BMC Evolutionary Biology

BioMed Central

Methodology article

Open Access

An ant colony optimization algorithm for phylogenetic estimation under the minimum evolution principle

Daniele Catanzaro[1], Rafflaele Pesenti[2] and Michel C Milinkovitch*[1]

Address: [1]Laboratory of Evolutionary Genetics, Institute for Molecular Biology and Medicine (IBMM), Université Libre de Bruxelles (U.L.B.), CP300, Rue Jeener et Brachet 12, B-6041, Gosselies, Belgium and [2]Dipartimento di Matematica Applicata, Universitá Ca' Foscari, Dorsoduro 3246 - 30123, Venice, Italy

Email: Daniele Catanzaro - dacatanz@ulb.ac.be; Rafflaele Pesenti - pesenti@units.it; Michel C Milinkovitch* - mcmilink@ulb.ac.be

* Corresponding author

Published: 15 November 2007

BMC Evolutionary Biology 2007, **7**:228 doi:10.1186/1471-2148-7-228

Received: 4 June 2007

Accepted: 15 November 2007

This article is available from: http://www.biomedcentral.com/1471-2148/7/228

© 2007 Catanzaro et al; licensee BioMed Central Ltd.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Background: Distance matrix methods constitute a major family of phylogenetic estimation methods, and the minimum evolution (ME) principle (aiming at recovering the phylogeny with shortest length) is one of the most commonly used optimality criteria for estimating phylogenetic trees. The major difficulty for its application is that the number of possible phylogenies grows exponentially with the number of taxa analyzed and the minimum evolution principle is known to belong to the $\mathcal{NP}$-hard class of problems.

Results: In this paper, we introduce an Ant Colony Optimization (ACO) algorithm to estimate phylogenies under the minimum evolution principle. ACO is an optimization technique inspired from the foraging behavior of real ant colonies. This behavior is exploited in artificial ant colonies for the search of approximate solutions to discrete optimization problems.

Conclusion: We show that the ACO algorithm is potentially competitive in comparison with state-of-the-art algorithms for the minimum evolution principle. This is the first application of an ACO algorithm to the phylogenetic estimation problem.

Background

The Minimum Evolution (ME) principle is a commonly used principle to estimate phylogenetic trees of a set Γ of n species (taxa) given an $n \times n$ symmetric matrix $\mathbf{D} = \{d_{ij}\}$ of evolutionary distances. First introduced by Kidd and Sgaramella-Zonta [1] and subsequently reinterpreted by Rzhetsky and Nei [2,3], the ME principle aims at finding a phylogeny characterized by minimal sum of branch lengths, under the auxiliary criteria that branches have a positive length and the pair-wise distances on the tree are not smaller than the directly observed pair-wise differences. Its biological justification is based on the fact that, when unbiased estimates of the true distances are available, the correct phylogenetic tree has an expected length shorter than any other possible tree [2,3] compatible with the distances in $\mathbf{D}$. More formally, the ME principle can be expressed in terms of the following optimization problem:

Problem 1. *Minimum Evolution under Least Square (LS)*

$$\min_{(\mathbf{X},\mathbf{v})} \quad \|\mathbf{v}\|_1$$
$$s.t. \quad f(\mathbf{D}, \mathbf{X}, \mathbf{v}) = 0$$
$$\mathbf{X} \in \mathcal{X}$$

where $||\cdot||_1$ is the $\mathcal{L}^1$-vector norm; $\mathbf{v}$ is a vector of the $2n$ - 3 edge lengths; $\mathbf{X}$ is a $n(n - 1)/2 \times (2n - 3)$ *topological matrix*[4] encoding a phylogenetic tree as an unrooted binary tree with the n taxa in Γ as terminal vertices (*leaves*); $\mathcal{X}$ is the set of all the topological matrices; finally, $f(\cdot, \cdot, \cdot)$ defines the level of compatibility among the distances in $\mathbf{D}$ and the distances induced by the phylogenetic tree edges. Any *optimal solution* $(\mathbf{X}^*, \mathbf{v}^*)$ of problem (1) defines a phylogenetic tree satisfying the minimum evolution principle. A topological matrix $\mathbf{X}$ is an *Edge-Path incidence matrix of a Tree* (EPT) (see [5], and additional files 1 and 2) that encodes a tree as follows: any generic entry $x_{ij,k}$ is set to 1 if the edge k belongs to the path from the leaf i to the leaf j, 0 otherwise. In the rest of the paper we refer to problem (1) as the *ME problem*.

The distance matrix $\mathbf{D}$ of problem (1) is estimated from the dataset, e.g., accordingly to any method described in [6-12]. Condition $f(\mathbf{D}, \mathbf{X}, \mathbf{v}) = 0$ typically imposes that, for any given EPT matrix $\mathbf{X}$, $\mathbf{v}$ minimizes the (weighted) sum of the square values of the differences between the distances in $\mathbf{D}$ and the corresponding distances induced by the phylogenetic tree edges [6,13]. In particular, under the unweighted least-square (also called Ordinary Least-Squares (OLS)) [2]:

$$\mathbf{v} = \mathbf{X}^{\dagger}\mathbf{D}^{\Delta} \tag{1}$$

where $\mathbf{X}^{\dagger}$ is the Moore-Penrose pseudoinverse of $\mathbf{X}$, and $\mathbf{D}^{\Delta}$ is a vector whose components are obtained by taking row per row the entries of the strictly upper triangular matrix of $\mathbf{D}$.

Others [14] and [15] have suggested the use of a Weighted Least-Squares (WLS) function:

$$\mathbf{v} = (\mathbf{X}^t\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{W}\mathbf{D}^{\Delta} \tag{2}$$

where $\mathbf{W}$ is a strictly positive definite diagonal matrix whose entries w_{ij} represent weights associated to leaves i and j. Finally, Hasegawa *et al*. [16] introduced a Generalized Least-Squares (GLS) function in which $\mathbf{v}$ is computed using:

$$\mathbf{v} = (\mathbf{X}^t\mathbf{C}^{-1}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{C}^{-1}\mathbf{D}^{\Delta} \tag{3}$$

where $\mathbf{C}$ is a strictly positive definite symmetric matrix representing the covariance matrix of $\mathbf{D}$. To avoid the occurrence of negative branch lengths [14,17], problem (1) can be modified as follows:

Problem 2. *Minimum Evolution under Linear Programming (LP)*

$$\begin{aligned} \min_{(\mathbf{X},\mathbf{v})} \quad & \|\mathbf{v}\|_1 & & \\ s.t. \quad & \mathbf{X}\mathbf{v} & \geq & \mathbf{D}^{\Delta} \\ & \mathbf{v} & \in & \mathbb{R}_+^{2n-3} \\ & \mathbf{X} & \in & \mathcal{X} \end{aligned}$$

Unfortunately, both problems (1) and (2) are $\mathcal{NP}$-hard [18]. In this context, let us observe that, given Γ, the cardinality of $\mathcal{X}$ is:

$$|\mathcal{X}| = (2|\Gamma| - 5)!! = (2n - 5)!! \tag{4}$$

where $n!!$ is the double factorial of n. Hence, the number of topological matrices grows exponentially with the number of leaves ([6], p. 25, and see additional files 1 and 2).

Problem (1) has received great attention from the scientific community such that exact and approximate algorithms to solve it have been developed. Exact algorithms for solving problem (1) are typically based on an exhaustive approach (i.e., enumerating all possible trees $\mathbf{X}$). As an example, PAUP* 4.0 [19] allows exhaustive search for datasets containing up to 12 taxa. A number of heuristics were also developed in the last 20 years. E.g., Rzhetsky and Nei [2,3] (i) start from a Neighbor-Joining (NJ) tree [20,21], (ii) apply a local search generating topologies within a given *topological distance* (see [2]) from the NJ tree, and (iii) return the best topology found. Kumar [22] further improved the approach as follows: starting from a topology, a leaf l is selects at each step and all possible assignments of l on the topology are tested. Despite that the neighborhood size in Kumar's approach is larger than in Rzhetsky and Nei's algorithm, it requires examining a number of topologies that is, at most, an exponential function of the number of leaves n: $(n - 1)!/2$, and generates solution in a shorter computing time. Finally, Bryant and Waddell [4] implemented programming optimisation and Desper and Gascuel [23] introduced a greedy search that both improved speed and accuracy of the search.

Here, we introduce the Ant Colony Optimization (ACO) algorithm for estimating phylogenies under the minimum evolution principle, and show that ACO has the potential to compete with other widely-used methods.

ACO (see [24,25] for an introduction, and [26,27] for recent reviews) is a widely-used metaheuristic approach for solving hard combinatorial optimization problems. ACO is inspired from the pheromone trail laying and following behavior of real ants. ACO implements indirect communication among simple agents, called (artificial) ants. Communication is mediated by (artificial) pheromone trails implemented as a probabilistic model to which the ants adapt during the algorithm's execution to reflect their search experience. ACO has proven a successful technique for numerous $\mathcal{NP}$-hard combinatorial optimization problems (see [28]), although no application to the ME phylogeny problem is currently known. Our specific implementation of the ACO algorithm exploits a stochastic version of the Neighbor-Joining (NJ) algorithm [20,21] to explore tree space.

Results and Discussion

Iterative addition

Given a set Γ of taxa, let us define a *partial tree* as a m-leaf tree whose leaves are taxa of a subset $\Gamma' \subset \Gamma$, with $m = |\Gamma'|$. Moreover, given a partial tree, with node set V and edge set E, let us say that we *add/insert a leaf* i (not yet in Γ') on the edge $(r, s) \in E$ (i.e., the edge joining the nodes $r, s \in V$), and generate a new partial tree with node set $\hat{V} = V \cup \{i, t\}$ and edge set $\hat{E} = E \cup \{(r, t), (t, s), (t, i)\} \setminus \{(r, s)\}$. In other words, we add a leaf i on an edge, divide that edge with a new node t, and join the leaf i to t. All algorithms described here build complete phylogenetic trees by iteratively adding one leaf at a time on the edges of a partial tree.

Primal bound

To generate a first upper bound [5] of the ME problem, we adapted the *Sequential Addition* (SA) greedy algorithm [6]. The Sequential Addition algorithm is less prone, than NJ, to generate a systematic local optimum at the end of the search (i.e., starting from "too good" a primal bound may lead to inefficient results [29]).

The pseudo-code of our version of the Sequential Addition algorithm is presented in Figure 1. In the initialization step, we arbitrarily chose a subset $\Gamma' \subseteq \Gamma$ of $m \leq n$ leaves, and we generate as initial m-leaf partial tree, i.e., an optimal solution of the problem (1) when only m leaves are considered. A each iteration, we join the leaf i to all possible leaves already present in Γ', and choose the solution that minimize tree length (we break possible ties randomly), hence, generating a new partial tree and new set $\Gamma' = \Gamma' \cup \{i\}$. We iterate the procedure until a tree with n leaves is obtained. Finally, fixing the topology matrix $\hat{\mathbf{X}}_m$, we determine the optimal edge weights by imposing $f(\mathbf{D}, \hat{\mathbf{X}}_m, \mathbf{v} = 0$, and return the length of the tree, i.e., the upper bound on the optimal solution of the ME problem.

Unfortunately, the computation complexity of our heuristic is $O((2m - 5)!! + n(n - m)^2)$. AT each iteration, given a partial tree, i.e., a k-leaf phylogenetic tree of the leaves in $\Gamma' \subset \Gamma$ with $k = |\Gamma'|$, and a leaf i not in Γ', the procedure generates all the different $(k + 1)$-leaf partial trees that can be obtained by adding the leaf i in each edge of the current partial tree.

The ant colony optimization algorithm

The specific ACO algorithm for the minimum evolution problem (hereafter ACO-ME) that we introduce here (cf. pseudo-code in Figure 2), is a hybrid between the Max-Min Ant System (MMAS) [30,31] and the Approximate Nondeterministic Tree Search (ANTS) [32]. Both methods are modifications of the original Ant System approach [33].

The core of the ACO-ME algorithm is the iteration phase, where the ants generate a set $\mathcal{T}$ of trees. Then, starting from the trees in $\mathcal{T}$, a local search is performed until a locally optimal tree is found and compared with the current-best tree. If stopping conditions are met the procedure ends, otherwise the iteration phase is repeated.

Each ant builds a phylogenetic tree by iteratively adding a leaf at a time to a partial tree. Following a *relation-learning* model [34], the choices performed by an ant about (i) which leaf to insert, and (ii) where to add it on the partial tree are based of a set of parameters $\{\tau_{ij}\}$ called *pheromone trails*. The values of the pheromone trail parameters $\{\tau_{ij}\}$ represent a stochastic desirability that a leaf i shares a direct common ancestor with a vertex j on a partial tree. The ants generate a new set $\mathcal{T}$ of trees and the pheromone trail parameters are updated at the end of the main iteration phase.

Let us now consider the algorithm in more details. It uses two identical data structures: s^* and s_k. The former stores the current-best complete *reconstruction* (solution) known, whereas the latter stores the best complete reconstruction obtained by the ants during iteration k. The algorithm also uses a variable n_a, i.e., the number of artificial ants. How to set the value of n_a is discussed in the Parameter settings section. In the initialization phase, s^* is first set to the reconstruction obtained by the Sequential Addi

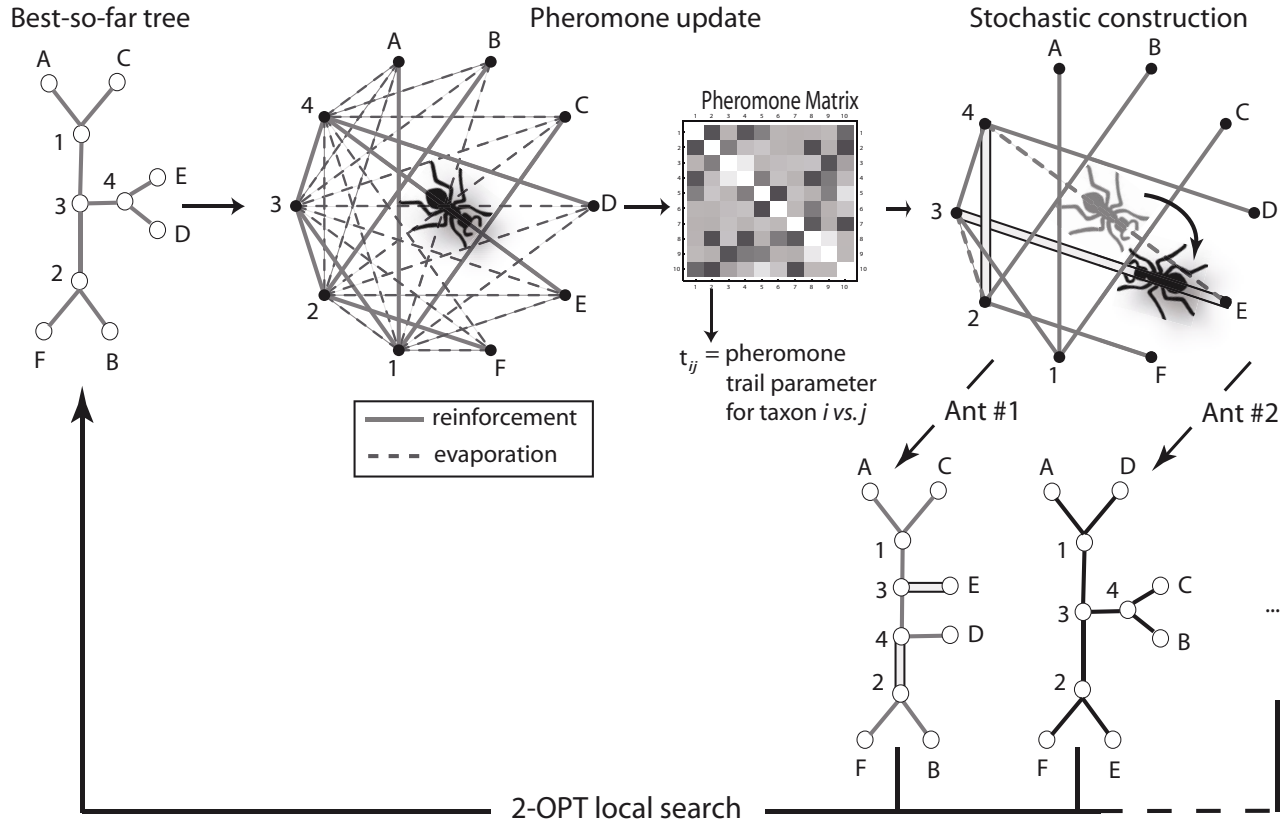

Figure 1
Principle of the ACO-ME algorithm. The core iteration includes three main steps: (i) the pheromone update phase during which artificial ants walk on a graph with all possible connections among the *n* taxa and (*n* - 2) internal nodes, and lay a trail of volatile pheromone on the branches of the starting tree; (ii) the stochastic construction phase during which new trees are built using both the heuristic information of the pairwise distances and the stochastic process guided by the newly-updated pheromone trail matrix (ants follow a given edge with a probability which is a function of the amount of pheromone on that edge); and (iii) the 2-OPT local search phase that corresponds to a local search using taxon swapping. The curved arrow indicates the stochastic jump of an ant from one edge to another. See text for details.

procedure `SequentialAddition`(m: integer, Γ : set)
$\Gamma' =$ an arbitrary subset of m leaves in Γ
$k = m$
$\hat{\mathbf{X}}_k =$ topological matrix of the phylogenetic tree for the leaves in Γ'
while $|\Gamma'| \leq |\Gamma|$ **do**
  $(i, j) = arg\min\{d_{ij} : i \in \Gamma \setminus \Gamma', j \in \Gamma'\}$
  $\hat{\mathbf{X}}_{k+1} =$ topological matrix of the tree obtained adding a leaf i on the external edge j of the tree defined by $\hat{\mathbf{X}}_k$
  $k = k + 1$
  $\Gamma' = \Gamma' \cup \{i\}$
end while
$\mathbf{v} =$ solution of the compatibility conditions $f(\mathbf{D}, \hat{\mathbf{X}}_m, \mathbf{v}) = 0$
return $|\mathbf{v}|_1$

Figure 2
High-level pseudo-code for the Sequential Addition heuristic.

tion algorithm and s_k is set to null, then the pheromone trail parameters are updated. We implemented the MMAS [30,31] method of pheromone update, where $\tau_{min} \leq \tau_{ij} \leq \tau_{max}$. Here, we set τ_{min} and τ_{max} to 0.0001 and 0.9999, respectively [35]. In the initialization phase, the pheromone trail parameters $\{\tau_{ij}\}$ are set to 0.5, i.e., all positions for leaf insertion have the same desirability.

Before describing the iteration phase, let us introduce some definitions. Let $\mathcal{G}_k$ be a partial tree with k leaves, $V(\mathcal{G}_k)$ the set of vertices of $\mathcal{G}_k$, and $\Gamma_{\mathcal{G}_k}$ the set of leaves of $\mathcal{G}_k$. Let us also use the *recursive distance* definition of [23,36]: if A and B are two non-intersecting subtrees from a tree $\mathcal{G}$, then the average distance between A and B is:

$$\Delta_{A|B} = \frac{1}{|A||B|} \sum_{i \in A, j \in B} d_{ij}. \quad (5)$$

In the iteration phase, each artificial ant r generates a complete phylogenetic tree using the ConstructCompleteReconstruction(r) procedure, as illustrated in Figure 3: ant r randomly selects four leaves from the set Γ, and builds a partial tree $\mathcal{G}_k$, k = 4, then, ant r (i) chooses, among the leaves not yet inserted in the partial topology, the leaf i defining the smallest distance d_{ij}, $j \in \Gamma_{\mathcal{G}_k}$, and (ii) computes the probability that i has a common ancestor with the vertex $j \in V(\mathcal{T}_k)$ using the formula suggested by ANTS [32]:

Procedure ACO-ME
  input: a set of leaves Γ, and the corresponding distance matrix **D**
  output: The best solution found s^*
  (* **comment**: initialization phase *)
  Choose $m \leq n = |\Gamma|$
  s^* =Sequential Addition(m, Γ)
  $s_k = NULL$
  n_a=DetermineNumberOfAnts()
  InitializePheromoneValues()
  (* **comment**: end initialization phase *)
  (* **comment**: main iteration phase *)
while not stop condition satisfied **do**
  $\mathcal{T} = \{\}$ (**comment**: $\mathcal{T}$ is a forest of complete solutions)
    (* **comment**: construction phase *)
  for ant=1 to n_a **do**
    $\mathcal{T} = \mathcal{T} \bigcup$ ConstructCompleteReconstruction(ant)
  end for
    (* **comment**: end construction phase *)
  2-OPT($\mathcal{T}$) (**comment**: Apply a 2-OPT local search to $\mathcal{T}$)
  $s_k = argmin\{f(t)|t \in \mathcal{T}\}$
  Update(s_k,s^*)
  UpdatePheromone(s_k,s^*)
end while
  (* **comment**: end main iteration phase *)
end ACO-ME

Figure 3
High-level pseudo-code for the ACO algorithm.

$$p_{ij} = \frac{\alpha\tau_{ij} + (1-\alpha)\eta_{ij}}{\sum_{q \in \Gamma \setminus \Gamma_{\mathcal{G}_k}} [\alpha\tau_{qj} + (1-\alpha)\eta_{qj}]} \quad (6)$$

where η_{ij} represents the "heuristic desirability" that leaf i shares a common ancestor with a vertex j of $V(\mathcal{G}_k)$ (whereas τ_{ij} represents the corresponding "stochastic desirability"). Finally, $\alpha \in [01]$ allows the relative weighting of heuristic and stochastic desirabilities. The heuristic desirability η_{ij} is computed as:

$$\eta_{ij} = (\Delta_{ij} - u_i - u_j)^{-1} \quad (7)$$

where $u_i = \sum_{j \notin V_{\mathcal{G}_k}} \Delta_{ij} / (|V_{\Gamma_{\mathcal{G}_k}}|)$, i.e., the sum of the distances from i to the leaves not yet inserted in the partial tree divided the number of leaves inserted in the partial tree.

Note that η_{ij}, Δ_{ij}, and u_i correspond to the quantities used in the Neighbor-Joining algorithm[20,21](see also[23]). Hence, computation of the vector $\mathbf{p}_i = \{p_{ij}\}$, for all $i \in \Gamma$, can be interpreted as the stochastic application of the Neighbor-Joining algorithm. A possible problem (not observed yet in practice in our analyses) is that η_{ij} can take negative values. Finally, ant r randomly chooses a vertex j on the basis of the probabilities $\mathbf{p}_i$, and the leaf i is added to the tree.

At the end of the construction phase, a set $\mathcal{T}$ of trees is obtained and a 2-OPT local search (with best-improvement and without candidate list [37,38]) is iteratively performed on each tree: two randomly-chosen leaves are swapped and the tree length is evaluated. Swap i is performed on the new tree if swap i-1 generated an improvement, otherwise it is performed on the old tree. To reduce the 2-OPT computational overhead, we perform no more than 10 swappings on each tree in $\mathcal{T}$. If the best tree generated by the 2-OPT local search is shorter than the tree in s^*, both s^* and s_k are updated, otherwise only s_k is updated.

The pheromone update completes the iteration phase: each entry τ_{ij} is updated following:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \varepsilon_{ij} \quad (8)$$

where

$$\varepsilon_{ij} = \begin{cases} \kappa\rho / l_{\mathcal{G}_{|\Gamma|}}, & \text{if } w_i \text{ adjacent to } w_j \text{ in } s^{best}; \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

where $\kappa \in \mathbb{R}$ and ρ, the *pheromone evaporation rate*, are two tuning constants, s^{best} is one of the tree s^* or s_k (see below), and $l_{\mathcal{G}_{|\Gamma|}}$ the length of s^{best}. When applying equation (8), if τ_{ij} is greater than τ_{max} or smaller than τ_{min}, then its value is set to τ_{max} or τ_{min}, respectively. We set to ρ 0.1, κ to $\kappa\rho \in [10^{-2}, 10^{-1}]$, and α to 0.7. Fine-tuning of these parameters might have a significant impact on search efficiency but such a systematic analysis is out of the scope of a proof-of concept for the use of ACO-ME. Finally, if the objective function does not decrease after 30 iteration, ACO-ME chooses s_k as s^{best} instead of s^* for the pheromone updating; if the objective function does not decrease after 30 additional iterations, then all $\{\tau_{ij}\}$ are reset to 0.5 and s^* is used for pheromone updating.

Parameter settings

We evaluated the performances of the ACO-ME algorithm under different values of the parameter κ(0.1, 0.5, and 1), and different numbers of ants (1 to 10). For each of the 30 possible combinations of these parameters values, we run ACO-ME for 1000 iterations. As suggested elsewhere (see [29]), we do not consider colony sizes larger than 10.

Relative performances are measured using a normalized index as in [39-41]:

$$I_j^k = \frac{x_j^{(x)} - x_j^{min}}{x_j^{max} - x_j^{min}} \quad (10)$$

where $x_j^{(k)}$ is the best solution found under parameter value k using dataset j, whereas x_j^{min} and x_j^{max} are respectively the best and worst solutions found on the instance j using the parameter value k. By definition, performance index values are in the interval [0, 1]. The optimal parameter value exhibits the smallest relative performance index (see box-and-whisker plot histograms in Figure (4, 5, 6). Figures 4, 5, and 6 indicate that, for small, medium, and large datasets, the optimal combinations of number of ants/κ are 7/1, 10/0.5, and 8/0.5, respectively. However, differences of performances are not spectacular among different combinations of parameter values (except that

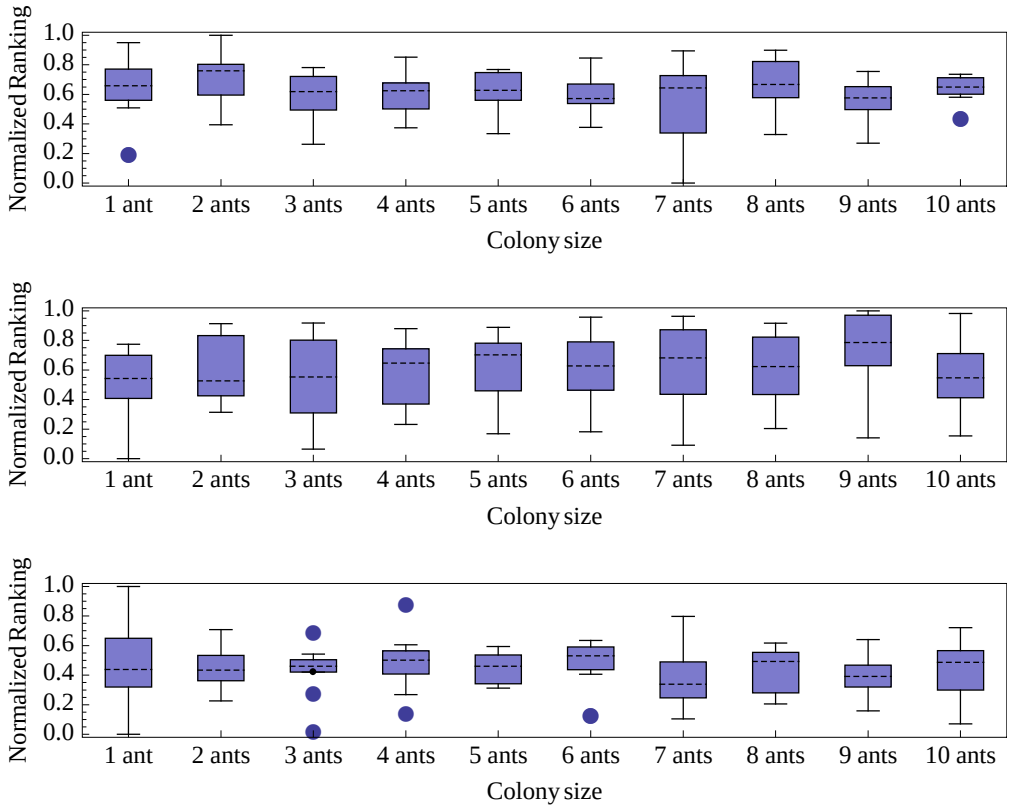

Figure 4
Normalized ranking of the ACO algorithm performances with small datasets (20 taxa) and $\kappa = 0.1$ (a), $\kappa = 05$ (b), and $\kappa = 1$ (c) versus colony size n_a.

performances are generally very low when a single at is used).

Experimental evaluation

We first used a set of distance matrices generated from real datasets: the dataset "551314.nex" that includes 55 RBCL sequences of 1314 nucleotides each, and the dataset "Zilla500.nex" that includes 500 RBCL sequences of 1428 nucleotides each. These datasets are available at [42]. Note that sequences in these datasets were aligned using ClustalX [43] and columns including gaps were excluded before computing pairwise distances. Second, we generated (i) 10 artificial instances of 20 taxa (also called *small instances*); (ii) 10 artificial instances of 50 taxa (also called *medium instances*); and (iii) 10 artificial instances of 100 taxa (also called *large instances*). Each artificial instance was generated by random sampling of taxa and partial character reshuffling of the Zilla500.nex data set. More explicitly, after random selection of the 20 or 50 or 100 taxa, we randomly reshuffled characters among taxa, for 50 percents of the aligned columns. As the reshuffling makes the dataset prone to yield undefined pairwise distances [44], we simply used the absolute number of differences between sequence pairs for generating the distance matrix. Edge lengths were computed using the standard OLS because WLS and GLS can potentially lead to inconsistent results *et al.* [45]. All numerical experiments were performed on a workstation Apple 64-bit Power Mac G5 dual processor dual core, with 8 Gb of RAM, and OS X. The ACO-ME source code is written in C/C++ and compiled using IBM XL C/C++ compiler version 6.0. We compared the quality (total length) of trees generated by the ACO-ME algorithm to those obtained using a classical hill-climbing algorithm (implemented in PAUP* 4.0 [19]) after a fixed run time of 1 minute. The starting tree was generated using the Neighbor-Joining algorithm [20,21], and the TBR branch-swapping operator [6] was used for exploring the solution space. PAUP* 4.0 was used with and without the "Steepest Descent" (SD) option. When SD is activated, all possible TBR are tried, and the rearrangement producing the largest decrease in tree length is selected, inducing a computational overhead similar to that of the 2-OPT local search implemented in our ACO-ME algorithm. Each algorithm was run 30 times on each of the two real datasets. Figure 7a and 7b show that ACO-ME performances are intermediate between hill-climbing with SD, and hill-climbing without SD. Fur-

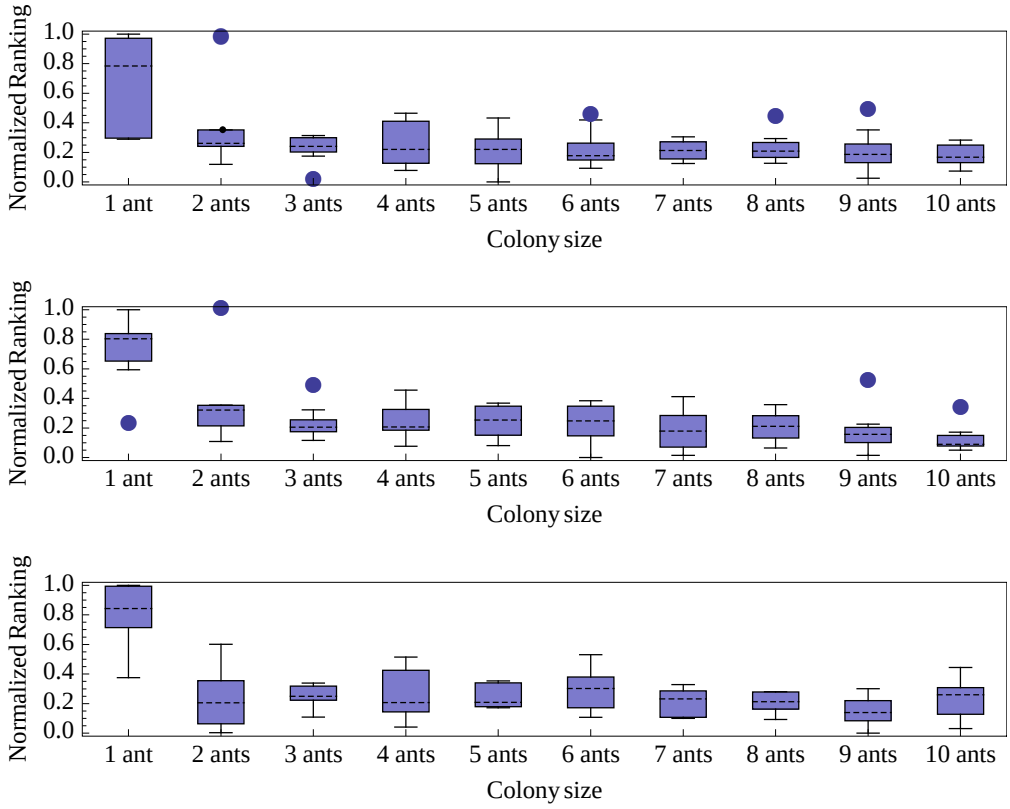

Figure 5
Normalized ranking of the ACO algorithm performances with medium datasets (50 taxa) and $\kappa = 0.1$ (a), $\kappa = 0.5$ (b), and $\kappa = 1$ (c) versus colony size n_a.

thermore, Figure 7a and 7b indicate that the relative performances of ACO-ME, in comparison to hill climbing, increase with larger datasets. Note that, contrary to our simple implementation of ACO-ME, the implementation of ME in PAUP* 4.0 [19] incorporates procedures [4,23] that greatly speed-up the OLS (reaching a complexity $O(n^2)$). We trust that implementation of these procedures in combination with further tuning of the ACO parameters (number of ants, relative weights of the heuristic information and stochastic pheromone parameters, etc) would lead to better performances of the ACO-ME algorithm. Figure 8a and 8b indicate that the relative performances described above are relatively stable trough time, especially for large data sets (at any time during the run, ACO-ME has similar performances than "hill-climbing without SD" and better performances than "hill-climbing with SD").

Conclusion

We introduce here an Ant Colony Optimization algorithm (ACO) for the phylogeny estimation problem under the minimum evolution principle and demonstrate the feasibility of this approach. Although much improvement in performances can probably be obtained through (i) modification of the local search phase, (ii) tuning of the ACO parameters (number of ants, relative weights of the heuristic information and stochastic pheromone parameters, etc), and (iii) implementation of speed-up procedures and optimization of the code, the current implementation of our algorithm already demonstrates that the ant colony metaphor can efficiently solve instances of the phylogeny inference problem.

Authors' contributions

All authors read and approved the final manuscript. Daniele Catanzaro, Raffaele Pesenti, and Michel C. Milinkovitch conceived the study and wrote the manuscript, Daniele Catanzaro performed the numerical analyses.

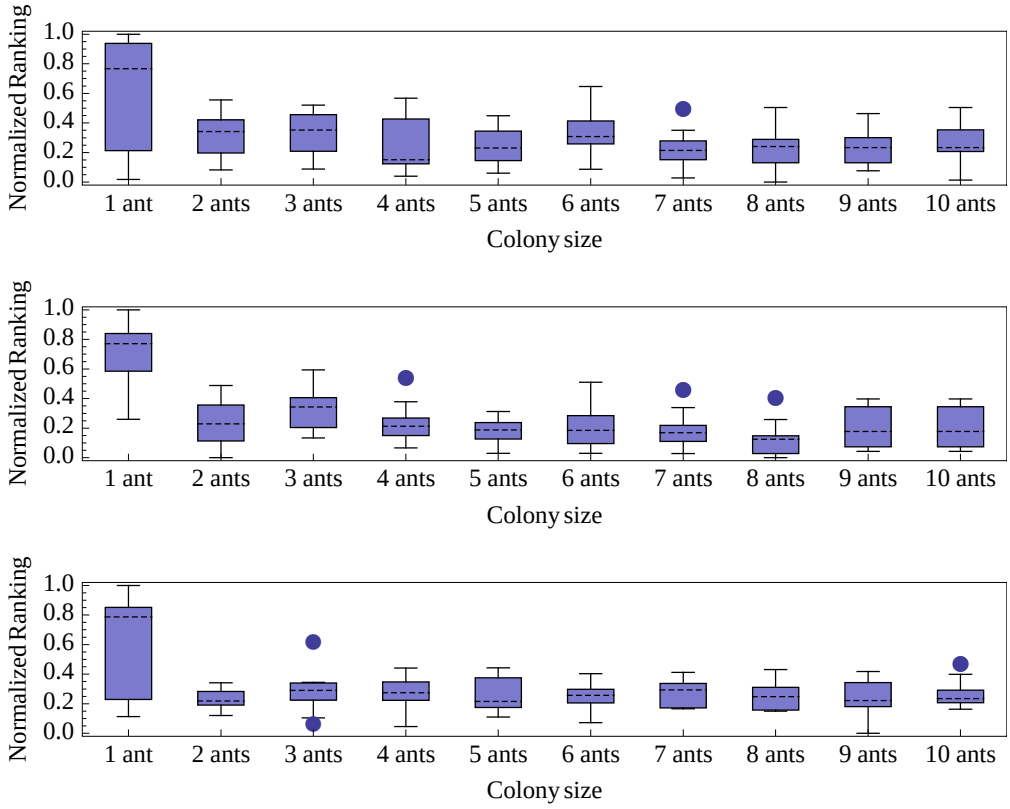

Figure 6
Normalized ranking of the ACO algorithm performances with large datasets (100 taxa) and $\kappa = 0.1$ (a), $\kappa = 0.5$ (b), and $\kappa = 1$ (c) versus colony size n_a.

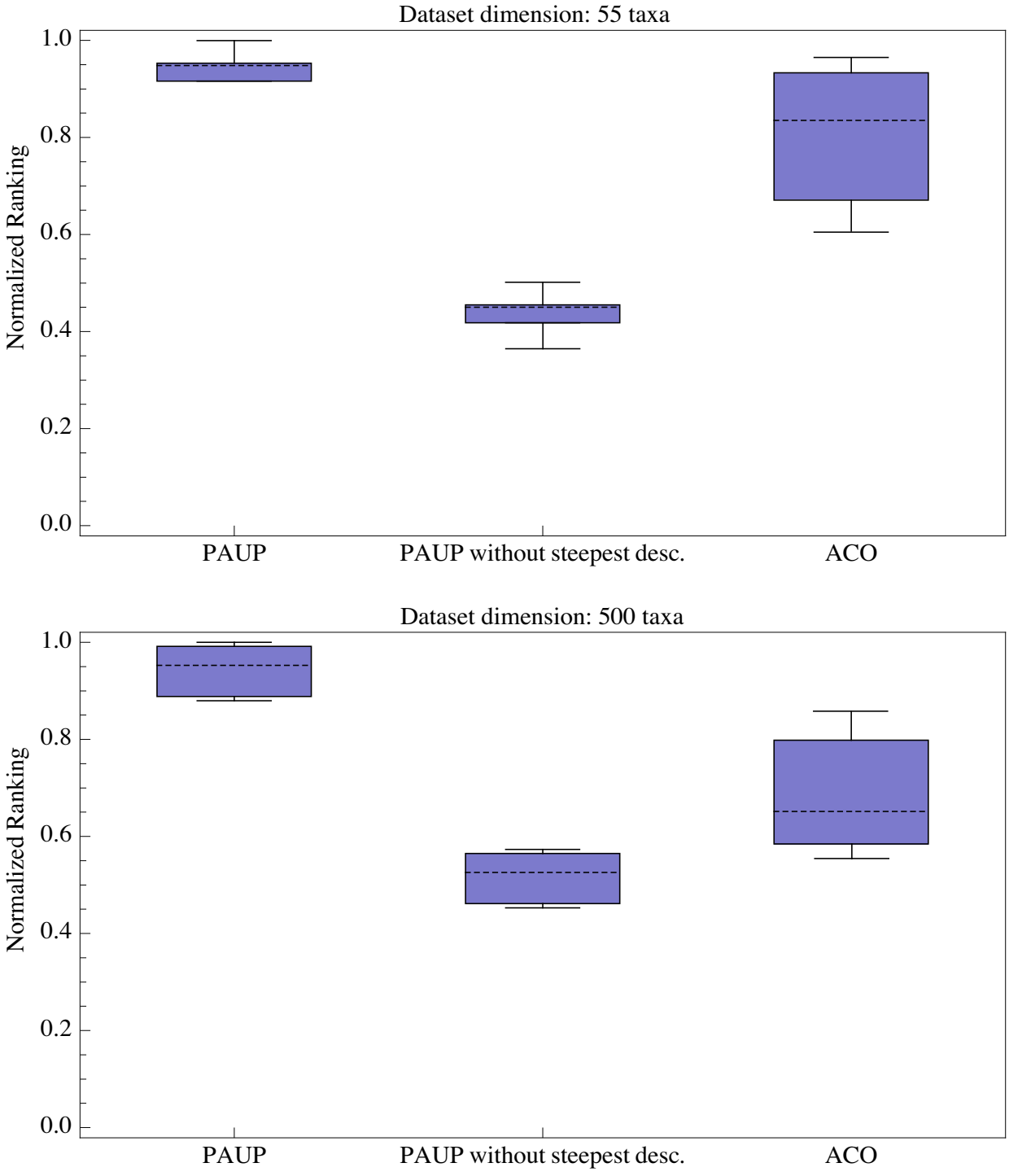

Figure 7
Comparison of performances between ACO-ME and hill-climbing (with and without Steepest Descent, SD) after a fixed run time of 1 minute on datasets of 55 (a) and 500 (b) taxa. A paired Wilcoxon test indicates that ACO-ME performances are significantly better (p-value = $3.92e^{-2}$ for 55 taxa dataset, and p-value = $6.821e^{-4}$ for 500 taxa dataset) than those of hill-climbing with SD, but significantly worst (p-value = $4.71e^{-3}$ for 55 taxa dataset, and p-value = $4.53e^{-4}$ for 500 taxa dataset) than those of hill-climbing without SD.

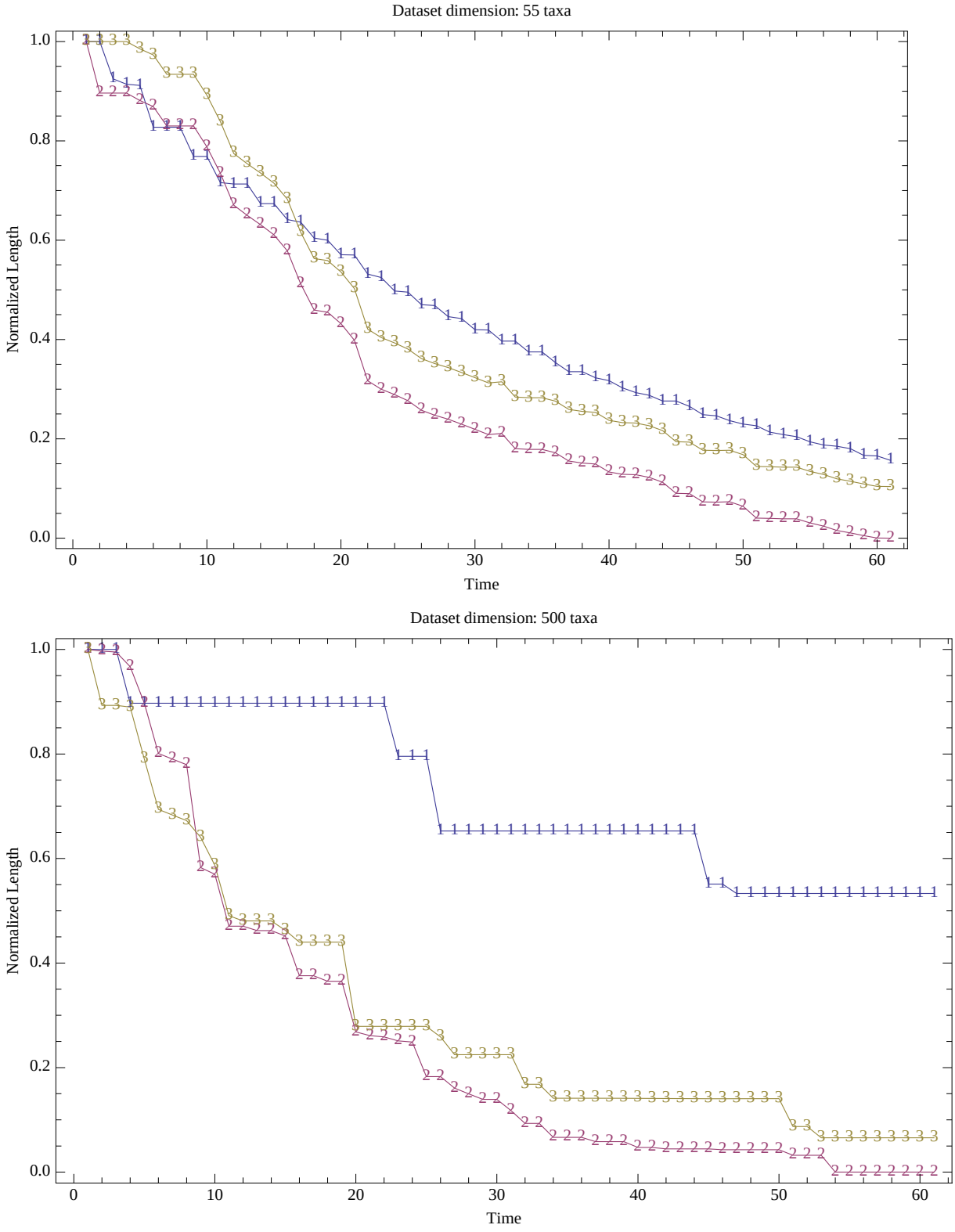

Figure 8
Comparison of score vs. running time for hill-climbing with steepest descent (line labeled "1"), hill-climbing without steepest decent (line labeled "2"), and ACO-ME (line labeled "3") on datasets of 55 (a) and 500 (b) taxa.

Additional material

Additional File 1

An ant colony optimization algorithm for phylogenetic estimation under the minimum evolution principle – supplementary material. The supplementary file includes discussions on the structure of the EPT matrices as well as how we generate and enumerate topologies in ACO-ME.
Click here for file
[http://www.biomedcentral.com/content/supplementary/1471-2148-7-228-S1.tex]

Acknowledgements

Daniele Catanzaro is a Research Fellow at the Belgian National Fund for Scientific Research (FNRS). This work was supported by the "Communauté Française de Belgique" (ARC 11649/20022770) and the "Région Wallone". We thank C. Korostensky, and Mike Steel for helpful discussions, as well as J. L. Deneubourg, L. Keller, and two anonymous reviewers for constructive and helpful comments on a previous version of this manuscript.

References

1. Kidd KK, Sgaramella-Zonta LA: **Phylogenetic analysis: concepts and methods.** *American Journal of Human Genetics* 1971, **23:**235-252.
2. Rzhetsky A, Nei M: **Statistical properties of the ordinary least-squares, generalized least-squares, and minimum evolution methods of phylogenetic inference.** *Journal of Molecular Evolution* 1992, **35:**367-375.
3. Rzhetsky A, Nei M: **Theoretical foundations of the minimum evolution method of phylogenetic inference.** *Molecular Biology and Evolution* 1993, **10:**1073-1095.
4. Bryant D, Waddell P: **Rapid evaluation of least-squares and minimum evolution criteria on phylogenetic trees.** *Molecular Biology and Evolution* 1998, **15(10):**1346-1359.
5. Nemhauser GL, Wolsey LA: *Integer and combinatorial optimization* Wiley-Interscience publication, New York, NY, USA; 1999.
6. Felsenstein J: *Inferring Phylogenies* Sinauer Associates, Sunderland, UK; 2004.
7. Hasegawa M, Kishino H, Yano T: **Evolutionary Trees From DNA Sequences: a Maximum Likelihood approach.** *Journal of Molecular Evolution* 1981, **17:**368-376.
8. Jukes TH, Cantor C: **Evolution of protein molecules.** In *Mammalian Protein Metabolism* Edited by: Munro HN. Academic Press, New York; 1969:21-123.
9. Kimura M: **A simple method for estimating evolutionary rates of base substitutions through comparative studies of nulceotide sequences.** *Journal of Molecular Evolution* 1980, **16:**111-120.
10. Lanave C, Preparata G, Saccone C, Serio G: **A New Method for Calculating Evolutionary Substitution Rates.** *Journal of Molecular Evolution* 1984, **20:**86-93.
11. Rodriguez F, Oliver JL, Marin A, Medina JR: **The general stochastic model of nucleotide substitution.** *Journal of Theoretical Biology* 1990, **142:**485-501.
12. Waddell PJ, Steel MA: **General Time Reversible Distances with Unequal Rates across Sites: Mixing Gamma and Inverse Gaussian Distributions with Invariant Sites.** *Molecular Phylogenetics and Evolution* 1997, **8:**398-414.
13. Cavalli-Sforza LL, Edwards AWF: **Phylogenetic analysis: Models and estimation procedures.** *American Journal of Human Genetics* 1967, **19:**233-257.
14. Beyer WA, Stein M, Smith T, Ulam S: **A molecular sequence metric and evolutionary trees.** *Mathematical Biosciences* 1974, **19:**9-25.
15. Fitch WM, Margoliash E: **Construction of phylogenetic trees.** *Science* 1967, **155:**279-284.
16. Hasegawa M, Kishino H, Yano T: **Dating the human-ape splitting by a molecular clock of mitochondrial DNA.** *Journal of Molecular Evolution* 1985, **22:**160-174.
17. Waterman MS, Smith TF, Singh M, Beyer WA: **Additive evolutionary trees.** *Journal of Theoretical Biology* 1977, **64:**199-213.
18. Day WHE: **Computational complexity of inferring phylogenies from dissimilarity matrices.** *Bullettin of Mathematical Biology* 1987, **49:**461-467.
19. Swofford DL: *PAUP* version 4.0* Sinauer, Sunderland, MA; 1997.
20. Saitou N, Nei M: **The neighbor-joining method: a new method for reconstructing phylogenetic trees.** *Molecular Biology and Evolution* 1987, **4:**406-425.
21. Studier JA, Keppler KJ: **A note on the neighbor-joining algorithm of Saitou and Nei.** *Molecular Biology and Evolution* 1988, **5:**729-731.
22. Kumar S: **A stepwise algorithm for finding minimum evolution evolutionary trees.** *Molecular Biology and Evolution* 1996, **13:**584-593.
23. Desper R, Gascuel O: **Fast and accurate phylogeny reconstruction algorithms based on the minimum evolution principle.** *Journal of computational biology* 2002, **9(5):**687-705.
24. Dorigo M, Caro GD: **The ant colony optimization meta-heuristic.** In *New Ideas in Optimization* McGraw-Hill; 1999:11-32.
25. Zlochin M, Birattari M, Dorigo M: **Model-based search for combinatorial optimization: A critical review.** *Annals of Operations Research* 2004, **131:**373-395.
26. Blum C: **Ant colony optimization: Introduction and recent trends.** *Physics of Life reviews* 2005, **2:**353-373.
27. Dorigo M, Birattari M, Stützle T: **Ant Colony Optimization – Artificial ants as a computational intelligence technique.** *IEEE Computational Intelligence Magazine* 2006, **1:**28-33.
28. Dorigo M, Stützle T: *Ant Colony Optimization* MIT Press, Cambridge, MA; 2004.
29. Stützle T, Hoos HH: *Stochastic Local Search : Foundations and Application* Morgan Kaufman, Elsevier; 2004.
30. Glover F, Kochenberger GA: *Handbook of Metaheuristics* Kluwer Academic Publishers, Boston, MA; 2003.
31. Stützle T, Hoos HH: **MAX-MIN Ant System.** *Future Generation Computer Systems* 2000, **16:**889-914.
32. Maniezzo V: **Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem.** *INFORMS Journal on Computing* 1999, **11:**358-369.
33. Dorigo M, Maniezzo V, Colorni A: **Ant System: optimization by a colony of cooperating agents.** *IEEE Trans Syst, Man, Cybern B* 1996, **26:**29-41.
34. Blum C, Roli A: **Metaheuristics in combinatorial optimization: overview and conceptual comparison.** *ACM Computing Surveys* 2003, **35(3):**268-308.
35. Blum C, Dorigo M: **The Hyper-Cube Framework for Ant Colony Optimization.** *IEEE Transactions on systems, man, and cybernetics – Part B: Cybernetics* 2004, **34(2):**1161-1172.
36. Gascuel O: *Mathematics of evolution and phylogeny* Oxford University Press, New York, NY, USA; 2005.
37. Bentley JL: **Fast algorithms for geometric traveling salesman problems.** *ORSA Journal on Computing* 1992, **4(4):**387-411.
38. Martin O, Otto SW, Felten EW: **Large-step Markov chains for the traveling salesman problem.** *Complex Systems* 1991, **5(3):**299-326.
39. Bianchi L, Birattari M, Chiarandini M, Manfrin M, Mastrolilli M, Paquete L, Rossi-Doria O, Schiavinotto T: **Hybrid metaheuristics for the vehicle routing problem with stochastic demands.** *Journal of Mathematical Modelling and Algorithms* 2006, **5:**91-110.
40. Birattari M, Dorigo M: **How to assess and report the performance of a stochastic algorithm on a benchmark problem: Mean or best result on a number of runs?** *Optimization Letters* 2006. **To Appear**
41. Birattari M, Zlochin M, Dorigo M: **Towards a theory of practice in metaheuristics design: A machine learning perspective.** *RAIRO – Theoretical Informatics and Applications* 2006, **40:**353-369.
42. [http://ueg.ulb.ac.be/metapiga2/].
43. Thompson JD, Gibson TJ, Plewniak F, Jeanmougin F, Higgins DG: **The ClustalX windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools.** *Nucleic Acid Research* 1997, **24:**4876-4882.
44. Catanzaro D, Pesenti R, Milinkowitch M: **A non-linear optimization procedure to estimate distances and instantaneous substitution rate matrices under the GTR model.** *Bioinformatics* 2006, **22(6):**.
45. Gascuel O, Bryant D, Denis F: **Strengths and limitations of the minimum evolution principle.** *Systematic Biology* 2001, **50:**621-627.