An Evolutionary Approach to the Inference of Phylogenetic Networks

Juan Diego Trujillo and Carlos Cotta

Dept. Lenguajes y Ciencias de la Computación, ETSI Informática,
University of Málaga, Campus de Teatinos, 29071 - Málaga - SPAIN
ccottap@lcc.uma.es

Abstract. Phylogenetic networks are models of the evolution of a set of organisms that generalize phylogenetic trees. By allowing the existence of reticulation events (such as recombination, hybridization, or horizontal gene transfer), the model is no longer a tree but a directed acyclic graph (DAG). We consider the problem of finding a phylogenetic network to model a set of sequences of molecular data, using evolutionary algorithms (EAs). To this end, the algorithm has to be adequately designed to handle different constraints regarding the structure of the DAG, and the location of reticulation events. The choice of fitness function is also studied, and several possibilities for this purpose are presented and compared. The experimental evaluation indicates that the EA can satisfactorily recover the underlying evolution model behind the data. A computationally light fitness function seems to provide the best performance.

1 Introduction

Phylogenies are used to represent the evolutionary history of a collection of organisms (represented by phenotypical information, or –as assumed throughout this paper– by molecular sequence data). Typically, this evolutionary history is represented as a tree, i.e., a hierarchy showing the degree of closeness among the organisms under study. As it turns out, inferring the *best* hierarchy is a formidably difficult task under several formulations [1, 2]. This hardness barrier can be circumvented using heuristic approaches; indeed, evolutionary algorithms (EAs) have been used in this domain with encouraging results, e.g., [3–7] among others. At any rate, there is an additional important fact we should not lose sight of: trees oversimplify our view of evolution, as it has been long recognized by biologists. There are many events in natural evolution in which the genetic material is not transferred in a hierarchical way, e.g., hybrid speciation, horizontal gene transfer, etc. These phenomena, usually called reticulations, give rise to edges that connect nodes from different branches of a tree, creating a directed acyclic graph structure that is usually called a phylogenetic network [8].

No single methodology for network reconstruction is widely accepted to date [9]. For example, the detection and identification of reticulation events has been approached by Hallett *et al.* [10] (focusing on horizontal gene transfer), and by Posada *et al.* [11] (focusing on recombination events). Nakhleh *et al.* [12] have

proposed a method that combines pre-existing consensus trees into a network with a single reticulation event. Finally, Gusfield *et al.* [13, 14] have devised several algorithms for binary input sequences, under different assumptions on where reticulation events take place, and how they work.

The methods mentioned above are in general based in deterministic approaches for finding provably good solutions, and hence the well-known limitations arising from the $P \neq NP$ conjecture apply. To the best of our knowledge, the inference problem has not been approached with metaheuristic techniques so far. However, this latter approach seems natural in this domain, given the success history of these techniques (EAs in particular) on the inference of phylogenetic trees. In this work, we propose an evolutionary approach to the phylogenetic-network inference problem, and show that it can be a useful tool in this domain.

2 Phylogenetic Networks

As mentioned in previous section, there exist some evolutionary events that do not fit in the tree-like view of evolution, e.g., hybrid speciation, recombination, and horizontal gene transfer. In general, these events require the use of rooted directed acyclic graphs (DAGs) for representing them. In the following, we will describe the notation used henceforth, as well as some crucial notions such as time coexistence, and topological distance metrics on phylogenetic networks.

2.1 Notation

Let $G(V, E)$ be a DAG. We will use the notation $E(G)$ and $V(G)$ to denote respectively the set of edges and vertices of a DAG G. A directed path P of length k from u to v in a graph G $(u, v \in V(G))$, is a sequence $P = \langle u_0, u_1, \cdots, u_k \rangle$ of nodes where $u = u_0$, $v = u_k$, and $(u_i, u_{i+1}) \in E(G)$ for $0 \leqslant i < k$. Let $\alpha(P) = u_0$, and $\omega(P) = u_k$ be the endpoints of path P. A node v is reachable from u in G if there exists a directed path from u to v; in that case, u is an ancestor of v. Unlike trees, there may be more that one directed path between two nodes in a DAG. These paths are also termed *positive time directed paths* for reasons that will be clear at a later point.

We can now define the in-degree $\delta^{\downarrow}$ of a node as the number of edges arriving to that node, and the out-degree $\delta^{\uparrow}$ as the number of edges that depart from that node. There are some degree constraints in DAGs representing phylogenetic networks. To be precise, a node $v \in V(E)$ is a *tree node* if (a) $\delta^{\downarrow}(v) = 0$ and $\delta^{\uparrow}(v) = 2$ [root (unique)], (b) $\delta^{\downarrow}(v) = 1$ and $\delta^{\uparrow}(v) = 0$ [leaf], or (c) $\delta^{\downarrow}(v) = 1$ and $\delta^{\uparrow}(v) = 2$ [internal tree node]. If a node v is not a tree node, then it must have $\delta^{\downarrow}(v) = 2$ and $\delta^{\uparrow}(v) = 1$. Such nodes are termed *network nodes*. An edge $e = (u, v) \in E(G)$ is a *tree edge* if and only if v is a tree node, and it would be a *network edge* otherwise. Notice that tree nodes describe mutations, and network nodes describe reticulation events. Fig. 1(a) shows an example of phylogenetic network. If given any edge in the network at least one of its endpoints is a tree node (and provided some constraints on the structure of reticulation events are fulfilled, see Sect. 2.2), the network is termed *reconstructible* [9].

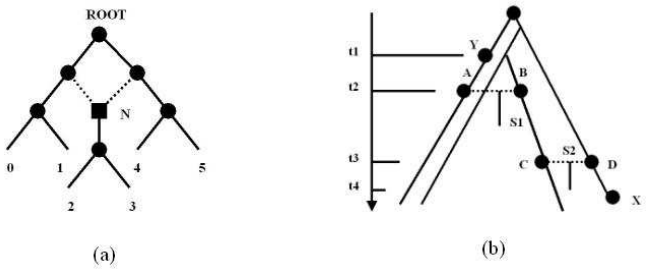


Fig. 1: (a) A phylogenetic network N on six observed species (and seven ancestral species). Tree nodes and network nodes are depicted with circles and squares respectively. Likewise, solid lines denote tree edges, and dashed lines denote the network edges. (b) X and Y cannot coexist in time.

2.2 Time Coexistence

A crucial consideration that must be taken into account in phylogenetic networks is the fact that each reticulation event defines a *simultaneity plane*: in order to have two species recombining their genomes, or having some genetic information transferred from a species to another, they must coexist in time. This way, the set of nodes $V(G)$ of a phylogenetic network G are implicitly ordered in time according to the particular reticulation events existing in G. To be precise, we can assign a specific time-stamp $t(v)$ to each node v in the network. Actually, the absolute values of these time-stamps are not relevant; what matters is their relative ordering. Thus, if there exists a time directed path between node u and node v, we would logically have $t(u) < t(v)$, i.e., u is an ancestor of v. However, if $e = (u, v)$ is a network edge, then $t(u) = t(v)$ because the reticulation events are instantaneous in the evolution time scale.

Consider Fig. 1(b). Let $t(Y) = t_1$ and $t(X) = t_4$, and let reticulation events s_1 and s_2 happen at time t_2 and t_3 respectively. Now notice that there does not exist a positive time directed path from Y to X. However, even though Y is not an ancestor of X, it is impossible to have a reticulation event between these two nodes: Y is an ancestor of A, that coexists with B; B is in turn an ancestor of C that coexist with D, an ancestor of X. Hence, $t_1 < t_2 < t_3 < t_4$. More formally, we say that two nodes $u, v \in V(G)$ cannot coexist in time if:

(a) u is an ancestor of v (or vice versa), or
(b) there is a sequence of positive time directed paths $P = \{P_1, P_2, \cdots, P_k\}$ such that $\alpha(P_1) = u$ (resp. v), $\omega(P_k) = v$ (resp. u), and for $1 \leqslant i < k$, there exists a network node whose parents are $\omega(P_i)$, and $\alpha(P_{i+1})$.

Time coexistence thus imposes constraints on where reticulation events can take place, and therefore on which DAGs actually represent a feasible phylogenetic network. These constraints will have to be observed when evolving networks within the inference algorithms.

2.3 Topological Distance Metrics on Phylogenetic Networks

Metrics for measuring the topological distance between networks are essential to interpret the results of an inference algorithm. They can be used to determine to which extent the features of a target network have been successfully recovered. For this purpose, we can resort to generalizations of well-known distance metrics for trees, such as the Robinson-Foulds (RF) distance [15].

The RF distance on trees uses the notion of *bipartition*: given an edge e in a tree T, we can partition the leaf set $\mathcal{L}$ of T into two disjoint sets $A(e)$ and $C(e)$, respectively comprising the leaves in $\mathcal{L}$ that are reachable from the root (resp. unreachable) through edge e. The notion of bipartition in trees is readily generalizable to *tripartitions* in networks. In this latter case, an edge e induces a tripartition $\langle A(e), B(e), C(e)\rangle$, where $A(e)$ comprising the leaves that are reachable from the root only through edge e, $B(e)$ comprises the leaves that are reachable from the root by a path that goes through e, and at least by another path that does not pass through e, and $C(e)$ is defined as before.

We denote by $\phi(e) = \langle A(e), B(e), C(e)\rangle$ the tripartition induced by the e. Two tripartitions $\phi(e_1)$ and $\phi(e_2)$ are equivalent ($\phi(e_1) \equiv \phi(e_2)$) if, and only if, $A(e_1) = A(e_2)$, $B(e_1) = B(e_2)$, and $C(e_1) = C(e_2)$. Now, two edges e_1, e_2 are compatible ($e_1 \equiv e_2$) if, and only if, $\phi(e_1) \equiv \phi(e_2)$. Let $\delta : \mathbb{B} \to \{0, 1\}$ be defined as $\delta(\texttt{TRUE}) = 1$ and $\delta(\texttt{FALSE}) = 0$. Let $\Gamma(G_1, G_2)$ be defined as

$$\Gamma(G_1, G_2) = \frac{1}{|E(G_1)|} \sum_{e_1 \in E(G_1)} \left(1 - \sum_{e_2 \in E(G_2)} \delta(e_1 \equiv e_2) \right) \tag{1}$$

It is then possible to define the *false negative rate* $FN(G, \tilde{G}) = \Gamma(\tilde{G}, G)$, and the *false positive rate* $FP(G, \tilde{G}) = \Gamma(G, \tilde{G})$ between an inferred network G and a target network $\tilde{G}$. Finally, the RF distance for networks can be estimated as $D_{RF}(G, \tilde{G}) = (FN(G, \tilde{G}) + FP(G, \tilde{G}))/2$. Notice that the RF distance is 0.0 for two identical networks, and 1.0 for two networks without any compatible edge.

3 EAs for Inferring Phylogenetic Networks

In order to tackle the inference of phylogenetic networks with EAs, we consider a direct approach in which the search is directly conducted on the space of all possible phylogenetic networks with given leaf set. Thus, each individual in the population of the EA represents a feasible phylogenetic network, internally encoded as an adjacency matrix. This means that (i) an initialization process producing feasible networks must be used, and (ii) the reproductive operators used must respect feasibility, i.e., they must always produce feasible offspring. The details of these operators will be described in Sect. 3.1.

Another central element in this EA is the fitness function. The RF metric defined in the previous section can be used for the off-line assessment of the results, but it cannot obviously be used during the evolution. On the contrary, the fitness function must evaluate a phylogenetic network on the basis of the

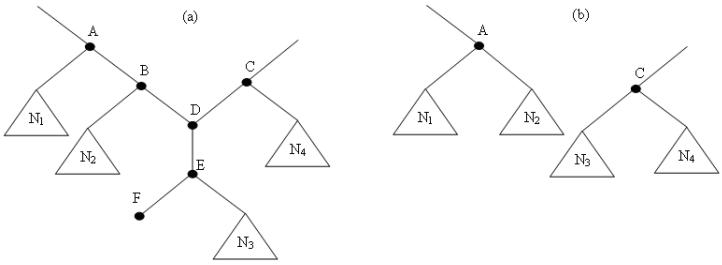


Fig. 2: After deleting node F, the subnetwork in (a) takes the shape shown in (b).

particular dataset of molecular sequences to be modelled. Several choices have been explored for this purpose. These are described in Sect. 3.2.

3.1 Evolutionary Operators

The first issue to be tackled in the EA is the generation of feasible networks for insertion in the initial population. To do so, each time a new network is required a random tree is firstly generated (this is done by firstly constructing a random permutation of the leaves; then, an initial tree is built with the first two leaves in the permutation, and the remaining leaves are subsequently inserted at random points of the partial tree until it is finally completed). Once the tree has been obtained, network nodes are inserted by randomly selecting pairs of tree nodes that can coexist in time.

After having generated a population of feasible networks, adequate reproductive operators must be used. Let us firstly consider the recombination operator. As usual, this operator takes information pieces from two individuals, and combines them to create a new feasible solution. In this case, these information pieces take the shape of subnetworks, and hence we can express the process in terms of pruning and grafting subnetworks. Let G_1 and G_2 be the networks to recombine, and let trees be represented in LISP notation. The process is as follows:

1. Select a random subnetwork N (rooted at a tree node) from G_1.
2. **for each** leaf $u \in N$ **do**
   (a) Find subnetwork U in G_2 such that $U = (h, (u), U')$ or $U = (h, U', (u))$.
   (b) Replace U by U' in G_2.
3. Select a random subnetwork V from G_2.
4. Replace V by $V' = (h', N, V)$, where h' is a new internal tree node.

This operator can be regarded as a generalization of the Prune-Delete-Graft (PDG) operator for trees [5, 6]. Thus, we have termed it NetPDG. Notice that some network node might be broken during the recombination process. This could happen either in step 1 (if there were a reticulation event between a node in N and another node not in N), or in step 2b (if a grandson of a network node were removed). Fig. 2 shows an example of this latter situation.

As to the mutation operator, it is based on rearranging the topology of a portion of the network. More precisely, to mutate network G, this operator (NetSCRAMBLE) selects a random subnetwork N from G, and generates another random network, spanning the same set of leaves, and having the same number of internal network nodes (network nodes with one parent in N and other parent outside N are broken). Notice that not only the number of network nodes in the child might be lower than that of the parental solution(s) (if some such nodes are broken during recombination –as described before– or during subnetwork scrambling in mutation); it can be also higher than it if no network node is broken, and new ones are transferred during recombination. In this work, we have opted for keeping a fixed, predefined number of network nodes in solutions. Hence, whenever a new solution has a higher or lower number of network nodes, it is repaired (breaking randomly selected network nodes, or adding new ones).

3.2 The Fitness Function

As it is the case for phylogenetic trees, the accuracy with which a phylogenetic network model the evolutionary history of a certain dataset can be computed via sequence-based methods (i.e., maximum parsimony [16], or maximum likelihood [17]) and distance-based methods [18]. Among these approaches, maximum likelihood is an appealing way of assessing the quality of a proposed phylogenetic model: they consider all possible evolutionary pathways compatible with the molecular data available, and are known to be asymptotically accurate [19]. We have thus opted for a maximum likelihood approach here.

The general setting is the following: we have a collection D of n sequences, representing some molecular data from n different species. Here, these sequences are taken from the alphabet $\Sigma = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, i.e., they represent DNA sequences. We assume a certain evolution model at the molecular level, indicating the likelihood that a certain character (nucleotides in this case) mutates into another one, say $\Pi = \{\pi_{ij}\}_{i,j \in \Sigma}$. Each site in the sequence is assumed to evolve independently. Likewise, we assume a certain mechanics for reticulation events, i.e., network nodes indicate recombinations, and these produce organisms whose genetic sequence is, e.g., the result of a uniform crossover of the parental sequences. When this general evolutionary framework is superimposed on a particular network N, we can calculate $P(D|N)$, that is, the likelihood that N gave rise to D. A potential drawback of this method is its computational cost. Related to this issue, we have considered several alternative formulations of the fitness function to estimate $P(D|N)$, as described below.

The first method is based on the formula used for likelihood calculation in trees. Let L^r_{k,s_k} be the likelihood of a network rooted at node k, given that that node has nucleotide s_k in site r. If node k is the parent of nodes i and j in the network, and both are tree nodes, then,

$$L^r_{k,s_k} = \left(\sum_{s_i \in \Sigma} \pi_{s_k,s_i} L^r_{i,s_i} \right) \left(\sum_{s_j \in \Sigma} \pi_{s_k,s_j} L^r_{j,s_j} \right) \ . \tag{2}$$

In case node i were a network node, the first term in Eq. (2) would have to be changed accordingly. To be precise, the state of node i would depend on the state of node k, and also on the state of the other parent, say node z. This bivariate dependency precludes the fast recursive evaluation of Eq. (2). To circumvent this issue, we can take into account the fact that, due to the semantics of recombination, the state of node k propagates with $1/2$ probability to node i. Thus, we approximate this first term as $1/2 \cdot L^r_{v,s_k}$, where v is the unique child of node i in the network. The same reasoning would apply to node j were it a network node. While this is a mere approximation of the exact likelihood of these network nodes, it allows a very fast recursive evaluation of the overall likelihood of the complete network. This evaluation is completed by noting that (i) the likelihood $L^r_{w,s}$ of a leaf is 1.0 if the r^{th} site of the w^{th} sequence is s, and 0.0 otherwise, and (ii) the complete likelihood of the network for site r is $L^r = \sum_{s_0 \in \Sigma} \pi_{s_0} L^r_{0,s_0}$, where π_s is the marginal probability of nucleotide s. Finally, since sites evolve independently, the likelihood of the network for the whole data is $L = \prod_{i=i}^{m} L^i$, m being the length of sequences. We term this evaluation method ABE (after approximate bayesian estimation).

Monte Carlo (MC) methods constitute an alternative to ABE providing an asymptotically exact numerical estimation of the network likelihood. This estimation is obtained by constructing N samples of the states of internal nodes in the network, and computing

$$L^r = \frac{1}{N} \sum_{i=1}^{N} \prod_{w} \pi_{s^i_{w'}, s_w} \tag{3}$$

where the inner product ranges over all leaves w of the network, w' is the parent node of a certain w, $s^i_{w'}$ is the i^{th} sampled state in the r^{th} site for node w', and s_w is the actual state in the r^{th} site of the w^{th} sequence. In order to have a correct MC integration, the probability of each sample must be proportional to its real likelihood. This can be easily accomplished by assuming a random state at the network root (following the marginal probabilities π_s), and simulating the evolution of this site along the network, using the stochastic model Π considered.

The MC method provides an asymptotically more accurate estimation of the exact likelihood, but it has a much higher computational cost than ABE. In order to alleviate this cost partially, a combined method (CMB) has been considered. The basic idea is identifying all the subtrees in the network (that is, maximal subgraphs without network nodes), using the MC method just to sample the states for the remaining nodes. Subsequently,

$$L^r = \frac{1}{N} \sum_{i=1}^{N} \prod_{v} L^r_{v,s^i_v} \prod_{w} \pi_{s^i_{w'}, s_w} \tag{4}$$

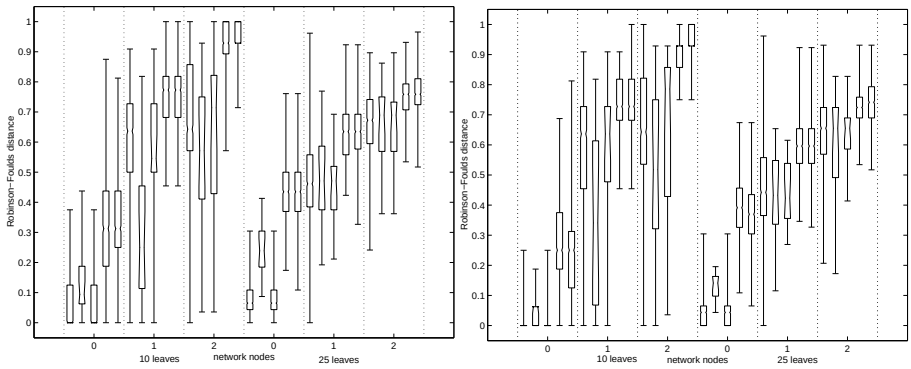
where the first product ranges over all internal nodes v being the root of a maximal subtree, s^i_v is the corresponding value in the i^{th} sample, the second product ranges over all leaves that are not part of a maximal subtree, and w', $s^i_{w'}$, and s_w are interpreted as before. Thus, the exact likelihood value is computed for maximal subtrees, and the cost of the MC component is reduced.

4 Experimental Results

The data used in the experiments have been synthetically generated from known evolution models, in order to allow an objective measurement of the extent to which the inference algorithms were capable of recovering the underlying model. The process consists of generating a random network with the desired number of leaves (n) and network nodes (k), and then constructing nucleotide sequences by simulating the evolution of r sites throughout the network. The stochastic evolution of sequences is done assuming the Kimura 2-parameter model [20] with transition rate $\alpha = 0.05$ and transversion rate $\beta = .025$. We have considered networks with $n \in \{10, 25\}$ leaves, $k \in \{0, 1, 2\}$ network nodes, and $r \in \{100, 250\}$ sites per sequence. Both the procedure for generating the dataset, and the parameters used are similar in related woks [12, 16].

A steady-state EA with standard parameters ($popsize = 100$, $p_X = .9$, $p_m = 1/\ell$, ℓ being the number of nodes, $maxevals = 1000n$, binary tournament selection) has been used in the experiments. No fine tuning of these parameters has been attempted. To allow a wider exploration of the capabilities of the EA, a different problem instance has been used in each run. This way we are evaluating the algorithm on many samples of the whole problem class, rather than just on a couple of instances. Results have been obtained for the three fitness functions described in Sect. 3.2. Twenty runs have been done for each parameter set for functions MC and CMB. Function ABE turns out to be around 50 times faster than MC (using 500 samples per evaluation), so we have conducted 1000 runs for it per parameter set. The best networks found are evaluated in terms of the RF distance with respect to the "real" network. For the network model considered, the most related approach in the literature is [12]. Unfortunately, the SPNET program used there is not available. For this reason, we have devised an combination of greedy-exhaustive heuristic for comparison purposes: we firstly construct a tree using an agglomerative technique such as complete-link (CL) or single-link (SL), and then exhaustively check all locations where to place the network nodes (one at a time), keeping the best network.

Complete results are shown in Fig. 3. As expected, the ABE function performs very well in the $k = 0$ case (i.e., tree models) since it captures the exact likelihood of each tentative solution. For $n = 10$, $k > 0$, the MC function provides better results than ABE (the MC estimation may be better than the approximation used in ABE); however, for $n = 25$, $k > 0$, the differences are negligible (not statistically significant, using a Wilcoxon ranksum test [21] since data is not normally distributed), and the best results of ABE are even better than those of MC for $k = 1$. In general, CMB performs similarly to ABE, and all three evolutionary approaches are much better (statistically significant) than CL and SL. Notice also that ABE can recover the original network in at least one run for all parameter settings except $n = 25$, $k = 2$. We have also conducted similar experiments with networks generated with an additional constraint: the parents of network nodes must be located at the same depth. The results are essentially the same as depicted in Fig. 3 for unconstrained networks.



(a) (b)

Fig. 3: Results for different instance sizes (from left to right in each group of five boxes: ABE, MC, CMB, CL, and SL). The boxes comprise the second and third quartile, and the whiskers indicate the range of the data. (a) Sequences of 100 nucleotides. (b) Sequences of 250 nucleotides.

5 Conclusions

We have analyzed several evolutionary approaches for the inference of phylogenetic networks from molecular data. The results indicate that EAs can be a useful tool in this domain, since it has been shown that they can provide network models very close to the real evolutionary model hidden in the data (sometimes recovering it in full), outperforming some ad-hoc heuristics as well. We have compared three different fitness functions for guiding the evolution. The ABE function seems to provide the best tradeoff between the quality of the results, and the computational cost implied.

Future work will be directed to analyze the generalizability of this evolutionary approach to other reticulation events. For example, recombination can be analyzed on diploid organisms (the offspring would inherit a full DNA sequence from each of the parents). The approach can be also readily adapted to tackle alternative assessment models such as maximum parsimony.

Acknowledgements. Thanks are due to Jorge Muruzábal for useful comments on bayesian methods and Monte Carlo integration. The second author is partially supported by MCyT project TIN2005-08818-C04-01.

References

1. Foulds, L., Graham, R.: The Steiner problem in phylogeny is NP-complete. Advances in Applied Mathematics **3** (1982) 439–49

2. Day, W., Johnson, D., Sankoff, D.: The computational complexity of inferring rooted phylogenies by parsimony. Mathematical Biosciences **81** (1986) 33–42
3. Matsuda, H.: Protein phylogenetic inference using maximum likelihood with a genetic algorithm. In Hunter, L., Klein, T.E., eds.: 1st Pacific Symposium on Biocomputing '96, London, World Scientific (1996) 512–523
4. Lewis, P.: A genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data. Molecular Biology and Evolution **15** (1998) 277–283
5. Moilanen, A.: Searching for the most parsimonious trees with simulated evolution. Cladistics **15** (1999) 39–50
6. Cotta, C., Moscato, P.: Inferring phylogenetic trees using evolutionary algorithms. In Merelo, J., et al., eds.: Parallel Problem Solving From Nature VII. Volume 2439 of Lecture Notes in Computer Science. Springer-Verlag, Berlin (2002) 720–729
7. Shen, J., Heckendorn, R.: Discrete branch length representations for genetic algorithms in phylogenetic search. In Raidl, G., et al., eds.: Applications of Evolutionary Computing 2004. Volume 3005 of Lecture Notes in Computer Science., Coimbra, Portugal, Springer (2004) 94–103
8. Huson, D., Bryant, D.: Application of phylogenetic networks in evolutionary studies. Molecular Biology and Evolution **23** (2006) 254–267
9. Moret, B., Nakhleh, L., Warnow, T., Linder, C., Tholse, A., Padolina, A., Sun, J., Timme, R.: Phylogenetic networks: Modeling, reconstructibility, and accuracy. IEEE Transactions on Computational Biology and Bioinformatics **1** (2004) 13–23
10. Hallett, M., Lagergren, J., Tofigh, A.: Simultaneous identification of duplications and lateral transfers. In: 8th Annual International Conference on Computational Molecular Biology. (2004) 347–356
11. Posada, D., Crandall, K.: The effect of recombination on the accuracy of phylogeny estimation. Journal of Molecular Evolution **54** (2002) 396–402
12. Nakhleh, L., Warnow, T., Linder, C.: Reconstructing reticulate evolution in species – theory and practice. In: 8th Annual International Conference on Computational Molecular Biology. (2004) 337–346
13. Gusfield, D., Eddhu, S., Langley, C.: Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. Journal of Bioinformatics and Computational Biology **2** (2004) 173–213
14. Song, Y., Wu, Y., Gusfield, D.: Efficient computation of close lower and upper bounds on the minimum number of recombinations in biological sequence evolution. Bioinformatics **21** (2005) i413–i422
15. Robinson, D., Foulds, L.: Comparison of phylogenetic trees. Mathematical Biosciences **53** (1981) 131–147
16. Nakhleh, L., Jin, G., Zhao, F., Mellor-Crummey, J.: Reconstructing phylogenetic networks using maximum parsimony. In: Computational Systems Bioinformatics Conference 2005, IEEE Press (2005) 93–102
17. Strimmer, K., Moulton, V.: Likelihood analysis of phylogenetic networks using directed graphical models. Molecular Biology and Evolution **17** (2000) 875–881
18. Makarenkov, V., Kevorkov, D., Legendre, P.: Modeling phylogenetic relationships using reticulated networks. Zoologica Scripta **33** (2005) 89–96
19. Huelsenbeck, J., Crandall, K.: Phylogeny estimation and hypothesis testing using maximum likelihood. Annual Review of Ecology and Systematics **28** (1997) 437–466
20. Kimura, M.: Estimation of evolutionary distances between homologous nucleotide sequences. Proceedings of the Natural Academy of Sciences **78** (1981) 454–458
21. Lehmann, E.: Nonparametric Statistical Methods Based on Ranks. McGraw-Hill, New York NY (1975)