2006

A Novel Approach to Phylogenetic Tree Construction using Stochastic Optimization and Clustering

Ling Qin

Yixin Chen

Yi Pan
Georgia State University, pan@cs.gsu.edu

Ling Chen

Recommended Citation

Qin *et al.*: A novel approach to phylogenetic tree construction using stochastic optimization and clustering. *BMC Bioinformatics* 2006, 7(Suppl 4):S24. doi: 10.1186/1471-2105-7-S4-S24

BMC Bioinformatics

BioMed Central

Research

A novel approach to phylogenetic tree construction using stochastic optimization and clustering

Ling Qin*[1], Yixin Chen[2], Yi Pan[3] and Ling Chen[1,4]

Address: [1]Department of Computer Science, Nanjing University of Aeronautics and Astronautics, Nanjing, 210096, China, [2]Department of Computer Science and Engineering, Washington University in St.Louis, USA, [3]Department of Computer Science, Georgia State University,34 Peachtree Street, Suite 1450, Atlanta, GA 30302-4110, USA and [4]Department of Computer Science, Yangzhou University, Yangzhou, 225009 China

Email: Ling Qin* - qllynne@nuaa.edu.cn; Yixin Chen - chen@cse.wustl.edu; Yi Pan - pan@cs.gsu.edu; Ling Chen - lchen@yzcn.net

* Corresponding author

from Symposium of Computations in Bioinformatics and Bioscience (SCBB06) in conjunction with the International Multi-Symposiums on Computer and Computational Sciences 2006 (IMSCCS|06)
Hangzhou, China. June 20–24, 2006

Published: 12 December 2006

BMC Bioinformatics 2006, **7**(Suppl 4):S24 doi:10.1186/1471-2105-7-S4-S24

Abstract

Background: The problem of inferring the evolutionary history and constructing the phylogenetic tree with high performance has become one of the major problems in computational biology.

Results: A new phylogenetic tree construction method from a given set of objects (proteins, species, etc.) is presented. As an extension of ant colony optimization, this method proposes an adaptive phylogenetic clustering algorithm based on a digraph to find a tree structure that defines the ancestral relationships among the given objects.

Conclusion: Our phylogenetic tree construction method is tested to compare its results with that of the genetic algorithm (GA). Experimental results show that our algorithm converges much faster and also achieves higher quality than GA.

Background

An evolutionary tree, or phylogenetic tree, is a model of the evolutionary history for a set of species. With more and more DNA and protein sequences have been obtained [1-3], the problem of inferring the evolutionary history and constructing the phylogenetic tree has become one of the major problems in computational biology. This is because the evolutionary relationship of species provides a great deal of information about their biochemical machinery. For example, RNA's secondary structure is most accurately determined by selecting correlated mutations of a class of related species.

A phylogenetic tree is a tree showing the evolutionary interrelationships among various species or other entities that are believed to have a common ancestor. In a phylogenetic tree, each node with descendants represents the most recent common ancestor of the descendants, and the edge lengths correspond to time estimates. Each node in a phylogenetic tree is called a taxonomic unit, and the leaves usually denote a set of objects (proteins, species, etc.). Internal nodes are generally referred to as Hypothetical Taxonomic Units (HTUs) as they cannot be directly observed [3-6].

To construct a tree from a set of species, one must have a metric to decide if a tree is better than another one. Many criteria have been proposed. But in general, they turn out to be NP-hard to optimize. There is still no consensus in the biology community on how to make a good tree.

One way to counteract this problem is to execute many different phylogenetic clustering methods resulting in various starting tree topologies. A choice from the generated trees gives rise to the best one. Another way to handle this problem is to use a global optimization technique to derive the optimal topology of the tree. In this paper, ant colony algorithm is applied both as a clustering method and as a global optimization technique so that the optimal tree can be found even with a bad initial tree topology.

During the past years, a number of efforts have been contributed to phylogenetic analyses using genomic sequences, which could be either whole genomes (complete gene sequence sets) or complete protein sequence sets [7-9]. There are three main methods for constructing phylogenetic trees: distance-based methods such as neighbour-joining, parsimony-based methods such as maximum parsimony, and character-based methods such as maximum likelihood or Bayesian inference [1,2]. The distance based approaches avoid the high computational complexity of multiple sequence alignment (including genome reorganization) to compute an evolutionary distance and try to construct the phylogenetic trees efficiently. The phylogenetic clustering method in this article belongs to the distance based category.

Ant Colony Optimization (ACO) is a new evolution simulation algorithm proposed by Italian researchers Dorigo et al [10]. Inspired by studies on biological ant colony, they recognize the similarities between the ants' food-hunting activities and TSP, and successfully resolve the TSP problems using the same principle that the ants have used to find the shortest route to food source via communication and cooperation, and it has been applied to lots of combinational optimization problems [10,11].

We note that in the ant colony algorithm, ants can volatilize a kind of chemical odour called pheromone when they encounter each other or in the process of seeking their fellows. Enlightened by this fact, we first apply weights of rejection and acceptance between the objects to form a complete digraph in which the vertexes represent the objects and the initial weight of each edge between vertexes is the weights of acceptance between the objects. The novel clustering process by artificial ants is illustrated in Fig. 1, 2, 3 and 4, during the process, the pheromone on each edge of the digraph will be updated with the artificial ants' adaptive movements, and some adaptive strategies are also presented to speed up the clustering progress. Finally the clusters got by the ants are used to progressively construct the phylogenetic trees.

Results

Constructing a specific digraph for the objects

Ants can volatilize a kind of chemical odour called pheromone when they encounter each other or in the process of seeking their fellows. Based on this kind of odour, ants will naturally attract those who have similar features and repel those that are different. In this paper, artificial ants were set to travel on the graph and deposits pheromone on the edges they passed. As showed in Fig. 1 and Fig. 2, in each step, the artificial ant selects the next vertex according to the acceptance weight in digraph and some heuristic information. The pheromone on each edge of the digraph will be updated with the artificial ants' adaptive movements, and some adaptive strategies are also presented to speed up the clustering progress.

Strong component analysis

The more similar the objects are, the higher the quantity of pheromone may be deposited on the edge between their vertexes. To make full use of the quantity of pheromone on each edge, we omit some connections whose pheromone value is less than a certain threshold to get a new digraph, and the strong connected components of the new digraph forms the finial clusters. This way, the initial objects are separated into a few clusters by the ant sub-colony. Finally these clusters obtained by the ants are used to construct the phylogenetic trees progressively.

Optimizing the phelogenetic trees

Artificial ants in the same sub-colony try to construct an independent phylogenetic tree as a solution of the problem by their cooperation; and different sub-colonies construct different trees so as to maintain diversity of candidates. After optimizing these trees, the performance of these solutions is improved. Meanwhile, the pheromones on the edges of high fitness valued trees are increased to strengthen the ants' clustering process.

The phylogenetic tree construction method showed in this paper is tested to compare its results with that of GA, experimental results show that our algorithm is easier to implement and more efficient. Comparing to GA, it can converge much faster and obtain higher solution quality.

Discussion

Reconstruction of the phylogeny is one of the most important problems in evolutionary study, which is very difficult for large data sets in macromolecular databases. The number of possible phylogenetic trees is exponentially large and the space of topologies cannot be searched exhaustively. Even heuristic searches can be very slow in

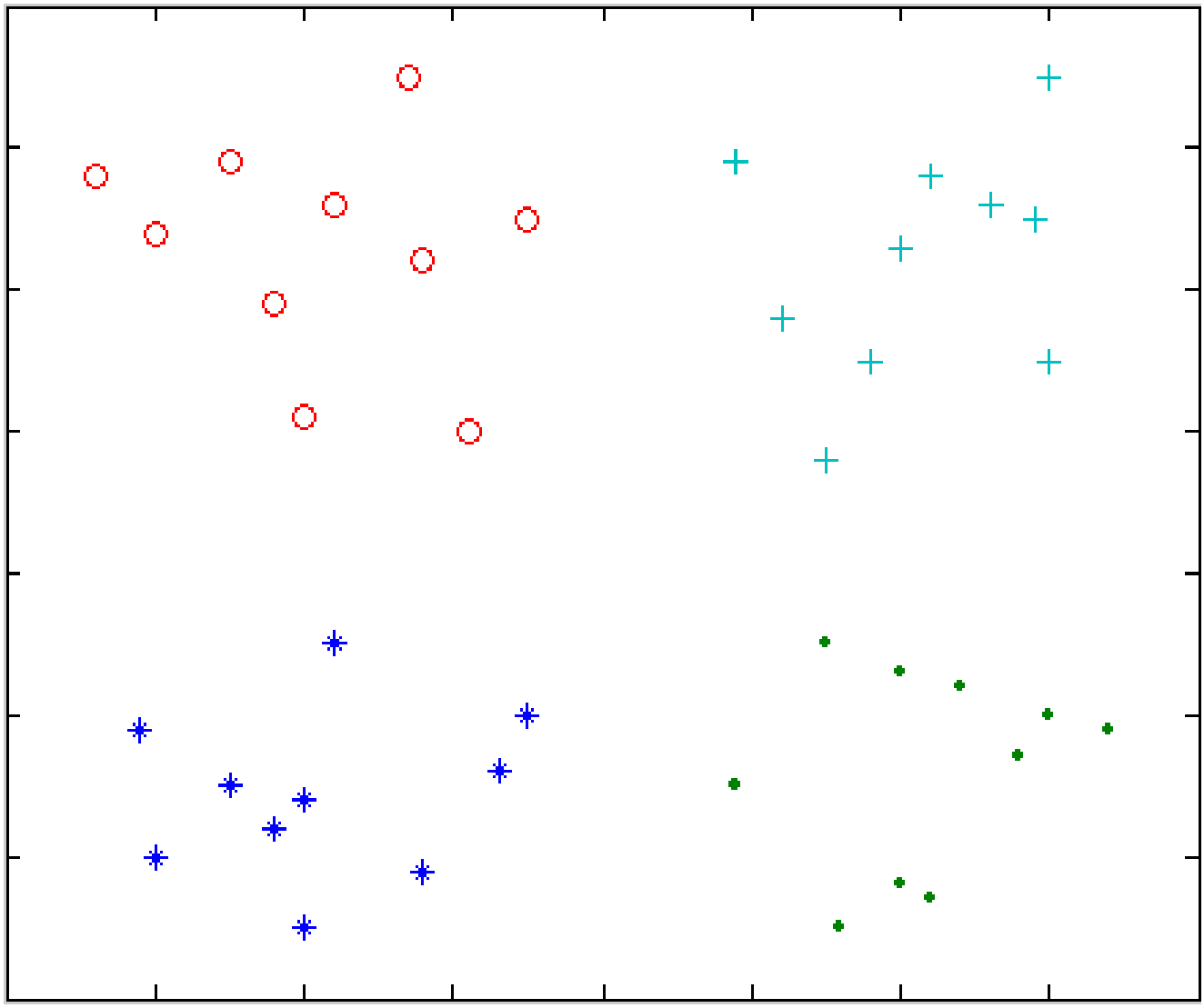

Figure 1
The initial objects. To illustrate the novel clustering process by artificial ants, we test a data set with four data types each of which consists of 10 two-dimensional object (x, y) which belong to four classes as shown in Fig. 1. Here x and y obey normal distribution $N(u, \sigma^2)$. The normal distributions of the four types of data (x, y) are [N(0.2,0.12), N(0.2,0.12)], [N(0.6,0.12), N(0.2,0.12)], [N(0.2,0.12), N(0.6,0.12)], [N(0.6,0.12), N(0.6,0.12)] respectively.

this case, especially when computationally intensive optimality criteria such as maximum likelihood (ML) are used.

An exhaustive search for the ML topology is usually computationally prohibitive for more than 20 taxa (species). At the same time, the clustering approach for a phylogeny inference is advantageous for a number of reasons, including the ability to model a variety of factors affecting nucleotide sequence evolution, robustness to violations of model assumptions, and resistance to long branch attractions. The use of stochastic algorithm provides an opportunity to develop new efficient and fast methods for phylogeny analysis.

Conclusion

The proposed adaptive ant colony algorithm for phylogenetic tree construction method (AAPTC) consists of three components, including initialization, constructing phylogenetic trees through clustering, and phylogenetic tree optimization.

In the stage of initialization, a weighted digraph is built where the vertices represent the data to be clustered and

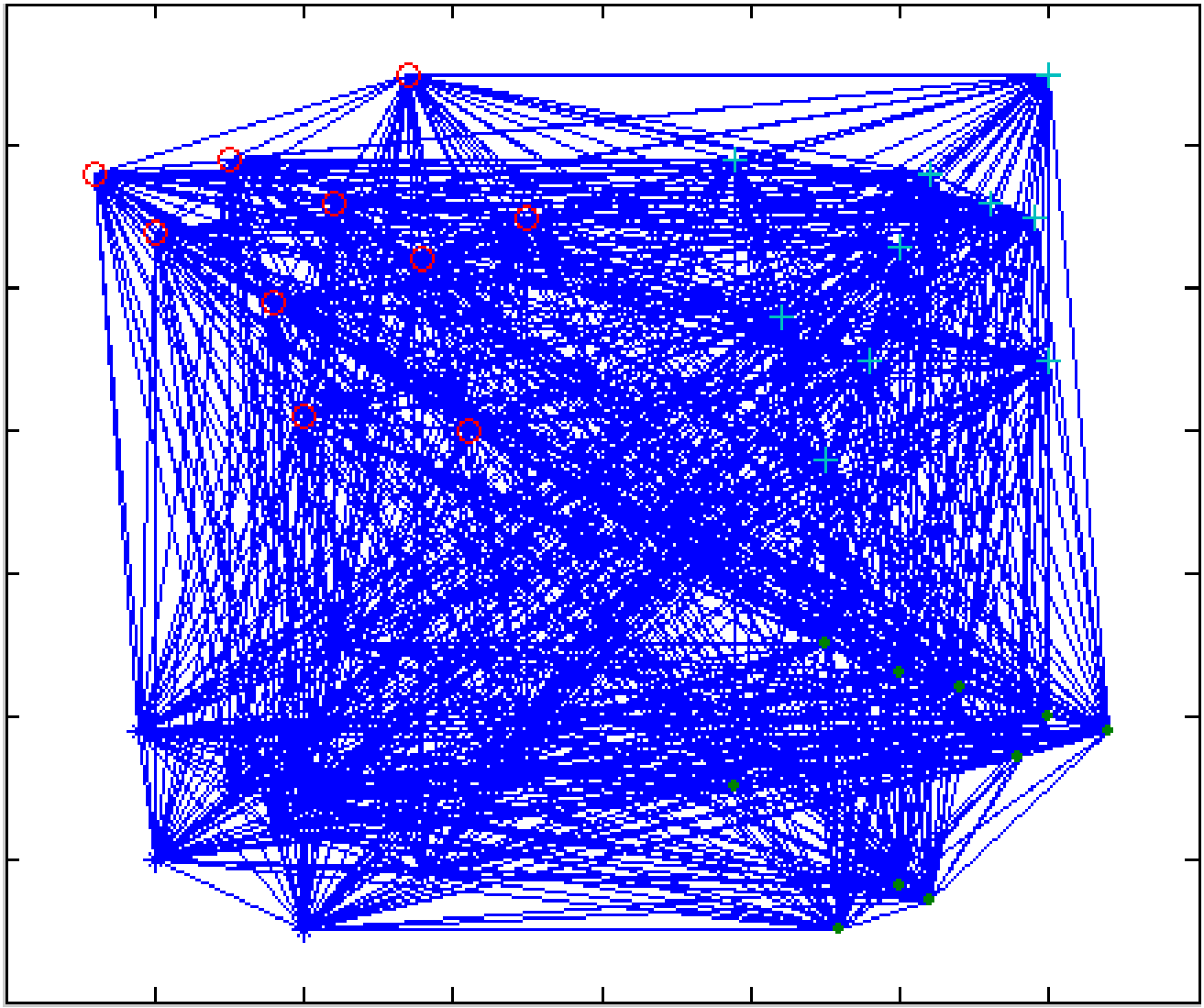

Figure 2
The initial pheromone digraph. Fig. 2 shows the initial pheromone digraph of this object set. The pheromone digraph obtained after 50 iterations of AAPTC is then modified by omitting the edges whose pheromone value is less than $\varepsilon = 1.95$.

the weight is the acceptance rate between the two objects it connected.

In the course of constructing the phylogenetic trees, the ants travel in the digraph and update the pheromone on the paths it passed. At each step, ants choose the next vertex according to a certain probability depending on the pheromone and the heuristic information of the edge. The digraph is first modified by omitting some edges whose pheromone value is less than a threshold, and then the strong connected components of the updated digraph are computed to form the clusters which are used to construct the phylogenetic trees.

After getting a group of phylogenetic trees, the ant colony and its pheromone feedback system act as a global optimization technique to derive the optimal topology of the phylogenetic tree.

The algorithm showed in this paper is tested using randomly generated sequences. Using the same sequences, we also test the GA method. Our experiments were implemented on Dell Precession workstation 380 with IntelP4 Hyper Threading Processor of 3.2 GHz and 800 M Front Bus Speed.

As showed in [3], the simulated data sets used are generated in two different ways. The first set of simulated data

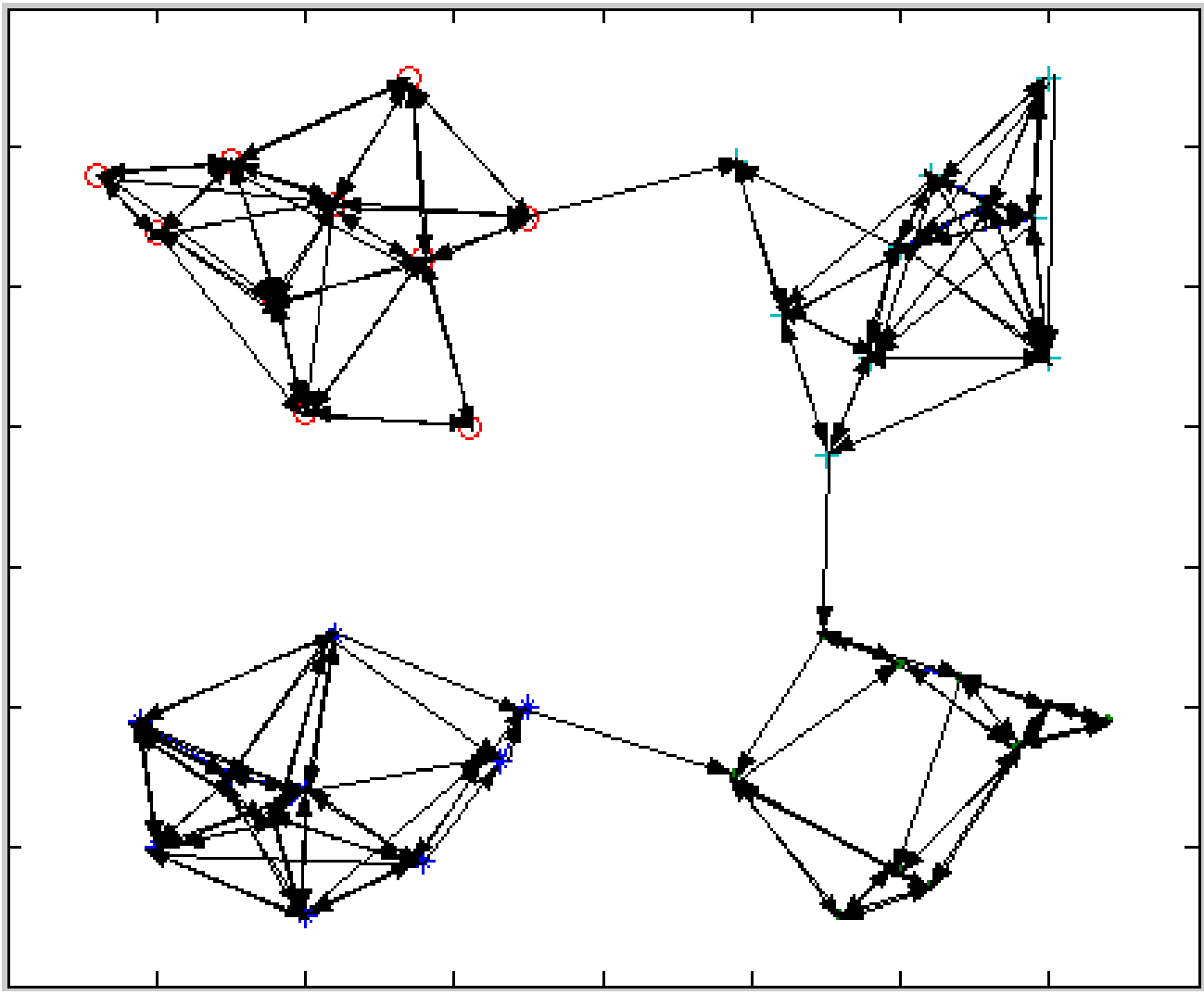

Figure 3
The modified pheromone digraph (after 50iterations). The modified digraph is shown in Fig. 3, objects with lower similarity form some strong connected components

consists of trees where the topology of the tree is fixed and randomly generated branch lengths are assigned to each node-to-node connection. The resulting distance matrix is used as input for the methods to be tested. In this way four sets were generated (S11, S12, S13 and S14) which consists of distance matrices defining ancestral relationships among 24, 96, 1000, and 4000 objects, respectively. The second sets S21, S22, S23 and S24 include stochastically generated distance matrices. For these data sets, the optimal tree is not known.

In fact, AAPTC not only provides a novel clustering method to obtain a group of good initial tree topologies, but also has global optimization on these trees. In this way, the AAPTC produces an ensemble of trees of almost similar quality. Whereas the GA method cannot guarantee the topology quality since its sole initial tree topology was generated by some other clustering methods such as NJ or FITCH. The ensemble of high qualified solutions of AAPTC allows experts to decide which topology is most likely since the quality criterion (the fitness value) used does not guarantee the optimal tree topology.

Tables 1, 2, 3 and 4 show the performance comparison of the two methods. In all the tables, the performance of a method is measured by the fitness value between the original and calculated distance matrices. The number of examined trees is depicted in parentheses. For AAPTC the mean, standard deviation, highest, and lowest fitness value derived from 50 independent trials are given. The

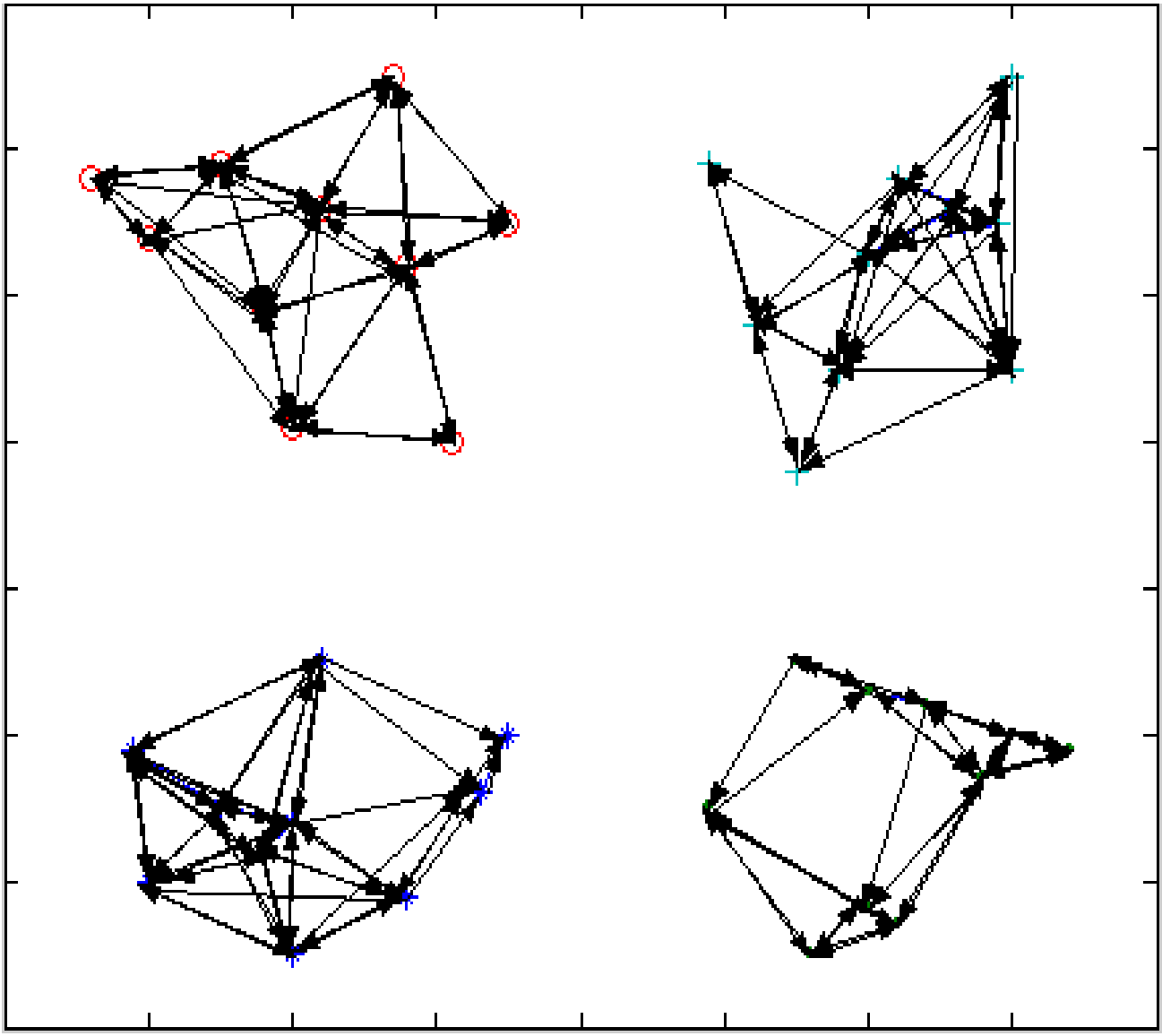

Figure 4
The finial strong connected components. As shown in Fig. 4, four strong connected components of the modified digraph are computed which form the finial clusters.

basic parameters are set as m = n ρ = 0.05 C = 10 q0 = 0.95. We also use the vertebrate dataset [5,12] to evaluate the performance of our algorithm. Vertebrate database contains in total 832 mitochondrial proteins from 64 vertebrates. The results of the neighbour join based phylogeny and taxonomy tree is shown in [8], and ant colony based phylogeny is shown as fig 5.

Methods

Ant Colony Algorithm

Here we briefly introduce AC and its applications using TSP as an example. In the TSP, a given set of n cities has to be visited exactly once and the tour ends in the initial city. We denote the edge between city i and j as (i, j) and its distance as d_{ij} (i, j ∈ [1, n]). Let $\tau_{ij}(t)$ be the intensity of pheromone on (i, j) at time t, and use $\tau_{ij}(t)$ to simulate the pheromone of real ants. Suppose m is the total number of ants, at time t the kth ant selects from its current city i to city j according to the following probability distribution:

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{r \in allowed_k} \tau_{ir}^{\alpha}(t)\eta_{ir}^{\beta}(t)} & j \in allowed_k \\ 0 & otherwise \end{cases} \quad (1)$$

Table 1: The experimental results on the first data set.

Dataset	GA				AAPTC			
	mean	S.D.	high	low	mean	S.D.	high	low
S11	8.78	0.06	9.81 (68345)	7. 40 (42157)	9.83	0.02	9.94 (65487)	9.77 (382914)
S12	5.39	0.35	9.75 (59623)	5.11 (34728)	9.67	0.04	9. 81 (57631)	9.62 (26738)
S13	9.64	0.11	9.83 (72100)	8.92 (32572)	9.88	0.06	9.96 (76895)	9.85 (15201)
S14	5.78	0.42	9.69 (69805)	5.03 (25437)	9.86	0.07	9.91 (82965)	9.70 (43346)

Table 1 gives the results on the first set of simulated data. The optimal tree topology of these data is known in advance because they are generated for a predefined tree topology, and the fitness value of the optimal solution is just 10. Form experimental results given in table 1, both methods could find the optimal tree topology. In the cases of S11, S12, S13 and S14, AAPTC finds the optimal topology for all trials, whereas the GA method falls into local convergence eight times with average fitness value of 5.39 and six times with average fitness value of 5.78 on S12 and S14 respectively. Therefore, ACTP can get more global and higher fitness valued phylogenetic trees than GA.

Where allowedk is a set of the cities can be chosen by the kth ant at city i for the next step, η_{ij} is a heuristic function which is defined as the visibility of the link between cities i and j, for instance it can defined as $1/d_{ij}$.

The relative influence of the trail information $\tau_{ij}(t)$ and the visibility η_{ij} are determined by the parameters α, β. When $\alpha = 1$ and $\beta = 0$, the algorithm becomes a complete heuristic algorithm with positive feedback and when $\alpha = 0$ and $\beta = 1$, it is just a traditional greedy algorithm. For every ant, its path traversing all the cities forms a solution. The intensity of pheromone is updated by Eq. (2):

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij} \quad (2)$$

Where $0<\rho<1$ represents the evaporation of $\tau_{ij}(t)$ between time t and t+1, $\Delta\tau_{ij}$ is the increment of the pheromone on (i, j) in step t, and $\Delta\tau_{ij}^{k}$ is the pheromone laid by the kth ant on it, it takes different formula depending on the model used.

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \quad (3)$$

For example, in the most popularly used model called "ant circle system", it is given as Eq.(4).

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} Q/L_k & \text{if the k th ant passes } (i,j) \text{ in current tour} \\ 0 & \text{otherwise} \end{cases}, \quad (4)$$

where Q is a constant and L_k is the total length travelled by the k^{th} ant.

Constructing phylogenetic trees by clustering and optimization

The overall algorithm for constructing the phylogenetic trees is given below.

Begin

Table 2: The experimental results on the second data set.

Dataset	GA				AAPTC			
	mean	S.D.	high	low	mean	S.D.	high	low
S21	9.12	0.06	9.79 (53135)	9.03 (42257)	9.67	0.01	9.88 (78753)	9.63 (47030)
S22	9.04	0.11	9.22 (69546)	8.98 (32419)	9.81	0.03	9.92 (68964)	9.73 (56229)
S23	8.87	0.17	9.14 (57453)	8.25 (34565)	9.87	0.05	9.89 (76843)	9.83 (36457)
S24	8.90	0.21	8.91 (56739)	8.56 (38897)	9.90	0.09	9.98 (66753)	9.84 (49332)

Table 2 shows the results on the second set of simulated data. We can also see that GA is more likely to fall into local convergence when the objects increase. But AAPTC can get the optimum from given data sets with large number of objects. And AAPTC proposes a more powerful phylogenetic clustering method so it can obtain high qualified solutions no matter how large the number of the objects extends.

Table 3: The average iterations.

Dataset	GA			AAPTC		
	mean	high	low	mean	high	low
S11	0.25	0.41	0.19	0.19	0.17	0.28
S12	0.38	0.43	0.32	0.26	0.39	0.22
S13	0.36	0.48	0.30	0.34	0.45	0.31
S14	0.55	0.62	0.46	0.42	0.47	0.39
S21	0.26	0.35	0.22	0.15	0.13	0.20
S22	0.34	0.44	0.28	0.29	0.37	0.24
S23	0.48	0.58	0.43	0.44	0.38	0.55
S24	0.62	0.68	0.56	0.51	0.46	0.59

Table 3 lists the computation time (hours) these two methods required to get the optimal solution from given data sets. From the experimental results on S11, S12, S21 and S22 we can see that, at the same condition, AAPTC saves much more time than GA.

1 Initialization

1.1 Initialize parameters: minC, m ε, γ, α, β;

1.2 Initialize the pheromone digraph;

1.3 For each ant in each sub-colony do

Chooses an initial object to visit randomly;

End for

2 While (not termination) do // 500 iterations

2.1 For each sub-colony do //m sub-colonies

2.1.1 Set a root node rt as the ancestor of all the objects;and let CO denote the current object set, the initial value of CO equals the given object set of the problem

2.1.2 While (not termination) do

//500 interations

For each ant k in current sub-colony do

//m ants

While ($allowed_k$ not empty) do

Compute probability function p;

Select the next object j to visit;

allowedk =allowedk -{ *j* };

End while

Reset allowedk= CO;

End for

Have local pheromone updating on each edge in the digraph according to the evolutionary distance between objects;

Adaptively update the parameters of α, β;

End while

2.1.3 Transfer the pheromone digraph to another digraph by omitting the edges whose pheromone value is less thanε ; find out the strongest connected components of the updated digraph as clusters. Join the small clusters with the nearest cluster till there are two clusters left, we denote them as clu1 and clu2;

2.1.4 let clu1 and clu2 be the internal node to denote the children of the root node rt

Table 4: The comparison for fitness value between GA and AAPTC.

Dataset	GA				AAPTC			
	mean	S.D.	high	low	mean	S.D.	high	low
GPCR's	8.92	0.12	9.13 (51535)	8.76 (41272)	9.80	0.04	9.91 (87689)	9.65 (38264)

It is clear from Table 4 that in case of the real data set, all methods give approximately the same results. Comparison between AAPTC and GA reveals that the AAPTC finds slightly better phylogenetic trees, according to the fitness values.

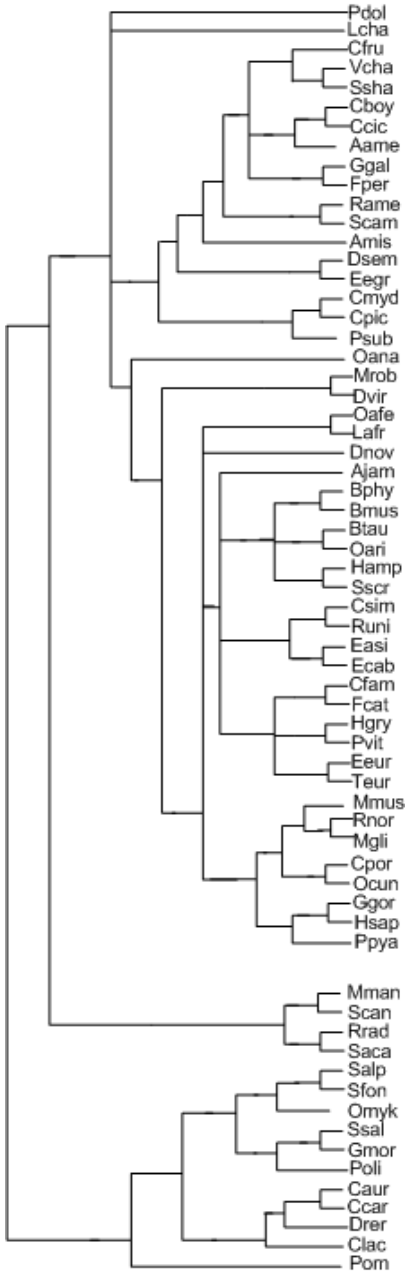

Figure 5
The ant colony based phylogeny on the 64vertebrates. Using the vertebrate dataset [8, 16], the ant colony based phylogeny is showed in fig. 5, we can see that the constructed phylogeny is largely consistent with the taxonomy tree showed in[8]. For example, all perissodactyls, birds, reptiles, bony fish etc. are grouped together, as they showed in neighbour join based phylogeny and the taxonomy tree of [8]. This demonstrates that our ant colony based phylogenies at least can equally performance with NJ based method. And from tables of experimental results we can easily find that ant colony based phylogeny has an advantage of easily computed and optimized results.

2.1.5 If the number of objects in clu1 is lower than 1 let rt=clu1, CO = { clu1} break;

2.1.6 If the number of objects in clu2 is lower than 1 let rt=clu1, CO = { clu2} break;

2.2 obtain each phylogenetic tree constructed by each sub-colony

End for

2.3 Calculate the fitness value of each sub-colony

2.4 Have crossover and mutation operation to improve the quality of the trees

2.5 Have global pheromone updating operation according to the fitness value of the constructed phylogenetic trees

End while

3 Output the phylogenetic trees constructed by the colony

End

In the while loop between line 2 and line 3, based on τ_{ave}, the parameter of threshold ε could be defined as $\varepsilon = \gamma^* \tau_{ave}$, where γ is a constant. The population size of the ant colony m is normally equal to n/2, here n is the number of the given objects. The value of parameters α and β are subject to be adjusted adaptively in process of the algorithm. In line of 2.4, the crossover and mutation operation are executed by branch moving, swapping techniques introduced in [13].

Initialization of the Pheromone Diagraph

The initialization stage of the algorithm constructs a weighted digraph with the vertexes representing the given objects and the weighted edges between vertexes representing the acceptance weight between the two objects it connected. The acceptance weight between two objects can be calculated from the evolutionary distance between the objects.

Definition 1: The set of objects

A set of n objects is defined as S=(CO, RT) where *CO* = $\{object_1, object_2, ...object_n\}$ represents the object set, and rt is the ancestor of all the objects in CO.

A similarity or evolutionary distance is often obtained by pair-wise comparisons of DNA or protein sequences. The measurements of the evolutionary distance can be classified into the following three categories: the first type usu-

ally let the number of homologous genes divided by the total number of genes, or its variants be the evolutionary distance [14]; in the second kind, regularities identified in genetic sequences by compression algorithms are used to represent biological significance for evolutionary history, but these data compression based methods often involve of aggregated errors [15]; in the third category, the evolutionary distance is measured by string composition based on the singular value decomposition (SVD) of a string frequency matrix [13], or on the composition vector on short strings of a fixed length or the information discrepancy on short strings of a fixed length [1,2]. In this paper, we use the cosine distance introduced in [4,5] as the evolutionary distance between the objects.

Definition 2: The evolutionary distance between objects

The evolutionary distance $d(object_i, object_j)$ between $object_i$ and $object_j$ is defined as:

$$d(object_i, object_j) = \frac{1 - C(object_i, object_j)}{2}, i, j = 1,2,\ldots n \quad (5)$$

Here, C ($object_i$, $object_j$) is the cosine of the angle between vector *i* and vector *j* defined in [3,4].

Definition 3: The mean distance and the shortest distance

We use $d_{mean}(object_i)$ to denote the mean distance from $object_i$ to all the other objects, namely

$$d_{mean}(object_i) = \frac{1}{n-1}\sum_{j=1}^{n} d(object_i, object_j) \quad (6)$$

We also denote the shortest distance from $object_i$ to all the other objects as $d_{\min}(object_i)$.

$$d_{\min}(object_i) = \min_{1\le j\le n, j\ne i} d(object_i, object_j) \quad (7)$$

Definition 4: The acceptance weights

For two objects $object_i$ and $object_j$ the acceptance weights for $object_i$ to $object_j$ is defined as Eq.(8):

$$accept_i(object_j) = \frac{d_{\min}(object_i) + d_{mean}(object_i)}{d(object_i, object_j)} \quad (8)$$

Similarly, the acceptance weights for $object_j$ to $object_i$ is as Eq.(9):

$$accept_j(object_i) = \frac{d_{\min}(object_j) + d_{mean}(object_j)}{d(object_i, object_j)} \quad (9)$$

From the definition we can see that the more similar two objects are, the greater acceptance weight to each other will be. We also can see that acceptance weight between two objects is not symmetric, namely, normally

$$accept_j(object_i) \ne accept_i(object_j) \quad (10)$$

According to the definitions above, we could form a weighed digraph where each vertex represents an object. Denote the weight of the directed edge from $object_i$ to $object_j$ as $\tau_{ij}(0)$. This value will be updated according to the pheromone deposited by the ants passing it. Its initial value $\tau_{ij}(0)$ is set as the acceptance weight:

$$\tau_{ij}(0) = accept_i(object_j) \quad (11)$$

In traditional ant colony algorithm, pheromone on all edges is usually initialized as zero. This is not helpful for ants to choose path at the early stages. However, in AAPTC, the proposed initial pheromone value set on the digraph is much important for ants' latter movements, that is to say it can make great influence on the initial topology of the phylogenetic trees. Based on this initial value, in the latter stages the ants will update this pheromone digraph for the construction and optimization of the phylogenetic trees.

Heuristic function

The heuristic function ηij in Eq.(1) is a problem dependent function that measures the "quality" of the edge (*i, j*) which connects the vertexes i and j representing the two objects. Here the "quality" means the preference of the edge to be selected by the ants. Obviously, the less distance between the two connected objects, the more preferred the edge should have. Therefore, η_{ij} should be associated with the distance between objects. So it is given by the following formula.

$$\eta_{ij} = 1/d\ (object_i, object_j) \quad (12)$$

Different from pheromone τ_{ij}, η_{ij} is static and unidirectional heuristic information determined by the distance information.

Pheromone Updating

In the algorithm, based on the following formula, pheromone on edge (*i,j*) is updated on the paths the ants just passed after each iteration.

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t)\sum_{k=1}^{m}\Delta\tau_{ij}^{k} \quad (13)$$

Here constant $\rho \in (0,1)$ is the coefficient of evaporation. At an individual iteration the pheromone on each path will be evaporated by a rate of ρ.

In the local updating period, $\Delta\tau_{ij}^{k}$ is the increment of τ_{ij} by ant k, which is defined by Eq.(14)

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/d(object_i, object_j) & if\ ant\ k\ passes\ path\ i-j \\ 0 & otherwise \end{cases} \quad (14)$$

Here Q is a constant. From the formulas above, it is easy to see that the more ants pass through an edge, the more pheromone deposited on it, and the more probability for the two vertexes connected by the edge to be included in the same strong connected component of the weighted digraph constructed in the third stage of the algorithm. In the global updating period, $\Delta\tau_{ij}^{k}$ is the increment of τ_{ij} by sub-colony *k*, which is defined as follows :

$$\Delta\tau_{ij}^{k} = \begin{cases} Q \cdot fitness(sub-colony_k) & if\ (i,j) \in Tree_k \\ 0 & otherwise \end{cases} \quad (15)$$

Here, $Tree_k$ denotes the phylogenetic tree constructed by sub-$colony_k$, fitness(sub-$colony_k$) is the fitness value of $Tree_k$. According to [9], once the topology and the branch lengths of Treek are determined, a new distance matrix can be deduced. By comparing this distance matrix D^k with the original distance matrix D (calculated from the given objects), a quality measurement showed in Eq.(16) can be assigned to the tree as its fitness value:

$$fitness(sub-colony_k) = C \cdot (1 - \sqrt{\frac{(\sum_{i=1} D_i^k - D_i)^2}{(\frac{n(n-1)}{2} - 1) \cdot \max_i(D_i^k, D_i)}}) \quad (16)$$

The summation extends over all n(n-1)/2 distances between the n objects. If the distances are concentrated within a narrow scope, a high fitness value will be assigned to the tree, and if the reconstructed distance equal the original distances, the fitness will reach the highest value C. By global pheromone updating, the pheromone deposited on the edges of high fitness valued trees will be much higher than others, thus the objects connected by these edges can hardly be separated by the ants during the clustering process.

Updating Parameters

The second stage of the algorithm consists the step of updating the value of α, β which are the parameters of the Eq.(1) which is the probability distribution for the ant's selecting the next vertex. In Eq.(1), parameters α, β determine the relative influence of the trail strength τ_{ij} and the heuristic information η_{ij}. At the initial stage of the algorithm, the pheromone value on each edge is relatively small. To speedup the convergence, the ants should select the path mainly according to the heuristic information η_{ij}. Therefore, the value of α should be relatively large in this stage. After some iteration, the pheromone values on the edges are increased, their influence become more and more important. Therefore the value of β should be relatively large. Since the adjustment of the values of α and β should be based on the strength of pheromone on the edges In Eq.(17) we define τ_{ave} as the average amount of pheromone on the pheromone digraph and in Eq.(18) define δ as the pheromone distributing weight to measure the distribution of pheromone on the graph.

$$\tau_{ave} = \frac{\sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} \tau_{ij}}{n(n-1)} \quad (17)$$

$$\delta = \frac{\sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} \left|\tau_{ave} - \tau_{ij}\right|}{n(n-1)} \quad (18)$$

Using the pheromone distributing weight δ, the algorithm updates the value of α, β as follows:

$$\alpha = \log^{1+\delta} \quad (19)$$

$$\beta = \frac{1}{\alpha} \quad (20)$$

Once the pheromone digraph is updated, α and β will be adaptively modified by the pheromone distributing weight and make influence on the effect of pheromone and heuristic function. By adjusting the value of α, β adaptively, the algorithm can accelerate the convergence and also can avoid local convergence and precocity. Therefore, this adaptive procedure is much important for AAPTC. Furthermore, since the amount of pheromone is an important measure for tree construction, the pheromone distributing weight δ is also a critical factor to terminate the iterations of the algorithm.

Authors' contributions

LQ conceived, designed and performed the study under the supervision of LC and YP; LQ and YC (Yixin Chen) collected and analyzed the data; LQ wrote the computer code; LQ and YC designed algorithms; LQ and LC wrote the manuscript; All authors have read and approved the final manuscript.

Acknowledgements

This paper is supported in part by a US Department of Energy ECPI grant, the Chinese National Natural Science Foundation under grant No. 60673060, Chinese National Foundation for Science and Technology Development under contract 2003BA614A-14, and the Natural Science Foundation of Jiangsu Province under contract BK2005047.

This article has been published as part of *BMC Bioinformatics* Volume 7, Supplement 4, 2006: Symposium of Computations in Bioinformatics and Bioscience (SCBB06). The full contents of the supplement are available online at http://www.biomedcentral.com/1471-2105/7?issue=S4.

References

1. Hodge T, Cope MJTV: **A Myosin Family Tree.** *Journal of Cell Science* 2000, **113:**3353-3354.
2. Saitou N, Nei M: **The neighbor-joining method: A new method for reconstructing phylogenetic trees.** *Mol Biol Evol* 1987, **4:**406-425.
3. Reijmers TH, Wehrens R, Daeyaert FD, Lewi PJ, *et al.*: **Using genetic algorithms for the construction of phylogenetic trees: application to G-protein coupled receptor sequences.** *Biosystems* 1999, **49:**31-43.
4. Hao B, Qi J: **Prokaryote phylogeny without sequence alignment: from avoidance signature to composition distance.** *Proceedings of the 2003 IEEE Bioinformatics Conference* 2003:375-385.
5. Xiaomeng W, Xiufeng W, Gang W, Dong X, Guohui L: **Phylogenetic Analysis Using Complete Signature Information of Whole Genomes and Clustered Neighbor-Joining Method.** *International Journal on Bioinformatics Research and Applications* 2006, **2(3):**219-248.
6. Wu X, Wan XF, Xu D, Lin GH: **Whole genome phylogeny based on clustered signature string composition.** *Posters in 2005 IEEE Computational Systems Bioinformatics Conference (CSB2005)* 2005:53-54.
7. Chen X, Wong SK, Li M: **A compression algorithm for DNA sequences and its applications in genome comparison.** In *proceedings of the sixth annual international computing and combinatorics conference (RECOMB) ACM press*; 2000:107-117.
8. Grumbach S, Tahi F: **A new challenge for compression algorithms: genetic sequences.** *Journal of Information Processing Management* 1994, **30:**866-875.
9. Hao B: **Fractals from genomes-exact solutions of a biology inspired problem.** *Physica* 2000, **A282:**225-246.
10. Dorigo M, Maniezzo V, Colomi A: **Ant system: Optimization by a colony of cooperating agents.** *IEEE Transactions on Systems, Man, and Cybernetics-Part B* 1996, **26:**29-41.
11. Ling C, Jie S, Ling Q: **An Adaptive Ant Colony Algorithm Based on Equilibrium of Distribution.** *Journal of Software* 2003, **14:**1148-1151.
12. Stuart G, Moffett K, Bozarth RF: **A comprehensive vertebrate phylogeny using vector representation of protein sequences from whole genomes.** *Molecular Biology and Evolution* 2002, **19:**554-562.
13. Kuntz P, Snyder D: **New results on ant-based heuristic for highlighting the organization of large graphs.** In *Proceedings of the 1999 Congress or Evolutionary Computation IEEE Press, Piscataway, NJ*; 1999:1451-1458.
14. Herniou E, *et al.*: **Use of Whole genome sequence data to infer baculovirus phylogency.** *Journal of virology* 2001, **75:**8117-8126.
15. Grumbach S, Tahi F: **A new challenge for compression algorithms: genetic sequences.** *Journal of Information Processing Management* 1994, **30:**875-866.