tree edges and the *co-evolution* edges is less than $B2$ then answer "No"; otherwise answer either "Yes" or "No."

We will show it by reduction from *gap–max–2–sat* that is known to be NP-hard (Håastad, 2001):

Problem 4. *gap–max–2sat*$[\rho_1{*}|C|, \rho_2{*}|C|]$

Input: Set U of variables, collection C of clauses over U such that each clause $c \in C$ has $|c| = 2$, and two positive real numbers, $\rho_1 < \rho_2 < 1$, such that $\rho_1{*}|C|$ and $\rho_2{*}|C|$ are integers.

Output: If there is a truth assignment for U that simultaneously satisfies at least $\rho_2{*}C$ of the clauses in C then answer "Yes"; if no truth assignment for U simultaneously satisfies more than $\rho_1{*}C$ of the clauses in C then answer "No"; otherwise answer either "Yes" or "No."

By (Håstad, 2001) $\frac{\rho_2}{\rho_1} \geq \frac{22}{21}$.

Theorem 2. *The gap–Ancest–co–evol*$[(|E_C|{-}\rho_2{*}|E_C|){*}W_2 + \rho_2{*}|E_C|{*}W_1,\ (|E_C| - \rho_1{*}|E_C|){*}W_2 + \rho_1{*}|E_C|{*}W_1]$ *problem is NP-hard.*

Proof. Given an input $< U, C, K >$to the *gap–max–2–sat* problem an input to the *Ancest–co–evol* problem which is similar to the reduction described in theorem 2. The only changes are the topology of the trees, the weight matrixes, and the fact that there is no penalty on the labels at the ends of the tree edges; see Figure 3); we assume that $W_2 > W_1$ and choose $B_1 = (|EC| - \rho_2{*}|E_C|){*}W2 + \rho_2{*}|E_C|{*}W_1$ and $B_2 = (|E_C| - \rho_1{*}|E_C|){*}W_2 + \rho_1{*}|E_C|{*}W_1]$.

By our reduction, if more than $\rho_2{*}|C\ |$ clauses can be satisfied $\Longleftrightarrow$ the *Ancest–co–evol* has a solution whose cost $<B_1$. If no more than $\rho_1{*}|C|$ clauses can be satisfied $\Longleftrightarrow$ the *Ancest–co–evol* does not have a solution whose cost$< B_2$. Thus if *gap–max–2sat*$[\rho_1{*}|C|, \rho_2{*}|C|]$ is NP-hard, then *gap–Ancest–co–evol*$[(|E_C| - \rho_2{*}|E_C|){*}W_2 + \rho_2{*}|E_C|{*}W_1, (|E_C| - \rho_1{*}|E_C|){*}W_2 + \rho_1{*}\ |E_C|{*}W_1]$ is NP-hard.

Thus, there is no algorithm with approximation ration $< \frac{(1-\rho_1){*}W_2/W_{1+\rho_1}}{(1-\rho_2){*}W_2/W_{1+\rho_2}}$. Since $W_2 > W_1$ the upper bound on the approximation ratio is gained for very large $W_2/W_1 : \frac{(1-\rho_1)}{(1-\rho_2)}$ ■

Corollary 1. *There is a constant ζ' such that there is no polynomial time algorithm for* Ancest-co-evol *with performance ratio better than ζ'.*

4. METHODS

4.1. An FPT algorithm and approximation heuristics

As we have shown in the previous section, the *Ancestral co-evolution* problem is NP-hard. In this section, we describe an FPT algorithm and an approximation algorithm for the *Ancestral co-evolution* problem. The approximation algorithm is described in Figure 4. It has three main steps: (1) The input *co-evolving forest* is partitioned into smaller *co-evolving forests*; (2) optimal labels are assigned to the internal nodes of each of these *co-evolving forests* by an FPT algorithm; in total, these labels are an approximation of the solution for the input *co-evolving forest*; (3) finally, the solution is further improved greedily.

We will start by describing an FPT algorithm that finds the optimal solution for the *Ancestral co-evolution* problem in time complexity that is exponential with the number of trees in S_F but polynomial with the other properties of the input. This algorithm is used in step 2 of the approximation algorithm where it is implemented on subsets of S_F. Similarly to many algorithms for computing the labels of internal tree nodes (Fitch and Margoliash, 1967; Sankoff, 1975), our algorithm has two phases (in the first phase, it traverses the *co-evolving forest* from the leaves to the root; in the second phase, it traverses the *co-evolving forest* from the root to the leaves). However, our algorithm is performed jointly for all the trees in each connected component.

Let $W^c_{e;(a,b)} = W^c_{(i,j);(a,b)}$ denote the cost of assigning *a* to node *i*, and *b* to node *j* where (*i, j*) is a co-evolution edge. Similarly, if (*i, j*) is a *tree edge* we use $W^b_{e;(a,b)} = W^b_{(i,j);(a,b)}$ to denote the cost of assigning *a*

```
A. Ancestral-Co-Evolver $(S_F, E_C(S_F), M)$
  $(S_{F1}, E_{C1})(S_{F1}, E_{C1}))...(S_{Fm}, E_{Cm})) \leftarrow$ Partite $(S_F, E_C(S_F), M)$
  for $i = 1 : m$ do
    $\hat{\lambda}(S_{Fi}) \leftarrow$ Mutual Ancestral $(S_{Fi}, E_{Ci})$
  end for
  Greedy Ancestral $(S_F, E_C(S_F), \hat{\lambda}(S_F)$

B. Partite $(G(V, E), M)$
  $(H_1, H_2, ..., H_{k'}) \leftarrow$ k-means $(G, k' = |V|/M)$
  for $i = 1 : k'$ do
    if $|H_i| > M$ then
      Partite $(H_i, M)$
    else
      Return $H_i$
    end if
  end for

C. Mutual Ancestral $(S_F, E_C)$
  Phase 1-from leaves to root:
  Leaves: $S_{\bar{x}}(\bar{v}) = \{ \begin{matrix} \overline{x} = \overline{v} & 0 \\ o.w. & \infty \end{matrix}$
  $S_{\overline{t}}(\overline{v}) = \min_{\overline{u}}\{\Sigma_j W^b_{(t_j,\ell_j);(v_j,u_j)} + S_{\overline{l}}(\overline{u})\} + \min_{\overline{w}}\{\Sigma_j W^b_{(t_j,k_j);(v_j,w_j)} + S_{\overline{k}}(\overline{w})\} + \sum_{j_1,j_2:(t_{j_1},t_{j_2}) \in E_c} W^c_{(t_{j_1},t_{j_2});(v_{j_1},v_{j_2})}$
  Phase 2-from root to leaves:
  Root: $\overline{v}* = argmin_{\overline{v}}\{S_{\overline{t}}(\overline{v})\}$
  $\overline{v}* = argmin_{\overline{v}}\{\Sigma_j W^b_{(t_j,k_j);(t*_j,v_j)} + S_{\overline{k}}(\overline{v})\}$

D. Greedy Ancestral $(S_F, E_C(S_F), \hat{\lambda}(S_F))$
  while There is improvement in the MP score do
    Find $e \in \{S_F, E_C(S_F)\}$ and labels for $e$ that improve the MP score.
    Update the labels of $e$.
  end while
```

FIG. 4. **(A)** The general algorithm for inferring ancestral states under co-evolution. **(B)** The algorithm for partitioning the co-evolutionary graph. **(C)** The algorithm for finding the ancestral states of a connected component (output of the Partite algorithm). **(D)** The Greedy stage of the algorithm.

and b to the two nodes of e respectively. Let $S_{\bar{t}}(\bar{v})$ denote the cost of a sub-forest in the *co-evolving forest* whose roots are $\bar{t}$, when assigning $\bar{v}$ to these roots.

Let tj denote the j-th node in the vector of nodes $\bar{t}$. Let $\bar{k}$ and $\bar{l}$ denote the corresponding vectors of children of $\bar{t}$ in the *co-evolving forest*. In the first phase, all $S_{\bar{t}}(\bar{v})$ are computed by the following dynamic programming formula (Fig. 4C):

$$S_{\bar{t}}(\bar{v}) = min_{\bar{u}}\left\{\sum\nolimits_j W^b_{(t_j,l_j);(v_j,u_j)} + S_{\bar{l}}(\bar{u})\right\} + min_{\bar{w}}\left\{\sum\nolimits_j W^b_{(t_j,k_j);(v_j,w_j)} + S_{\bar{k}}(\bar{w})\right\} + \sum\nolimits_{j_1,j_2:(t_{j_1},t_{j_2})\in E_c} W^c_{(t_{j_1},t_{j_2});(v_{j_1},v_{j_2})}$$

In the second phase, the algorithm traverses the sub-forest from the roots to the leaves, and optimal values are assigned to the internal nodes of the *co-evolving forest* by the following dynamic programming formula (Fig. 4C):

For the roots of the *co-evolving forest*:

$\bar{v}^* = argmin_{\bar{v}}\{S_{\bar{t}}(\bar{v})\}$

For a general vector of internal nodes $\bar{k}$ corresponding to the same organism in the *co-evolving forest* (after the values $\bar{t}^*$ were assigned to its parents, $\bar{t}$):

$$\bar{v}* = argmin_{\bar{v}}\left\{\sum\nolimits_j W^b_{(t_j,k_j;(t_j^*,v_j)} + S_{\bar{k}}(\bar{v})\right\}$$

The running time of this algorithm on a *co-evolving forest* with m' trees of size n is $O(n \cdot 2^{3*m'})$ since, in each of the $O(n)$ vectors of internal nodes (nodes corresponding to the same organism) it checks all the $2^{m'}$ possible labels of the vector of internal nodes *vs.* the $2^{m'} \times 2^{m'}$ possible simultaneous labels to all nodes corresponding to the vectors of the direct descendant of this vector.

As the running time of the algorithm described above is exponential with the size of *co-evolving forest*, the general algorithm (Fig. 4A) has an initial stage (stage 1) where the input graph is partitioned into small enough connected components.

As the running time of the algorithm described above is exponential with the size of the largest connected component in the *co-evolutionary graph*, the general algorithm (Fig. 4A) has an initial stage where the input graph is partitioned into small enough connected components. The input to the general algorithm (ACE) includes the maximal size of a connected component after this stage. Let M denote this parameter.

This step is described in Figure 4B and is performed by an algorithm that recursively implements k-means (MacQueen, 1967) on the *co-evolutionary graph*. On the first iteration, the number of clusters is $|V|/M$ where $|V|$ is the number of vertices in the co-evolutionary graph. If the size of some cluster is larger than M, the algorithm is executed recursively on this cluster to further partite it to smaller connected components. The algorithm stops when all parts of the graph (connected components) are smaller than M. Though the problem of clustering is NP-hard, in practice, and as reported in the next section, the k-means algorithm is very fast.

The input to this step is a weighted graph whose edges correspond to the edges in the *co-evolutionary graph*. The weights of the graph edges can be any measure that represents the strength of the co-evolution between the corresponding trees (for example, the correlation between the phyletic pattern of the corresponding proteins).

The final step of the *Ancestral-Co-Evolver* algorithm is a greedy stage (Fig. 4D). In each iteration, the greedy algorithm searches for an edge and labels its ends in a way that improves the cost of the *co-evolving forest*. As demonstrated in the simulations in Section 5.2, this algorithm converge to a local optimum faster than the running time of the dynamic programming stage with $M = 7$. Note that the greedy algorithm can be stopped after a certain number of iteration if it does not converges to a local optimum. The greedy stage can be used as an independent algorithm when running it from various initial points (e.g., one of the initial points can be the ML solution).

Let $\hat{\lambda}(S_F)$ denote the labels found by the *Ancestral-Co-Evolver* algorithm. Let $(\hat{\lambda}(S_F))$ denote the parsimony score induced by these labels. Let $MP^-(\hat{\lambda}(S_F))$ denote the parsimony score induced by these labels when not considering the co-evolution edges *between* the connected components found by the *Partite* algorithm; let E^- denote this set of co-evolution edges. An upper bound on the approximation ratio of the general algorithm is given in the following observation:

Observation 2. *The approximation ratio of the* Ancestral-Co-Evolver *algorithm is* $\leq \frac{MP(\hat{\lambda}(S_F))}{MP-(\hat{\lambda}(S_F))}$.

Proof. The labels found for the output of the *Partite* algorithm are optimal for that graph (the co-evolutionary graph without E^-), and as this output includes less co-evolution edges than S_F, the optimal MP score is $\geq MP^-(\hat{\lambda}(S_F))$. ■

We used Observation 2 in order to estimate the approximation ratios of the different algorithms in the simulations.

One important property of the algorithm is that it enables a trade-off between accuracy and speed. A larger M (smaller co-evolving sub-forests) increases the running time exponentially but at the same time increases the accuracy of the solution; $M = |S_F|$ will give the optimal solution for the *Ancestral co-evolution* problem.

Finally, it is important to note that by weighting the *co-evolution edges* relatively to the *tree edges* we can control the relative influence of these two sources of information (co-evolution *vs.* the evolutionary tree) on the resulting labels. Thus, for example, it is easy to see that (and a very similar proof can be outlined for the case where *tree edges* are rational numbers):

Let $\hat{\lambda}(S_F/E_C)$ denote the set of the optimal labels of S_F when not considering the co-evolution edges (i.e., $E_C = 0$).

Observation 3. *If the weight tables of the* tree edges *are natural numbers and all the entries in the weight tables of the* co-evolution edges $< \frac{1}{|E_c(S_F)|}$ *then the* optimal *labeling of the* co-evolving forest *is* one *of the optimal labels in* $\hat{\lambda}(S_F/E_C)$.

Proof. Suppose a labeling λ_1 has the optimal cost on the co-evolution forest S_F and is not in $\hat{\lambda}(S_F/E_c)$. Thus, the contribution of the tree edges to the cost of λ_1 is greater than those of labelings in $\hat{\lambda}(S_F/E_c)$. When considering only the *tree edges*, the minimal difference between the costs of two solutions that have different costs is ≥ 1. On the other hand, the maximal contribution of all the *co-evolution edges* to the cost of any labeling is less than 1. As all weights are positive, λ_1 cannot be an optimal solution to the complete co-evolution forest (as any solution in $\hat{\lambda}(S_F/E_c)$ has a lower cost). ■

By Observation 1, the *ancestral co-evolution* problem without *co-evolution edges* describes a conventional maximum likelihood problem. Thus, by Observation 3, if we choose small enough weights for the *co-evolution edges* our method can be used for choosing *one* of the optima (or a point very close to *one* of the optima) of the maximum likelihood function—the one that is supported by co-evolutionary relations.

4.2. Weight tables and weighting of the co-evolution edges

A pair of evolutionary trees was connected by a co-evolution edge if the following conditions were satisfied: (1) we found an evidence for physical or function interaction between the two corresponding proteins (based on String database (2) The co-occurrences distribution of the two proteins in a large group of organisms (we used the data from Tuller et al., 2009a) was far from being uniform (see an example in Section 5.3). The values in the co-evolutionary weight tables were based on the co-occurrences distribution in the organisms from Tuller et al. (2009a) (i.e., each table included the four possible probabilities that the corresponding labels of the pair of proteins will appear in an organism).

We tested several values for the weights of the *co-evolution edges* compared to the *tree edges*. At one extreme, the entries of the tree edges weight tables are multiplied by a very large constant. In this case, the tree edge weight tables are dominant compared to the weights of the *co-evolution edges* (the solution is one of the ML/MP solutions). At the second extreme, the fifth weighting, the *co-evolution edge* weights are dominant compared to the tree edge weights. In this case, the entries of the tree edges weight tables are multiplied by a very large constant.

Let $MP^b(S_F, W)$ denote the parsimony score when solving the *ancestral co-evolution* problem with weighting W and when considering only *tree edges*. Let $MP^c(S_F, W)$ denote the parsimony score when solving the *ancestral co-evolution* problem with weighting W and when considering only *co-evolution edges*. In this work, we used the weighting, W*, that optimizes the sum of the two sources of information (co-evolution, and the evolutionary trees); i.e., we used $W^* = argmin_W\left(\frac{MP^c(S_F, W)}{min_W(MP^c(S_F, W))} + \frac{MP^b(S_F, W)}{min_W(MP^b(S_F, W))}\right)$.

5. EXPERIMENTAL RESULTS

5.1. Simulated evolution

To analyze the performances of the algorithms described in the previous section we generate a probabilistic process that describes the evolution of a *co-evolving forest*. In the simulation, each character evolves along the branches of the evolutionary trees, but also has correlations with the other characters that interact with it.

The simulation was performed as in Tuller et al. (2009a): The topology of the trees in the *co-evolving forest*, their edges' lengths (probability of mutation according to JC model), and the *co-evolution* edges between pairs of nodes at the root of the trees, were all chosen randomly. We also assigned an additional parameter, *pc*, to each pair of nodes that are *co-evolutionary legal*. This parameter denotes a small probability that a *co-evolution* edge between the node pair will appear/disappear. For simplicity we assume a binary alphabet.

To simulate the co-evolutionary process we generated a probabilistic process where each character evolves along the branches of the evolutionary trees (stage 1), but also has correlations with the other characters that interact with it (stage 2). The process has two main stages that are performed sequentially: Stage (1) (Fig. 5): The label of a node is generated after the label of its ancestor was generated (according to the corresponding tree branch length). The initial node is the root. Stage 2 (Fig. 5): After the labels of the nodes corresponding to all the sites of a certain organism in the *co-evolving forest* are generated, the label of each node is switched according to those of its neighbors (i.e., nodes in other trees that are connected to it with co-evolution edges). The switching is performed by randomly traversing all the nodes in the network and setting the label of each node to a value ("0" or "1") that appears in the majority of its neighbors (the value of a node cannot be changed after it was set).

In each experiment, the input to the algorithms included the following: (1) The real tree topology. (2) A noisy version of the edge lengths; we translate the input to a weight table in the following way: let p_e denote the mutation probability along the *tree edge e*; the weight entry of the corresponding edge is $-log(p_e)$ if the labels at the ends of the edge are equal, and $-log(1-p_e)$ if the labels at the ends of the edge are not equal. (3) As *co-evolution* edges we took the set of the *co-evolution* edges that was generated between the nodes corresponding to one of the organisms at the leaves (Fig. 5C). (4) The weight entries of the *co-evolution edges* were computed in each experiment by the phyletic patterns of the corre-

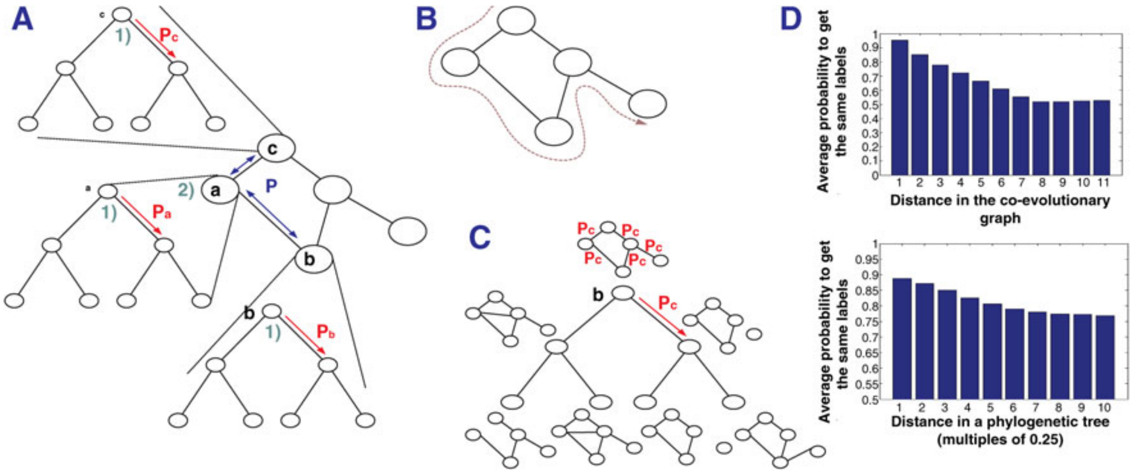

A
c
1)
Pc
c
a
2)
a
P
1)
Pa
b
b
1)
Pb
B
C
Pc
Pc
Pc
Pc
Pc
b
Pc
D
Average probability to get the same labels
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1 2 3 4 5 6 7 8 9 10 11
Distance in the co-evolutionary graph
Average probability to get the same labels
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
1 2 3 4 5 6 7 8 9 10
Distance in a phylogenetic tree (multiples of 0.25)

FIG. 5. Simulation of co-evolution. **(A)** The two major steps: (1) Red (first stage): evolution according to the evolutionary trees. (2) Blue (second stage): selection according to the interacting sites (co-evolution). **(B)** Stage (2) involves a random walk along all the nodes corresponding to a certain organism in the *co-evolving forest*; the value of a node is set to be similar to the value in the majority of its neighbors. **(C)** At the beginning of the simulation, random *co-evolution edges*, between pairs of roots in the *co-evolving forest* are added. There is a small probability that some of these initial *co-evolution edges* will be lost or that new ones will emerge. **(D)** Average probability to get the same label in a pair of nodes in this simulation as a function of the distance in the *co-evolutionary graph* and in the phylogenetic tree. Nodes that are closer in the *co-evolutionary graph* or in the phylogenetic tree tend to be more similar.

sponding pairs of nodes at the leaves. Let $p_{a,b}$ denote the empirical probability that a pair of labels (a, b) for the nodes connecting a *co-evolution edges* appears at the leaves. The corresponding weight for this entry is $-\,log(p_{a,b})$.

5.2. Simulation results

We compared the performances of the following algorithms: (1) The Partitioning algorithm (Fig. 4B) with the Dynamic Programming algorithm (Fig. 4C). (2) The greedy algorithm (Fig. 4D.) with a few initial points (one of them is the the ML solution). (3) The ACE algorithm (all the stages in Fig. 4). (4) The ML and MP algorithms that do not consider the *co-evolution edges*. Let *DP X* denotes a Dynamic Programming algorithm with $M = X$; let *ACEX* denotes an ACE algorithm with $M = X$.

A summary of the simulation results appears in Figure 6; sub-figures A–C depict the running times (log scale) and sub-figures D–F describe the approximation ratios as functions of the size of the *evolutionary*

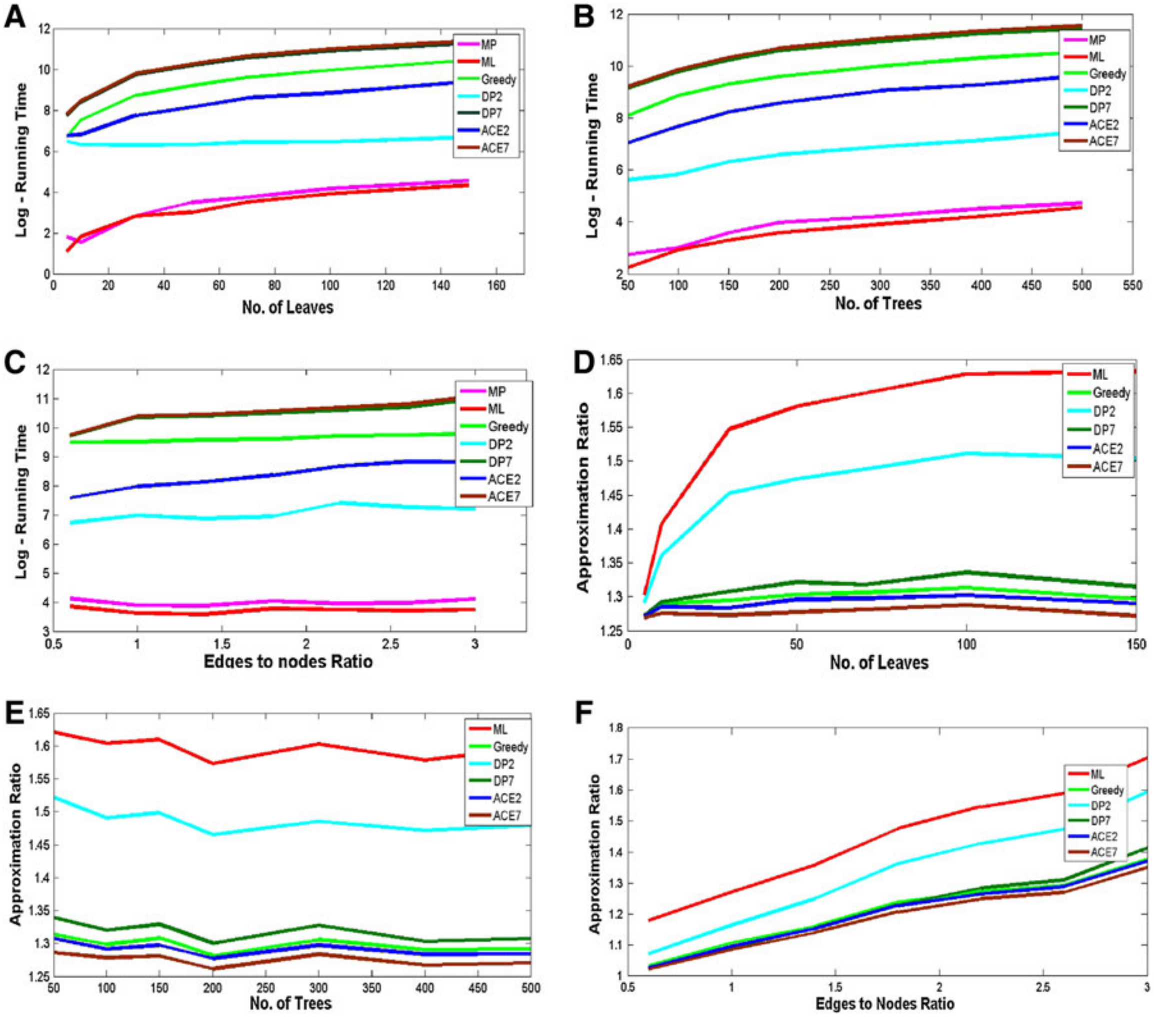


FIG. 6. Running time and accuracy of the partitioning algorithm with the Dynamic Programming algorithm, the greedy algorithm, the ACE, and the ML/MP algorithms. **(A–C)** The *log* running time (in ms) of the algorithms as function of the size of the *evolutionary trees* (*A*; 200 trees and 2.5 *co-evolution edges* per node in the *co-evolutionary graph*), number of *co-evolutionary trees* (*B*; 2.5 *co-evolution edges* per node in the *co-evolutionary graph*, each *evolutionary tree* with 70 leaves), and the number of *co-evolution edges* per node in the *co-evolutionary graph* (*C*; 200 *evolutionary trees*, each with 70 leaves). **(D–F)** The upper bound on the approximation ratio (Observation 2) of the solution found by each of the algorithms as function of the number of leaves in each *evolutionary tree* (*D*), the number of *evolutionary trees* (*E*), and the number of *co-evolution edges* per node in the *co-evolutionary graph* (*F*).

trees, the number of *evolutionary trees*, the number of co-evolution edges per node in the *co-evolutionary graph*. All the running finished in less than a four minutes. As can be seen, the running time increases exponentially with *M* (see *ACE*7, *DP*7 *vs. ACE*2, *DP*2 sub-figures A–C) while the running time of the greedy algorithm alone is larger than *DP*2 but exponentially smaller than *DP*7. As can be seen, in the case of running time, the most influential parameter is the number of *evolutions trees* in the input.

In the case of the approximation ratio (the upper bound from Observation 2), the most influential parameter is number of co-evolution edges per node in the *co-evolutionary graph*. As can be seen, *ACE*7 performs better than all the other algorithms and always has approximation ratio of <1.3. Interestingly, the greedy algorithm is only a few percentages worse. These results support using the greedy algorithm if running time matters. The fact that the *upper bound* of the *ACE*7 < 1.3 demonstrates that our approach can find solutions that are very close to the optimal ones. As we used here an *upper bound* on the approximation ratio the actual ratio can be significantly lower.

In the simulation, as in the case of biological data (see the next section), the ML solutions (when ignoring co-evolution) are relatively similar to the solutions found by our approach. Thus, it is not surprising that approximation ratio of ML is bound to be <1.7. On the other hand, the margin between the approximation ratio of the ML and that of the ACE is significant: up to 30%. This result demonstrates the essentiality of our approach.

5.3. *A biological example: reconstruction of the ancestral genome content the Fungi*

Using the method outlined above we set to reconstruct the ancestral genome content of 17 Fungi whose evolutionary tree appears in Figure 7. The input included 33,931 families of Fungi orthologs (downloaded from Wapinski et al., 2007) and a total of 20,317 co-evolution edges.

We represented each of the 17 genomes by a binary string of length 33931 where ‘1’ in the *x* position of a string means that there is a gene/protein from the *x* group of orthologs in this genome, and ‘0’ means that there is no gene/protein from the *x* group of orthologs in this genome. We used Neyman’s two state model (Neyman, 1971), a version of Jukes Cantor (JC) model (Jukes and Cantor, 1969) for inferring the edge

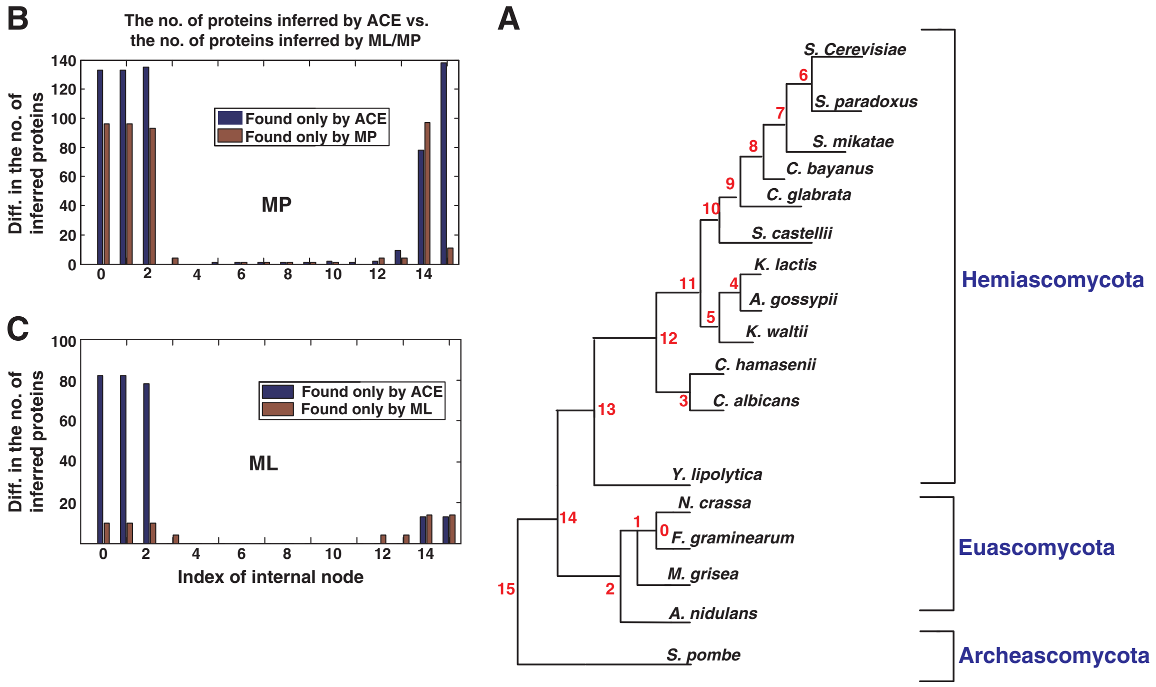


FIG. 7. **(A)** The evolutionary tree of the Fungi that was analyzed by the ACE. **(B)** Summary of the results for the MP case; the number of proteins inferred by ACE and not by MP and vice versa. **(C)** Summary of the results for the ML case; the number of proteins inferred by ACE and not by ML and vice versa.

lengths of the tree by maximum likelihood. This was done by PAML (Yang, 1997). These edge lengths correspond to the probabilities that a protein will appear/vanish along the corresponding lineage.

As co-evolution edges we use various physical and functional interactions that were downloaded from String et al. (2009) (http://string.embl.de/) (Tuller et al., 2009b) reported the relation between co-evolution and similar functionality). We filtered co-evolutionary edges for which the ratio between the highest and lowest probability in the co-occurrence distribution table was less than 4.25. The weights in the tables were computed according to the co-occurrence probabilities of the corresponding pairs of orthologs. We used the weighting whose corresponding solution optimizes both the score of the *co-evolution edges* and the score of the *tree edges* (see more details in Subsection 4.2). The annotation of ancestral proteins was based on the GO annotations of *S. cerevisiae*.

We compared our results of the ACE to those obtained by using only *tree edges* (by using MP and ML).

The total running time of the ACE algorithm on this large biological dataset was about 1.5 hours on a conventional PC. Figure 7 summarizes the results of the analysis by ACE. As can be seen, ACE removed/added hundreds of proteins to ML/MP labels of the internal nodes of the evolutionary trees. A major fraction of the discrepancies between the results of the ACE algorithm and those obtained with the conventional methods (ML and MP) appears in the *Euascomycota* subtree (internal nodes 0–2 in the Fungi tree; Fig. 7A). In general, ACE mainly added proteins to these nodes, implying that ML/MP underestimated the size of these ancestral genomes. Additionally, both in the case of MP and ML, the ACE added/removed many proteins from internal nodes 14 and 15. It is important to note that the likelihood score of the ACE solution was only 0.25% lower than the ML score, demonstrating that this solution was *very* close to the ML point. Similarly, when the ACE was implemented with MP the parsimony score of its solution was only 2.2% higher than that of the MP solution. These solutions, however, are supported by the co-evolutionary information and thus are more biologically plausible.

We further analyzed the proteins added by the ACE to the ML solution for the ancestors of the *Euascomycota* (internal nodes 0–2): the nodes with the largest number of discrepancies between ML and ACE. The groups of proteins added to each of these nodes were very similar (around 95% similarity); thus we report only the results for node 2, the ancestor of the *Euascomycota*.

ACE added 89 proteins to the ML solution of internal node 2. Various pieces of evidence support the biological plausibility of the addition of these proteins by ACE: First, the group of proteins added to this node was enriched with proteins that take part in basic and essential metabolic processes. Specifically, it was enriched with the cellular process: *protein amino acid phosphorylation* (p-value = 0.00054), *amine transport* (p-value = 0.00153), and *amino acid transport* (p-value = 0.00518); all p-values passed the False Discovery Rate (FDR) control for multiple hypothesis testing (Benjamini and Hochberg, 1995). Second, all these proteins have orthologs in most of the analyzed Fungi (on average in 76% of the Fungi); this fact also supports the essentiality of these proteins. Third, in *S. cerevisiae*, many of these proteins are part of the same complex with proteins inferred by the standard ML (note that co-evolutionary relations used by ACE do not necessarily imply association in the same complex).

The following are two typical examples that further demonstrate the three points mentioned above:

Example 1. Three of the proteins added by the ACE are orthologs of *S. cerevisiae TPK1*, *TPK2*, and *TPK3*. The presence of at least one of these genes is required for normal growth in *S. cerevisiae* (Toda et al., 1987). These genes are part of the *cAMP-dependent protein kinase complex* that also includes another protein (*BCY1*), which was inferred by ML.

Example 2. Orthologs of the *S. cerevisiae FAT1* gene appear in 76% of the analyzed Fungi. In *S. cerevisiae*, this protein forms a complex with *FAA1* or *FAA4* that imports and activates exogenous fatty acids (Zou et al., 2002). Both *FAA1* and *FAA4* were inferred by ML.

5.4. Reconstructs missing values at the leaves of the evolutionary tree by the ACE

At the next first stage, to further demonstrate the advantages of ACE over conventional ML/MP and to study how co-evolutionary relations improve error rate we performed the following procedure: (1) We randomly flipped 1.5%–4.5% of the values (500–1500 sites) with co-evolutionary relations in 6%–18% of the genomes (1–3 genomes), from absence to presence or vice versa. (2) We reconstructed the ancestral states of the co-evolutionary forest based on the altered genomic contents. (3) We then “fixed” the values at the leaves that were flipped by choosing labels that optimize the score of the co-evolutionary network given the states inferred in (2). (4) Steps 1–3 were repeated 7 times and the resulting error-rates were

Table 1. Reconstructs Missing Values at the Leaves of the Evolutionary Tree by the ACE

Mode	*Sample mode*	*Sampling size organisms*	*Sampling size sites*	*i.i.d.*	*Coev norm 1*	*Coev norm 2*	*Coev norm 3*	*Coev norm 4*	*Coev norm 5*	*Coev only*
ML	Coev only	1	500	0.043	0.043	0.014$^{\#}$	0.0134	0.00914	0.00714*	0.00714
ML	Coev only	2	1000	0.068	0.068	0.024$^{\#}$	0.0212	0.0202	0.0159	0.0154*
ML	Coev only	3	1500	0.107	0.107	0.0438$^{\#}$	0.04	0.039	0.035*	0.036
ML	Uniform	1	500	0.116	0.116	0.113$^{\#}$	0.113	0.113	0.11285*	0.932
ML	Uniform	2	1000	0.142	0.142	0.138$^{\#}$	0.138	0.1378	0.137*	0.934
ML	Uniform	3	1500	0.186	0.186	0.179$^{\#}$	0.179	0.179	0.1787*	0.933
ML	No Coev	1	500	0.11$^{*,\#}$	0.11	0.11	0.11	0.11	0.11	1
ML	No Coev	2	1000	0.147$^{*,\#}$	0.147	0.147	0.147	0.147	0.147	1
ML	No Coev	3	1500	0.192$^{*,\#}$	0.192	0.192	0.192	0.192	0.192	1
MP	Coev only	1	500	0.099	0.0365$^{\#}$	0.0268	0.0186	0.0134	0.00485	0.00428*
MP	Coev only	2	1000	0.183	0.068$^{\#}$	0.0433	0.0369	0.028	0.0146	0.0146
MP	Coev only	3	1500	0.288	0.1196$^{\#}$	0.073	0.0594	0.0472	0.0341*	0.0348
MP	Uniform	1	500	0.146	0.1437$^{\#}$	0.1437	0.14285	0.141*	0.141	0.934
MP	Uniform	2	1000	0.176	0.168	0.166	0.165	0.165	0.164	0.937
MP	Uniform	3	1500	0.237	0.2255$^{\#}$	0.2219	0.22	0.2188	0.2176*	0.934
MP	No Coev	1	500	0.14$^{*,\#}$	0.14	0.14	0.14	0.14	0.14	1
MP	No Coev	2	1000	0.18$^{*,\#}$	0.18	0.18	0.18	0.18	0.18	1
MP	No Coev	3	1500	0.23$^{*,\#}$	0.23	0.23	0.23	0.23	0.23	1

$^{\#}$, the maximal decrease in the error-rat; *, the minimal error rate.

averaged (to decrease the bias; small changes in the number of repeats do not change the conclusion of this analysis).

We considered three different modes of samplings of the sites to be flipped: (1) *Coev only*—sample only values that have co-evolutionary relations. (2) *Uniform*—sample uniformly from all values. (3) *No Coev*—sample only values that do not have co-evolutionary relations.

The mean error-rate of the inferred values at the leaves for the above procedure and for different MP/CE or ML/CE weightings, different sampling modes, MP and ML, as well as for different percent of flipping is provided in Table 1.

As can be seen, in the case of *Coev only* sampling, using the co-evolutionary information reduces the error rate by more than 50% where usually most of the improvement is achieved in the case of the first or the second weighting (whose MP/ML score is also relatively high). In the case of the *Uniform* sampling most of the improvement is also achieved in the case of the first or the second weighting but the improvement is more modest. As can be seen, in the case of the *No Coev* sampling, the error rate was much higher (more than 500% higher) than in cases where we sampled values with co-evolutionary relations. Finally, usually the weighting that optimized the error-rate included the two sources of information (co-evolution, and the evolutionary trees), demonstrating that they are both important.

This analysis further supports the use of co-evolutionary information on top of conventional ML/MP approaches.

6. CONCLUSION

In this study, we formally described a new computational approach for reconstructing ancestral genomic sequences using information about co-evolution. Our model captures co-evolutionary dependencies between different proteins and uses this information to disambiguate the labels of the reconstructed ancestral genomes. We showed that this computational problem is NP-hard, and described algorithms for solving it. We demonstrated the performances of our approach by analyzing simulated input and by analyzing the ancestral genome content of the Fungi, showing that our approach can be used in practice and that it outperforms the convention ML/MP approaches.

In the future we intend to generalize this work in several ways. First, currently, our approach is presented in the setup of ancestral genome reconstruction, due to the importance of this problem and because co-evolutionary information can be readily obtained on the gene/protein level. However, it is important to note that the potential scope of our approach goes beyond the ancestral genome reconstruction problem, to tackle the more general problem of ancestral sequence reconstruction: that is, the reconstruction of different sites or domains in proteins, and even (provided that sufficient information is available) the reconstruction of single sites in DNA or RNA sequences (Yeang et al., 2007; Yeang and Haussler, 2007; Lockless and Ranganathan, 1999; Pedersen et al., 2006; Knudsen and Hein, 1999; Rzhetsky, 1995). In this setup, the success of such future applications depends on the existence of reliable co-evolutionary information on the individual site level. For example, information about the secondary structure of RNA sequences (Cannone et al., 2002) and proteins (Kabsch and Sander, 1983) can be used for determining what pairs of sites co-evolve.

Second, it is also clear that our approach can be generalized in the future to more complex reconstruction models for example, using non-binary alphabets (Tuller et al., 2009a), dependency between close sites, and various versions of maximum likelihood. Third, we intend to design algorithms that may compete with those described in this work. Specifically, we intend to check algorithms that are based on the belief propagation (Kschischang et al., 2001) approach. Fourth, we believe that a generalization of the approach described in this work can be used for *joint* inference of ancestral genomes and protein interactions or for *joint* inference of ancestral genomes and metabolic networks of ancestral and extant organisms.

Fifth, as we mentioned in Section 2, the model that we deal with in this article is a generalized parsimony and not a full probabilistic model (like Bayesian network or Markov network). Another open computational direction is to design a full probabilistic model for the problem of ancestral reconstruction that will take into account co-evolutionary relations.

Finally, in this study we assume that different proteins have an identical phylogenetic tree. In general, this assumption is not true; due to horizontal gene transfer (HGT) (Doolittle and Bapteste, 2007) and gene duplication events different orthologs may have different phylogenetic trees (Wapinski et al., 2007). However, at the level of entire genomes the "signal" of one (or a small number of) phylogenetic tree (the "tree of life") usually emerge (Ulitsky et al., 2006; Ge et al., 2005; Puigbo et al., 2009). In this study, we indeed deal with the organismal level: reconstruction of the gene content of ancestral genomes. We believe that co-evolutionary constraints play an important role also in the case of horizontal gene transfer (similarly to the cases of vertical inheritance). Thus, our approach should be even more useful when the analyzed organisms underwent many horizontal gene transfer events and/or gene duplications and there is no information about the different tree topologies of the different protein families.

Suppose the different protein families have different tree topologies and these different topologies are given. If we know how to embed the protein trees in the organisms tree we actually also know the solution of the ancestral reconstruction problem (the number of proteins from each family in each ancestral organism; see Fig. 8). If the embedding is not known, we believe that usage of co-evolutionary information can be useful also for solving the embedding problem (Page and Charleston, 1997)). Co-evolutionary

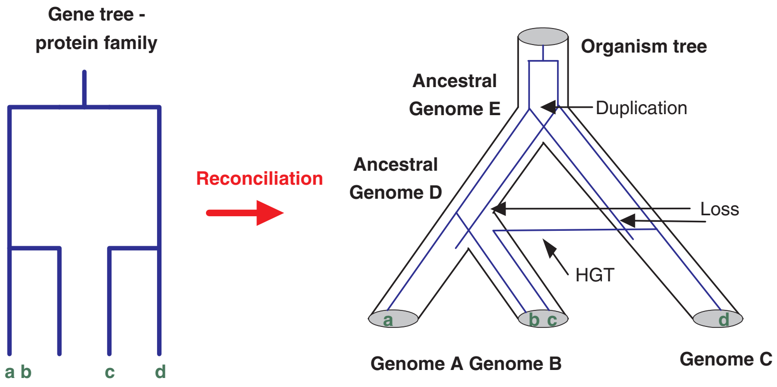


FIG. 8. The problem of embedding a gene tree in a species tree (or the tree reconciliation problem). In this problem we seek an explanation to the differences between the two trees (the gene tree and the species tree). In this figure, a gene tree with four leaves (*a*, *b*, *c*, *d*) is embedded in a species tree with three leaves (*A*, *B*, *C*), and two internal nodes (*D*, *E*). As can be seen, the solution to the embedding problem includes the solution to the ancestral genome problem (the number of proteins from each gene family in each ancestral organism); however, it includes additional information (full explanation to the differences between the gene tree and the species tree).

information can add constraints that are related to the number of proteins from each family in each ancestral organism; thus, it can "guide" the algorithms that search the full explanation for the evolution of a gene family within the species tree.

ACKNOWLEDGMENTS

T.T. is a Koshland Scholar at Weizmann Institute. M.K. and E.R. were supported by grants from the Israel Science Foundation.

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Barry, D., and Hartigan, J. 1987. Statistical analysis of humanoid molecular evolution. *Stat. Sci.* 2:191–210.

Benjamini, Y., and Hochberg, Y. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Statist. Soc.* 57:289–300.

Blanchette, M., Green, E.D., Miller, W., et al. 2004. Reconstructing large regions of an ancestral mammalian genome in silico. *Genome Res.* 14:2412–2423.

Cai, W., Pei, J., and Grishin, N.V. 2004. Reconstruction of ancestral protein sequences and its applications. *BMC Evol. Biol.* 4:e33.

Cannone, J.J., Subramanian, S., Schnare, M.N., et al. 2002. The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other rnas. *BioMed Central Bioinformatics* 3, 15.

Chor, B., Hendy, M.D., Holland, B.R., et al. 2000. Multiple maxima of likelihood in phylogenetic trees: an analytic approach. *Mol. Biol. Evol.* 17:1529–1541.

Cohen, O., Rubinstein, N.D., Stern, A., et al. 2008. A likelihood framework to analyse phyletic patterns. *Phil. Trans. R. Soc. Lond. B. Biol. Sci.* 363:3903–3911.

Csurös, M., and Miklós, I. 2009. Streamlining and large ancestral genomes in archaea inferred with a phylogenetic birth-and-death model. *Mol. Biol. Evol.* 26:2087–2095.

Doolittle, W.F., and Bapteste, E. 2007. Pattern pluralism and the tree of life hypothesis. *Proc. Natl. Acad. Sci. U.S.A.* 104:2043–2049.

Elias, I., and Tuller, T. 2007. Reconstruction of ancestral genomic sequences using likelihood. *J. Comput. Biol.* 14:216–237.

Barker, D., et al. 2007. Constrained models of evolution lead to improved prediction of functional linkage from correlated gain and loss of genes. *Bioinformatics* 23:14–20.

Pazos, F., et al. 1997. Correlated mutations contain information about protein-protein interaction. *J. Mol. Biol.* 271:511–523.

Wu, J., et al. 2003. Identification of functional links between genes using phylogenetic profiles. *Bioinformatics* 19:1524–1530.

Marino-Ramirez, L., et al. 2006a. Co-evolutionary rates of functionally related yeast genes. *Evol. Bioinform.* 2295–2300.

Jensen, L.J., et al. 2009. String 8–a global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Res.* 37:D412–D416.

Chena, Y., et al. 2006b. The coordinated evolution of yeast proteins is constrained by functional modularity. *Trends Genet.* 22:416–419.

Felder, Y., and Tuller, T. 2008. Discovering local patterns of co-evolution. *Proc. RECOMB-CG* 55–71.

Felsenstein, J. 1993. Phylip (phylogeny inference package) version 3.5c. Technical report. Department of Genetics, University of Washington, Seattle.

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

Fitch, W.M., and Margoliash, E. 1967. Construction of phylogenetic trees. *Science* 155:279–284.

Garey, M.R., and Johnson, D.S. 1979. *Computer and Intractability*. Bell Telephone Laboratories, New York.

Gaucher, E.A., Thomson, J.M., Burgan, M.F., et al. 2003. Inferring the palaeoenvironment of ancient bacteria on the basis of resurrected proteins. *Nature* 425:285–288.

Ge, F., Wang, L., and Kim, J. 2005. The cobweb of life revealed by genome-scale estimates of horizontal gene transfer. *PLoS Biol.* 3.

Hacia, J.G., Fan, J.B., Ryder, O., et al. 1999. Determination of ancestral alleles for human single-nucleotide polymorphisms using high-density oligonucleotide arrays. *Nat. Genet.* 22:164–167.

Håstad., J. 2001. Some optimal inapproximability results. *J. ACM* 48:798–859.

Hillis, D.M., Huelsenbeck, J.P., and Cunningham, C.W. 1994. Application and accuracy of molecular phylogenies. *Science* 264:671–677.

Hudek, A.K., and Brown, D.G. 2005. Ancestral sequence alignment under optimal conditions. *BMC Bioinform.* 6:273.

Jermann, T.M., Opitz, J.G., Stackhouse, J., et al. 1995. Reconstructing the evolutionary history of the artiodactyl ribonuclease superfamily. *Nature* 374:57–59.

Jin, G., Nakhleh, L., Snir, S., et al. 2006. Maximum likelihood of phylogenetic networks. *Bioinformatics* 22:2604–2611.

Juan, D., Pazos, F., and Valencia, A. 2008. High-confidence prediction of global interactomes based on genome-wide coevolutionary networks. *Proc. Natl. Acad. Sci.* U.S.A. 105:934–939.

Jukes, T.H., and Cantor, C.R. 1969. Evolution of protein molecules. In Munro, H.N., ed. *Mammalian Protein Metabolism.* Academic Press, New York.

Kabsch, W., and Sander, C. 1983. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22:2577–2637.

Knudsen, B., and Hein, J. 1999. RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics* 15:446–454.

Koshi, M., and Goldstein, R. 1996. Probabilistic reconstruction of ancestral protein seuences. *JME* 42:313–320.

Krishnan, N.M., Seligmann, H., Stewart, C. et al. 2004. Ancestral sequence reconstruction in primate mitochondrial dna: Compositional bias and effect on functional inference. *MBE* 21:1871–1883.

Kschischang, F.R., Frey, B.J., and Loeliger, H. 2001. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory* 47:498–519.

Li, G., Steel, M., and Zhang, L. 2008. More taxa are not necessarily better for the reconstruction of ancestral character states. *System. Biol.* 57:647–653.

Lockless, S.W., and Ranganathan, R. 1999. Evolutionarily conserved pathways of energetic connectivity in protein families. *Science* 286:295–299.

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

MacQueen, J.B. 1967. Some methods for classification and analysis of multivariate observations. *Proc. 5th Berkeley Symp. Math. Statist. Probabil.* 1:281–297.

Neyman, J. 1971. Molecular studies of evolution: A source of novel statistical problems, 127. In Gupta, S., and Jackel, Y., eds. *Statistical Decision Theory and Related Topics*. Academic Press, New York.

Ouzounis, C.A., Kunin, V., Darzentas, N., et al. 2006. A minimal estimate for the gene content of the last universal common ancestor–exobiology from a terrestrial perspective. *Res. Microbiol.* 157:57–68.

Page, R.D., and Charleston, M.A. 1997. From gene to organismal phylogeny: reconciled trees and the gene tree/species tree problem. *Mol. Phylogenet. Evol.* 7:231–240.

Pagel, M. 1999. The maximum likelihood approach to reconstructing ancestral character states of discerete characters on phylogenies. *System. Biol.*, 48:612–622.

Pedersen, J.S., Bejerano, G., Siepel, A., et al. 2006. Identification and classification of conserved rna secondary structures in the human genome. *PLoS. Comput. Biol.* 2:e33.

Puigbo', P., Wolf, Y.I., and Koonin, E.V. 2009. Search for a "tree of life" in the thicket of the phylogenetic forest. *J. Biol.* 8:59.

Pupko, T., Peer, I., Shamir, R., et al. 2000. A fast algorithm for joint reconstruction of ancestral amino acid sequences. *Mol. Biol. Evol.* 17:890–896.

Rascola, V.L., Pontarottia, P., and Levasseura, A. 2007. Ancestral animal genomes reconstruction. *Curr. Opin. Immunol.* 19:542–546.

Rzhetsky, A. 1995. Estimating substitution rates in ribosomal RNA genes. *Genetics* 141:771–783.

Sankoff, D. 1975. Minimal mutation trees of sequences. *SIAM J. Appl. Math.* 28:35–42.

Sato, T., Yamanishi, Y., Kanehisa, M., et al. (2005). The inference of proteinprotein interactions by co-evolutionary analysis is improved by excluding the information about the phylogenetic relationships. *Bioinformatics* 21:3482–3489.

Tauberberger, J.K., Reid, A.H., Lourens, R.M., et al. 2005. Characterization of the 1918 influenza virus polymerase genes. *Nature* 437:889–893.

Thornton, J.W., Need, E., and Crews, D. 2003. Resurrecting the ancestral steroid receptor: ancient origin of estrogen signaling. *Science* 301:1714–1717.

Toda, T., Cameron, S., Sass, P., et al. 1987. Three different genes in *S. cerevisiae* encode the catalytic subunits of the cAMP-dependent protein kinase. *Cell* 50:277–287.

Tuller, T., Birin, H., Gophna, U., et al. 2009a. Reconstructing ancestral gene content by co-evolution. *Genome Res.* Nov 30. [Epub ahead of print].

Tuller, T., Kupiec, M., and Ruppin, E. 2009b. Co-evolutionary networks of genes and cellular processes across fungal species. *Genome Biol.* 10.

Ulitsky, I., Burstein, D., Tuller, T., et al. 2006. The average common substring approach to phylogenomic reconstruction. *J. Comput. Biol.* 13:336–350.

Wapinski, I., Pfeffer, A., Friedman, N., et al. 2007. Natural history and evolutionary principles of gene duplication in fungi. *Nature* 449:54–61.

Yang, Z. 1997. PAML: a program package for phylogenetic analysis by maximum likelihood. *Comput. Appl. BioSci.* 13:555–556.

Yang, Z., Kumar, S., and Nei, M. 1995. A new method of inference of ancestral nucleotide—and amino acid sequences. *Genetics* 141:1641–1650.

Yeang, C.H., and Haussler, D. 2007. Detecting coevolution in and among protein domains. *PLoS Comput. Biol.* 3:e211.

Yeang, C.H., Darot, J.F., Noller, H.F., et al. 2007. Detecting the coevolution of biosequences—an example of RNA interaction prediction. *Mol. Biol. Evol.* 24:2119–2131.

Yu Gorbunov, K., Lyubetskaya, E.V., Asarin, E.A., and Lyubetsky, V.A. 2009. Modeling evolution of the bacterial regulatory signals involving secondary structure. *Mol. Biol.* 43.

Zhang, J., and Rosenberg, H.F. 2002. Complementary advantageous substitutions in the evolution of an antiviral rnase of higher primates. *Proc. Natl. Acad. Sci. USA.* 99:5486–5491.

Zou, Z., DiRusso, C.C., Ctrnacta, V., et al. 2002. Fatty acid transport in *Saccharomyces cerevisiae*: directed mutagenesis of fat1 distinguishes the biochemical activities associated with fat1p. *J. Biol. Chem.* 277:31062–31071.

Address correspondence to:
Dr. Tamir Tuller
Faculty of Mathematics and Computer Science
Weizmann Institute of Science
Rehovot, Israel

E-mail: tamir.tuller@weizmann.ac.il