Generalized Neighbor-Joining:

More Reliable Phylogenetic Tree Reconstruction*

William R. Pearson†, Gabriel Robins, and Tongtong Zhang

Department of Computer Science, University of Virginia

† Department of Biochemistry, University of Virginia

Charlottesville, VA 22908

Key Words: phylogenetic reconstruction, neighbor-joining, least-squares, minimum-evolution, solution space sampling

*The corresponding author is Professor William Pearson, wrp@virginia.edu, 804-924-2818, FAX: 804-924-5069

Abstract

We have developed a phylogenetic tree reconstruction method that detects and reports multiple, topologically distant, low cost solutions. Our method is a generalization of the Neighbor-Joining (NJ) method of Nei and Saitou, and affords a more thorough sampling of the solution space by keeping track of multiple partial solutions during its execution. The scope of the solution space sampling is controlled by a pair of user-specified parameters - the total number of alternate solutions and the number of alternate solutions that are randomly selected - effecting a smooth tradeoff between run time and solution quality and diversity. This method can discover topologically distinct low cost solutions. In tests on biological and synthetic datasets using either the least-squares distance or minimum-evolution criterion, the method consistently performed as well as, or better than, either the Neighbor-Joining heuristic or the PHYLIP implementation of the Fitch-Margoliash distance measure. In addition, the method identified alternative tree topologies with costs within 1 or 2% of the best, but with topological distances 9 or more partitions from the best solution (16 taxa); with 32 taxa, topologies were obtained 17 (least-squares) - 22 (minimum-evolution) partitions from the best topology when 200 partial solutions were retained. Thus, the method can find lower cost tree topologies and near-best tree topologies that are significantly different from the best topology.

1 Introduction

Reconstruction of ancestral relationships from contemporary data is widely used to provide both evolutionary and functional insights into biological systems. The explosive increase in available DNA sequence data has increased interest in phylogenetic analysis of multi-gene and domain-swapped protein families. Three general classes of phylogenetic reconstruction methods are commonly used for analysis of sequence datasets: parsimony methods (Swofford *et al.*, 1996), distance based methods (Fitch and Margoliash, 1967), and maximum likelihood methods (Felsenstein, 1982; Felsenstein, 1988). Parsimony- and distance-based methods are most often used, largely because they are faster computationally and allow a larger number of potential phylogenetic trees to be evaluated.

Reconstruction of an evolutionary history for a set of contemporary taxa based on their pairwise distance is computationally intractable (i.e., NP-complete) for various optimality criteria (Foulds and Graham, 1982; Day, 1987), including the least-squares criterion[1] and the minimum-evolution criterion[2]. Various heuristics have been proposed to search for solutions of desired quality (Felsenstein, 1988; Bandelt and Dress, 1992; Swofford *et al.*, 1996), and

[1] According to the least-squares criterion, the best phylogeny for the input distance matrix is the one that minimizes $\sum_{1 \leq i \leq j \leq n} (t_{ij} - d_{ij})^2$ where d_{ij} is the distance between taxa i and j in the input distance matrix, and t_{ij} is the sum of all the branch lengths along the unique path connecting taxa i and j in the postulated phylogeny.

[2] By the minimum-evolution criterion, the best phylogeny for the input distance matrix is one that minimizes the sum of all edge lengths, where edge lengths are assigned to minimize the least-squares deviation.

the majority of these methods are greedy, which always employ moves that Are “locally best” and may not necessarily lead to global optima (Swofford *et al.*, 1996). Among the greedy approaches, the Neighbor-Joining method (Saitou and Nei, 1987; Studier and Keppler, 1988) is widely used by molecular biologists due to its efficiency and simplicity.

Greedy methods are efficient because they explore only a small portion of the solution space[3]. However, greedy methods can fail to find the best overall solution if they become “trapped” in local optima. In addition, because only a small fraction of the solution space is examined, a greedy heuristic typically will not report (or detect) alternative solutions with distinct topologies that may fit the data nearly as well, or even equally well. Neglecting such alternative solutions can produce misleading inferences regarding the evolutionary history. For instance, Wilson et al. concluded that all humans originated from Africa, because their tree-building method failed to discover alternative, near-optimal, trees that were consistent with a different geographical history (Wilson *et al.*, 1989; Maddison, 1991).

To improve the reliability of phylogenetic tree reconstructions, we propose a scheme which samples the solution space more extensively by repeatedly using the Neighbor-Joining algorithm (Saitou and Nei, 1987). Instead of tracking only a single, locally-best tree as Neighbor-Joining does, our scheme maintains multiple partial solutions as it progresses. The method explores all possible trees derivable from the set of current partial solutions in a single neighbor-joining step, and then selects a subset of these partial solutions to pass on to the next iteration. This approach is competitive with Neighbor-Joining in recovering distinct

[3] A solution space is the set of all possible phylogenies spanning the given taxa. Taxa correspond to leaves in a tree that spans them.

low-cost topologies, while still being computationally efficient.

2 Methods

2.1 The Neighbor-Joining Method

The Neighbor-Joining method (NJ) was initially proposed by Saitou and Nei (1987), and later modified by Studier and Kepler (1988). Neighbor-Joining seeks to build a tree which minimizes the sum of all edge lengths, i.e., it adopts the minimum-evolution (ME) criterion. A number of studies have corroborated NJ's performance in reconstructing correct evolutionary trees (Saitou and Imanishi, 1989; Kuhner and Felsenstein, 1994; Huelsenbeck, 1995). For small numbers of taxa, NJ solutions are likely to be identical to the optimal ME tree (Saitou and Imanishi, 1989).

Neighbor-Joining begins with a star tree, then iteratively finds the closest neighboring pair (i.e. the pair that induces a tree of minimum sum of edge lengths) among all possible pairs of nodes (both internal and external). The closest pair is then clustered into a new internal node, and the distances of this node to the rest of the nodes in the tree are computed and used in later iterations. The algorithm terminates when $n - 2$ internal nodes have been inserted into the tree (i.e., when the star tree is fully resolved into a binary tree). The Neighbor-Joining heuristic is illustrated in Figure 1B.

Fig. 1 goes near here.

Although the Neighbor-Joining method runs quickly, it returns only the single best solution found by its greedy search strategy. This solution can be further improved with post-processing by rearranging branches and swapping subtrees (Rzhetsky and Nei, 1992; Swofford, 1996), but such improved solutions tend to remain topologically similar to the original starting-point solutions. To increase our confidence in the solution reliability, it is natural to ask if there are other solutions, with different topologies, that are equally well-supported by the distance matrix data.

Solution spaces can exhibit many alternate local optima (Penny *et al.*, 1995). For instance, among all 15 possible trees of 5 taxa, 2 of them (T_1 in Figures 1C and T_2 in Figure 1B) fit the input matrix (Figure 1A) best. However, these two trees have very different topologies; they share no common internal edges. Indeed, according to the partition distance metric (see Section 2.4), T_1 and T_2 are the most dissimilar possible.

2.2 The Generalized Neighbor-Joining Method

Our Generalized Neighbor-Joining (GNJ) method samples the solution space extendedly by keeping track of multiple partial solutions as it progresses (the number of partial solutions K is an input parameter). Unlike the Neighbor-Joining method, which follows only a single path towards a solution, GNJ performs a more thorough search of the solution space by tracking and exploring many potentially-good paths. That is, GNJ retains promising partial solutions, which may not be locally-optimal, but which have the potential for substantially greater cost savings in subsequent steps. An execution example of GNJ on the matrix of Figure 1A is shown in Figure 2.

Fig. 2 goes near here.

The Generalized Neighbor-Joining (GNJ) algorithm can select in several ways the K partial solutions that are passed on to the next iteration. A simple strategy would save the K best (i.e., least cost) partial solutions; alternatively, partial solutions can be chosen at random. Selecting the best solutions tends to improve solution quality, while selecting alternates randomly tends to increase solution diversity. We implement a hybrid scheme that balances these two extremes: the top $Q \leq K$ least-cost solutions are selected, along with an additional $D = K - Q$ "topologically diverse" solutions[4]. If there are more than Q least-cost pairs at a given iteration, GNJ will select Q of them *arbitrarily*, which makes the GNJ method non-deterministic.

To achieve topological diversity, at each iteration, after selecting the best Q partial solutions, the remaining partial solutions are partitioned into G groups according to their topological distances from a best partial solution (partial solutions within the same group are equidistant from the best partial solution). We then obtain an additional D "topologically diverse" partial solutions for the next iteration by selecting the top $\lfloor \frac{D}{G} \rfloor$ solutions from each group[5]. Thus, at the last step towards solving a 16-taxa problem, alternate solutions can be as many as 13 partitions away from the best current solution. In this case, if $D = 50$, at least $\frac{50}{13} = 3$ best solutions at topological distances 1 and 2 are saved, and at least the 4 best solutions at topological distances 3 through 13 are saved. For 32 taxa and $D = 100$, at least 3 solutions will be saved at each topological distance. Because the maximum topological distance increases linearly with the number of taxa, the above strategy ensures that the number of

[4]The parameter names Q and D are mnemonic for *quality* and *diversity*, respectively.

[5]If G does not divide D, we select one additional solution from each of the $D - (\lfloor \frac{D}{G} \rfloor \cdot G)$ groups corresponding to the topological distances farthest from the best partial solution.

topologically distinct good solutions can remain relatively constant by increasing D linearly, rather than exponentially, with the number of taxa.

A similar idea is employed in the stepwise minimum-evolution tree building method (Kumar, 1996). At each iteration, for a given partial tree, this method first identifies the leading node (i.e., the node most likely to be joined to another node), and forms the set of next-step Neighbor-Joining trees by clustering each node with the leading node. This strategy restricts the solution space somewhat, but it requires exponential time to run, which makes it practical only for small datasets. Moreover, it does not explicitly consider alternate solutions at different topological distances (see below), so it is less likely to identify topologically distinct alternatives.

Different combinations of Q and D ($K = Q + D$) enable a smooth trade off between quality vs. diversity. As Q increases with respect to D (for a fixed K), lower-cost solutions are favored over ones with diverse topologies, while for smaller values of Q, the solution space exploration becomes more broad, and topologically different local optima are more likely to emerge. We note that if $K = Q = 1$ (and thus $D = K - Q = 0$), the single solution returned by our Generalized Neighbor-Joining approach is *identical* to the solution produced by the original Neighbor-Joining method (Saitou and Nei, 1987; Studier and Keppler, 1988). Here only the best-cost partial solution is passed to each subsequent iteration, which is exactly what is done by Neighbor-Joining. Thus, GNJ directly generalizes the Neighbor-Joining method.

Additional strategies for expanding the search of phylogenetic tree-space might be considered. The GNJ approach can be abstractly divided into two phases: (1) a tree generation component which produces multiple partial solutions, and (2) a partial solution evaluation function which favors certain preferred partial solutions over others. The overall run time per iteration of the combined method is asymptotically no greater than the slowest of these two components.

The algorithm described in Section 2.2 utilizes the Neighbor-Joining method as the partial tree generation mechanism in phase (1), while using the minimum-evolution criterion (implicit in the Neighbor-Joining method) in filtering candidate partial solutions in phase (2). However, *any* combination of existing algorithms or heuristics for tree generation and tree evaluation can be incorporated into this general template.

For example, we can evaluate partial trees at each step using the least-squares deviation optimality criterion. An alternative scheme for tree generation might allow arbitrary partitions at intermediate steps (i.e., “join” *any* number of taxa rather than exactly two). In this case, a number of existing efficient partitioning heuristics (Alpert and Kahng, 1995) can be readily applied to generate more promising and diverse partial solutions. Likewise, the method for selecting topologically diverse partial solutions might select more solutions from more distant topologies, rather than uniformly sampling the topological distances as is done in this implementation.

The GNJ program is written in the ‘C’ programming language and is available from `ftp://ftp.virginia.edu/pub/fasta/GNJ`. To make the GNJ results more usable in practice, we output the trees obtained by GNJ in a computer-readable format that can be readily processed by other programs (e.g., the `consense` program in the PHYLIP package). Moreover, we summarize the leaf partitions found among the GNJ solutions below a threshold cost, and rank them by decreasing frequencies.

2.3 Datasets

We tested the GNJ heuristic in the UNIX environment. Two types of distance matrices were used to evaluate the algorithm:

(1) Distance matrices were constructed for nucleotide sequences generated by randomly mutating an "ancestral" sequence along a model evolutionary tree using the treeDNA program (Felsenstein, personal communication) with the Kimura two-parameter model for mutation rates (Kimura, 1980). Three types of topologies were used for the model trees: topologies of minimum diameter (which we refer to as *Type 'A'*), topologies of maximum diameter (*type 'B'*), and a mixture of both (*type 'C'*). Here, the *diameter* of a topology is defined as the maximum number of edges connecting any two leaf nodes within the topology. Therefore, topologies of type 'A' are most "branchy" (i.e., they resemble a complete binary tree), while topologies of type 'B' are more "stringy". *type 'A'* trees were the most challenging, and are used for most of the figures.

Divergence rates ranging from 0.005 (internal branches) to 0.50 (leaf or external branches) were used to produce the synthetic data. Two different type 'A' and type 'B' datasets were examined. type 'A1' and 'B1' datasets used divergence rates ~ 0.02 (32 taxa) – ~ 0.05 (8-taxa) for internal edges and ~ 0.4 for external edges (thus, the ratio of external to internal branch rates varied from 10 for 8 taxa to 35 for 32 taxa). Type 'A2' and 'B2' trees used rates of ~ 0.005 for the central (internal) edges and ~ 0.50 for the external (leaf) edges (external/internal ratios of 100).

(2) Several biological datasets were examined, including immunological data from 9 frog species (Saitou and Nei, 1987), data from 13 viral *env* V3 fragments and *gag* P17 (Leitner

et al., 1996), and 47 aligned TCP-1 chaperonin 60 family members (Jon Sund Blandfort, personal communication). For DNA sequences, the distance matrices were computed with the `dnadist` progam in the PHYLIP package (Felsenstein, 1993), using the Kimura 2-parameter model (Kimura, 1980). For protein sequences, the distance matrices were computed with the `protdist` program in the PHYLIP package (Felsenstein, 1993), using the Dayhoff PAM matrix model (Dayhoff, 1978). We obtain 30 biological data sets of 8, 16, or 32 taxa by randomly sampling the original data sets. Results on the different biological datasets were similar; only results on the chaperonin distances (referred to as dataset 'R1') are reported.

2.4 Algorithms Compared

We evaluated the datasets using three algorithms: (1) *NJ:* The Neighbor-Joining method (Saitou and Nei, 1987; Studier and Keppler, 1988), as implemented in the PHYLIP package (Felsenstein, 1993); (2) *FM:* The Fitch-Margoliash method for fitting topologies to distance matrices with respect to the least-squares criterion (Fitch and Margoliash, 1967), as implemented in the PHYLIP package (Felsenstein, 1993); and (3) *GNJ:* the Generalized Neighbor-Joining method, described in this paper.

In addition, we examined every possible tree topology for synthetic and biological data over 8 taxa. This exhaustive method is guaranteed to return a global optimum (i.e. the lowest-cost topology). Because of the sheer size of the solution space, the optimal method is feasible only for datasets containing fewer than ten taxa.

The solutions from the different algorithms were evaluated using either the least-squares or the minimum-evolution criterion. Least-squares tree cost is computed by assigning non-negative edge lengths in a way that minimizes the least-squares deviation. Minimum-evolution tree cost is computed as the sum of such edge-lengths in a tree.

To improve further the solution quality, we also applied a post-processing optimization step which rearranges subtrees as follows. Given a topology, we compute the cost of all the trees resulting from swapping/exchanging subtrees around each of the internal edges of the topology. Then, the lowest-cost tree is chosen as the new current tree, and its neighborhood is investigated in turn. We iterate this process until no further improvement can be obtained.

Topological distances in this paper are based on the partition metric (Robinson and Foulds, 1981; Penny and Hendy, 1985; Steel and Hendy, 1993), which measures the number of edges common to a given pair of binary trees. Each internal edge naturally partitions the set of leaf nodes into two subsets. Two trees spanning the same set of leaves have a common edge if removing this edge induces the same two subsets of leaf nodes. Thus, the partition distance between any two trees is defined as the number of edges in one tree for which there is no corresponding equivalent edge in the other tree. Since each binary tree of n leaves has $n-3$ internal edges, distances under the partition metric can be represented as integers between 0 and $n-3$.

3 Results

Like Neighbor-Joining, Generalized Neighbor-Joining (GNJ) seeks to identify phylogenetic tree topologies and branch lengths that best fit distance data. GNJ improves on Neighbor-Joining by identifying near-optimal topologies that are significantly different from the best solution found in the search (there are typically many near-optimal solutions that differ only slightly from the best solution; we seek topologically-distant alternatives). In the results below, we first show that the datasets that we examine contain topologically distinct, low-cost solutions. We then demonstrate that the GNJ algorithm can find these low-cost alternative solutions, by examining two measures of success: (1) the number of alternative trees found by GNJ with a near-optimal cost; and (2) the maximal topological (partition) distance between the near-optimal solutions and the optimal solution found.[6] In both tests, we seek the largest number of solutions with cost nearest to optimal, but with topological distance that is far away.

3.1 Comparison of GNJ with exhaustive 8-taxa searches

To judge how effectively the GNJ approach finds alternative topologically-distinct solutions, we first characterized the actual number and diversity of near-optimal solutions by enumerating all 10,395 different trees for datasets with 8 taxa and calculated the cost for each tree topology (Fig. 3). Tree-costs were optimized using either the minimum-evolution criterion or the least-squares criterion. Because the different cost criteria may have different distributions of costs, we plot the number of trees obtained as a function of the fractional cost range: $(c_x - c_{min})/(c_{max} - c_{min})$, where c_x is the least-squares or minimum-evolution cost of a specific tree topology, c_{min} is the minimum (and for exhaustive searches, optimal) cost under that criterion, and c_{max} is the cost of the worst topology. For the 8 taxa data, c_{min} and

[6] In the case of more than 8 taxa, where an exhaustive search for the optimal solution is computationally infeasible, we compare to the best solution found instead of to optimal.

c_{max} are known because the cost of every possible topology has been calculated.[7]

For the synthetic data, it is possible to ask how often the low cost trees found by the GNJ algorithm were consistent with the original tree that was used to produce the distance data. However, the lowest cost least-squares or minimum-evolution tree was often different from the original tree. Trees from type 'A1' and 'B1' data are used for most of the figures because the difference between the original tree cost and the best tree cost was typically between 0 and 0.1 with the median between 0.01 and 0.03 of the cost range. Trees from type 'A1' and 'B1' synthetic data behaved very similarly to trees from the biological datasets. For the 'A2' and 'B2' datasets, the median original tree cost was 0.4 – 0.7 of the cost range. Thus, because of the high external/internal rate ratio, the best tree frequently had a cost substantially lower than the original tree and these datasets have a large number of distinct local minima, which are not seen with the biological datasets or with the type 'A1' and 'B1' trees.

Fig. 3 goes near here.

Fig. 3 shows how the number of trees and the topological distance between the alternative solutions increases over the fractional cost range. The results from three different datasets are shown using either the minimum-evolution or the least-squares cost criterion. In these plots, more challenging datasets have a larger number of near-optimal trees and greater topological

[7]For larger datasets, c_{min} is approximated from the minimum cost obtained for all the tree-searches on the dataset, and c_{max} is approximated from the maximum cost obtained by sampling 100 trees randomly. Thus for the 16 and 32 taxa datasets, c_{min} may not be the optimal minimum cost and c_{max} may not be the highest (worst) cost, but these approximations should differ only slightly from the true values.

distance at lower fractional cost. In general, there are more near-optimal trees with the least-squares criterion than with the minimum-evolution criterion and those trees tend to be more topologically distinct (Fig. 3). For example, with the biological data (Fig. 3C), there were 14.6 trees on average with least-squares cost within 0.01 of optimal, but only 2.6 trees ≤ 0.02 when the minimum-evolution cost is calculated. Furthermore, when the cost is less than 0.01, the maximum topological distance for near-optimal trees is greater for the least-squares trees than for the minimum-evolution trees.

The "branchy" type 'A1' synthetic dataset tends to produce a larger number of near-optimal, topologically distant trees than the type 'B1' (Fig. 3) datasets. When the type 'A2', 'B2' and 'C2' datasets were examined (data not shown), type 'A2' datasets were the most challenging, and, for trees with cost ≤ 0.01, the number of trees and topological distance between the trees was about twice as high for type 'A2' compared to type 'A1'. The biological dataset appears more challenging than the type 'A1', 'B1' and 'B2' synthetic datasets, but less challenging than the type 'A2' dataset (Fig. 3 and data not shown). We focus our attention on the number and diversity of trees with cost-range 0.01–0.05 both because these cost-ranges are intuitively close—between 1% and 5% of the best cost found—and because, for the type 'A1' and 'B1' synthetic data, 0.01–0.05 spans the range of cost differences between the original trees used to generate the distance data and the best trees found for the data.

Ideally, the GNJ algorithm would find each of the near-optimal solutions that can be found when every tree-topology is examined. Thus, we use the number of solutions, their average cost, and their diversity to gauge the effectiveness of GNJ (Fig. 4 – 6), and compare GNJ with an exhaustive search (Fig. 3). We seek combinations of 'Q' and 'D' that approach the distribution of solutions seen in the exhaustive search. Fig. 4A shows that the GNJ algorithm effectively identifies virtually all sub-optimal solutions with costs ≤ 0.05 on the synthetic dataset, as long as $Q > 0$. (Results, not shown, using the minimum-evolution cost criterion are indistinguishable) Only when $Q = 0$, $D = 50$ is the number of near-optimal solutions different from the number found in the exhaustive search; that is, some of the lowest-cost alternative solutions are missed. The curves in Fig. 4B, D report the average maximum topological distance; i.e. the maximum topological distance among all the trees with cost less than the ordinate is determined for each of the 30 datasets, and the 30 maximum distances are averaged. Again, when $Q > 0$ the alternate solutions found by the GNJ algorithm are as diverse as those found by the exhaustive search, for costs within 10% of optimal. (We also examined the maximum topological distances for the data in Fig. 4, and found that they were very similar to the exhaustive search if $Q > 0$, data not shown.)

Fig. 4 goes near here.

The biological 'R1' dataset is more challenging in some ways—there is a larger number of alternate solutions with low cost (Fig. 4C) and the low-cost solutions appear more topologically diverse (Fig. 4D). For the biological dataset, GNJ begins to miss solutions with costs > 0.005 that are found by the exhaustive search. At a fractional cost of 0.01, 11 of 15 solutions are found by GNJ with $Q \geq 25$, and 18 out of 31 are found at fractional cost 0.02. As with the synthetic dataset (Fig. 4B), when $Q = 0$, some of the best near-optimal solutions are missed. The results in Fig. 4 suggest that for small (8 taxa) problems, the GNJ algorithm identifies alternate, near-optimal, topologically-distant solutions very effectively.

3.2 GNJ performance with 16 and 32 taxa

For larger datasets, it is not computationally feasible to examine the solution space exhaustively, so we cannot directly compare the GNJ results to the optimal solution. (Likewise, we cannot guarantee that the lowest-cost solution is optimal, but it is likely to be near optimal.) Nonetheless, we can still evaluate how the GNJ algorithm benefits from saving multiple $K = Q + D$ solutions by examining a range of Q, D pairs (Figs. 5–6). When type 'A1' synthetic datasets with 16 taxa are searched, the largest number of low-cost solutions are again found when $Q > 0$, and the most topologically diverse solutions are found when $D > 0$. (Fig. 5 shows the results using the minimum-evolution cost criterion; results using the least-squares criterion, not shown, are similar.) For these datasets with $K = 100$, the trade-off between quality Q and diversity D is clear-cut. Below 0.01 there is little difference in diversity as Q and D change; above 0.02, $D \geq 50$ gives the best results. On the biological dataset (Fig. 5C), searches with $Q \geq 100$ find almost twice as many (62–78) solutions with fractional cost ≤ 0.01 as searches with $D = 90$ or 100 (33–37 solutions). The difference in performance with respect to Q and D increases at higher fractional costs. However, while reducing D increases the number of low-cost solutions found, it also decreases the diversity of the solution set. For these data, $Q = D = 50$ seems to be the best compromise.

Fig. 5 goes near here.

When searches are performed on 32-taxa data (Fig. 6), the importance of D in improving the diversity of the solutions is more apparent. As before, solutions with $Q = D = 250$ appear to provide a good balance between finding the largest number of low-cost solutions and finding the most diverse solutions. We note that as Q increases from 250 to 500, there is little change in the number of trees with fractional cost ≤ 0.02 on the synthetic type 'A1' dataset (Fig. 6A), and that the maximum topological distance among those solutions increases very little as D increases from 250 to 500. Thus, for this synthetic data, although $K = 500$ retains only a tiny fraction of up to 10^{40} possible 32-taxa tree topologies, the data in Fig. 6A and Fig. 10 suggest that most of the lowest-cost solutions, and many of the topologically diverse solutions, are found.

Fig. 6 goes near here.

3.3 Comparison with other methods

Thus far, our results suggest that GNJ can identify alternative, near-optimal solutions when K ranges from 50 (8-taxa) to 200 (32-taxa). In this section, we compare GNJ with different $K = Q + D$ to two popular phylogenetic tree reconstruction methods for distance data, the Neighbor-Joining method (Saitou and Nei, 1987) and the Fitch-Margoliash algorithm (Fitch and Margoliash, 1967) as implemented in the PHYLIP package (Felsenstein, 1993). As before, we consider both synthetic and biological datasets with different numbers of taxa, and we compare two cost criteria: the minimum-evolution criterion used for Neighbor-Joining searches, and the least-squares criterion used by Fitch-Margoliash. In these tests, we again consider two measures of success: quality (cost) and diversity. We evaluate the quality of the solutions in two ways: (a) the fraction of the time (for the 30 test datasets) that a near-optimal solution is found; and (b) the average cost of the best solutions found. To evaluate diversity, for each distance matrix, we first compute the maximum topological distance between pairs of near-optimal GNJ solutions. Diversity is then measured by computing (a) the maximum, as well as (b) the average of these distances, over 30 datasets.

Fig. 7 goes near here.

For 8-taxa type 'A1' data, GNJ finds solutions of very high quality that are as diverse as the exhaustive search when $K > 5$ and $Q > 2$ (Fig. 7). When $Q > 2$, a solution within a cost range of 0.01 of optimal is found 100% of the time. For $Q = 0$, GNJ finds a solution within 0.01 of optimal less than 20% of the time when $D = 5$, and less than 40% of the time when $D > 5$. On the same datasets, Neighbor-Joining finds a < 0.01 minimum-evolution solutution and Fitch-Margoliash finds a < 0.01 least-squares solution more than 95% of the time. The average cost data in Fig. 7A shows that the best solutions found by Neighbor-Joining and Fitch-Margoliash are typically within 0.01 of the cost range, but those found by GNJ ($K \geq 20$ and $Q \geq 2$) are optimal. Thus, GNJ consistently finds solutions with cost lower than either Neighbor-Joining or Fitch-Margoliash. Moreover, comparison of both the largest maximum topological distance and the average maximum topological distance (Fig. 7B) shows that when the optimal solution was found by GNJ, the diversity of solutions found (with costs < 0.01 of optimal) is as large for the GNJ solution set as for those found by the exhaustive search. GNJ performed as well as the exhaustive seach on the much more challenging type 'A2' data as well (not shown).

Fig. 8 goes near here.

Results for 16-taxa are shown in Figs. 8 and 9. On the synthetic type 'A1' dataset, GNJ found a solution within 0.01 of the best cost 100% of the time when $Q > 0$. For this data, Neighbor-Joining found a < 0.01 cost solution only 80% of the time using the minimum-evolution criterion, while Fitch-Margoliash always found a < 0.01 cost solution.

Once again, GNJ found solutions with lower average cost. For type 'A2' data (not shown), Neighbor-Joining and Fitch-Margoliash found < 0.01 solutions only 30 – 55% of the time for the least-squares criterion, and 25 – 37% of the time for the minimum-evolution criterion while GNJ found the best minimum-evolution solution 100% of the time when $Q \geq 5$. GNJ found the best least-squares solution more than 80% of the time on type 'A2' data with $Q \geq 5$. As K increased from 20 to 200, the cost of the best solutions consistently improved with GNJ. While we cannot compare the GNJ diversity to the diversity that would be found by an exhaustive search, increasing K from 20 to 200 improves the average maximum diversity, and as before, $Q = D$ seems to provide low-cost solutions with high diversity.

Fig. 9 goes near here.

When 16-taxa biological data are examined, the Neighbor-Joining and Fitch-Margoliash algorithms perform quite well (Fig. 9). However, even with this data, the average cost of the best solutions found improves from about 10^{-4} to 10^{-6} when GNJ is used and $Q \geq 25$.

Fig. 10 goes near here.

Neighbor-Joining, Fitch-Margoliash, and GNJ all perform well on 32-taxa type 'A1' (Fig. 10) and biological data (not shown) using a cost threshold of 0.02 or 0.05 (not shown). However, it is surprising how diverse the GNJ solutions are when solutions with costs within 2% of the best cost are included; GNJ found alternate low cost solutions that share fewer than half of the internal edges (two trees share an internal edge if the edge induces the same leaf bi-partitions in both trees).

Comparison of the cost and the diversity of GNJ solutions with $K = 200$ and $K = 500$ suggests that, $K = 500$, which increases the run time 2.5-fold, is probably unnecessary, since neither the quality of the solutions nor the diversity increases significantly with the higher K. Again, using $Q = D$ provides a good balance of quality and diversity.

3.4 Post-processing

Rzhetsky and Nei have observed that for small datasets, NJ solutions are likely to be topologically close to the optimal solution (Rzhetsky and Nei, 1992). We examined how post-processing (described in Section 2.4) affects the number and diversity of the low-cost solutions, and how post-processing might improve Neighbor-Joining, Fitch-Margoliash, and GNJ -based initial solutions. The post-processing algorithm examines all the trees that can be formed by swapping (exchanging) subtrees around each of the internal edges in the tree, thus considering all the alternative trees that are within one partition distance from the initial tree. If a topology is found with a lower cost (least-squares or minimum-evolution), the process is repeated, until no topological neighbor is found with a lower cost. If the GNJ algorithm finds alternate solutions that are on different sides of a single shallow cost basin, post-processing should reduce the number and diversity of low-cost alternate trees. This seems to be the case for the biological 'R1' data (Fig. 11A) and the synthetic type 'A1' data (similar to the biological 'R1' data, not shown). Alternatively, if GNJ actually finds distinct local minima (with respect to cost), the number of trees may decrease dramatically but the topological distance between alternate solutions should remain substantial. Multiple distinct local minima are found with the synthetic type 'A2' data.

Fig. 11 goes near here.

The results of post-processing on the 16-taxa datasets suggest that GNJ is capable of identifying alternate, topologically-distinct local minima when they exist (Fig. 11). As expected, the number of distinct solutions drops dramatically (because of convergence) when the GNJ solutions are post-processed. For the biological 'R1' data (Fig. 11A,C), the drop is more than 30-fold, as it is with the synthetic type 'A1' data (not shown). However, for the synthetic type 'A2' data, which is derived from trees in which the cost for the original tree is frequently mid-way between the best and worst costs, the drop is only 2–3-fold and the average maximum topological distances drops only about 20%. Thus for this very difficult dataset, many of the alternate solutions found by GNJ cannot be reached by local branch-swapping from the best solution, and distinct local minima have been found. Fig. 12 compares the performance of Neighbor-Joining, Fitch-Margoliash, and GNJ, each followed by post-processing, on 16-taxa type 'A2' data. Post-processing improves the performance of Neighbor-Joining and Fitch-Margoliash in finding a solution with cost < 0.01 from about 30–50% success to 50–70% success; GNJ is 100% successful with every combination of K, Q, and D. Again, GNJ finds lower-cost solutions that Neighbor-Joining and Fitch-Margoliash fail to find, even after exhaustive post-processing. For these data, Neighbor-Joining and Fitch-Margoliash appear to sometimes find local minima (with respect to cost), while GNJ finds more global minima.

Fig. 12 goes near here.

The average maximum topological diversity on the difficult type 'A2' data decreases only slightly with post-processing and the maximum topological diversity is as high after post-processing as before. This result—topologically diverse solutions despite a dramatic decrease in the number of low-cost solutions—implies that GNJ has found alternate local minima that cannot be reached by local branch swapping from the lowest-cost solution. Since our post-processing strategy begins from the K solutions found by conventional GNJ, GNJ without post-processing can detect topologically distinct alternative local minima. For the synthetic type 'A2' datasets, low-cost solutions that are topologically distinct local minima appear often. For example, for $Q = 50$ and $D = 50$, the least-squares solutions differ by 11 (out of a maximum 13 possible) branch swaps (partitions) are found in at least one dataset, and differ by 2.9 swaps on average (minimum-evolution solutions differ by as much as 9 partitions, and 4.3 on average). This suggests that topologically distinct solutions have been found by GNJ on 25–50% of the 30 synthetic datasets.

The results with the biological and synthetic type 'A1' datasets contrast starkly to the diversity found with the synthetic type 'A2' data. With the least-squares criterion and post-processing, the average maximum topological diversity is about 0.5 and the maximum diversity is 4, implying that distinct solutions are found in only about 10% of the datasets. With the minimum-evolution criterion, the maximum diversity is the same but the average maximum is 1.1; again, topologically diverse solutions may be found for 25% of these 16-taxa data. For the synthetic type 'A1' data, the average diversity drops from about 4 to 1.1 (least-squares) or from 7 to 2 (minimum-evolution).

Although post-processing can improve the quality of GNJ solutions without significantly reducing their diversity, the time required to post-process K alternative solutions can be prohibitive when the number of taxa (and thus the number of branch swaps that must be tested) is large (> 16). However, comparison of Fig. 12A and the non-post-processed data (not shown) suggests that post-processing does not improve the solution quality significantly when Q and $D \geq 50$, and thus the extra computation is unnecessary.

3.5 Run Time

GNJ uses computation time roughly proportional to the number of partial solutions maintained during execution (K), and cubic in the number of taxa analyzed. Average run times of Neighbor-Joining, Fitch-Margoliash and GNJ for various input sizes are shown in Table 1. GNJ is considerably slower than Neighbor-Joining (which is one of the fastest tree construction algorithms available, because it does not evaluate any alternative trees), and 3 ($K = 200$) to 8-fold ($K = 500$) slower than Fitch-Margoliash for the 32-taxa datasets.

During its execution, GNJ keeps track of K partial solutions. At each iteration, as the next pair of taxa is removed from the "star" tree, GNJ explores all the candidate solutions derivable from the current K partial solutions via a single Neighbor-Joining step. Since each partial tree induces $O(n^2)$ candidate trees by grouping one of the $O(n^2)$ possible node pairs in the tree, the cost of all the resulting candidate trees requires $O(n^2)$ evaluation time. Therefore, each GNJ iteration requires $O(K \cdot n^2)$ time to examine the cost of all $O(K \cdot n^2)$ candidate trees.

At each iteration, GNJ must also select K candidate trees to pass on to the next iteration. In this version, the selection process requires all $O(K \cdot n^2)$ candidate trees to be sorted by cost. Currently, the time required by each iteration of GNJ is dominated by the sorting time which is $O(K \cdot n^2 \cdot \log(K \cdot n^2)) = O(K \cdot n^2 \cdot (\log K + \log n))$. We anticipate that the amount of data to be sorted can be reduced, and that in future versions, the GNJ cost calculation will dominate the run time. Since GNJ has a total of $n - 3$ iterations, the overall run time for GNJ is $O(K \cdot n^3 \cdot (\log K + \log n))$.

4 Discussion

The Generalized Neighbor-Joining algorithm is explicitly designed to explore broadly phylogenetic tree solution spaces and seek low-cost solutions that are topologically distant. To achieve this goal, GNJ maintains multiple partial solutions at each iteration, and incorporates both quality (tree cost) and diversity (topological distance) in selecting the set of partial solutions that will be passed on to the next iteration. The solution space sampling is controlled by the parameters K, Q, and D, which specify the number of partial solutions to retain and the balance between quality (Q) and diversity (D) in selecting alternate solutions.

For datasets of small size (i.e. $\leq$ 10-taxa), GNJ can perform better than Neighbor-Joining and Fitch-Margoliash algorithms by maintaining $K = 20$–50 partial solutions. For example, for biological datasets over 9-taxa, all 8 trees of cost close to the NJ cost (under the least-squares criterion) are obtained by GNJ while maintaining only $K = 50$ partial solutions. For synthetic datasets of 8-taxa, GNJ finds the optimal solution whenever $K \geq 20$ and $Q \geq 2$.

For datasets with 16 or 32-taxa, both the topological diversity and the quality of GNJ solutions improves as K increases. For the datasets that we examined, low-cost solutions were efficiently found with $K \geq 20$ for 8-taxa and $K \geq 50$ for 16-taxa. Increasing K did little to improve either the quality or the diversity of the solutions. 32-taxa problems are far more challenging (and more common in the molecular biology literature). While $K \geq 200$ (32-taxa) was effective in finding low-cost solutions, increasing K improved the quality, and to a lesser extent the diversity, of the 32-taxa solutions.

We believe that the post-processing results show that GNJ is capable of identifying low-cost, topologically-distinct solutions that cannot be found simply by successively examining every topology near to individual low-cost trees. "Falling into local minima" is an inherent flaw of any phylogenetic search method that examines only a small portion of the solution space. The post-processing results for the biological 'R1' data suggest that this data probably has single, very broad local minimum with many different low-cost topologies but very few, if any, alternate solutions that cannot be found by post-processing (local branch swapping). In contrast, the synthetic type 'A2' data does appear to have several distinct local minima, which were found by GNJ. While it is reasuring to learn that GNJ is capable of finding alternative local minima when they exist, more extensive simulations will be required to characterize the conditions under which large numbers of distinct local minima occur. While post-processing may not be necessary to find high quality solutions, the decrease in diversity with post-processing should improve our confidence that a dataset does not have many topologically distinct low-cost solutions.

Our results suggest that GNJ performs best when $Q = D = \frac{K}{2}$, and that $K = 200$ provides an excellent balance between computation time and solution quality/diversity for up to 32-taxa. For more than 50-taxa, $K = 500$ or 1000 may provide better solutions; however, this will depend greatly on the structure of the phylogenetic tree solution space. For large numbers of taxa, one can judge whether a larger K is likely to provide novel solutions by performing searches with 100 and 200. If $K = 200$ does not find any low-cost trees that were missed with $K = 100$, it is unlikely that $K = 500$ (or more) will uncover additional novel trees either.

GNJ is considerably slower than traditional Neighbor-Joining, and for large problems ($\geq$ 32-taxa and $K \geq 200$), it is much slower than Fitch-Margoliash as well. However, a GNJ run is more accurately compared to multiple Fitch-Margoliash searches where the taxa are successively added in different order, a process that can easily increase the amount of time required by 20 to 50-fold. Because GNJ explicitly seeks out topologically diverse solutions, we believe that it is more likely to identify distinct alternatives than additional Fitch-Margoliash trials.

This paper considers the generalization of the neighbor-joining partitioning strategy to which the distance cost measures seem ideally suited. However, the method of retaining many partial solutions during a partitioning strategy can be applied to maximum parsimony methods, and perhaps to maximum likelihood -based approaches as well. We are currently developing a broader generalization of the approach that can be applied to character-based, rather than distance-based, cost criteria.

Acknowledgments

We wish to thank Dr. Douglas Taylor for his comments on our draft. This research was supported by a grant from the National Library of Medicine (LM04961). Gabriel Robins received additional support from an NSF Young Investigator Award and a Packard Foundation Fellowship.

References

Alpert, C. and Kahng, A. B. 1995. Recent directions in netlist partitioning: a survey. *Integration: The VLSI Journal,* **19**, 1–81.

Bandelt, H. J. and Dress, A. 1992. Split decomposition: a new and useful approach to phylogenetic analysis of distance data. *Mol. Phylogenet Evol.* **1**, 242–252.

Day, W. 1987. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bull. Math. Biol.* **49(4)**, 461–467.

Dayhoff, M. O. 1978. *Supplement 3.* Atlas of Protein Sequence and Structure. Natl. Biomed. Res. Found., Washington, D.C.

Felsenstein, J. 1982. Numerical methods for inferring evolutionary trees. *Quar. Rev. Biol.* **57** (1), 379–404.

Felsenstein, J. 1988. Phylogenies from molecular sequences: inference and reliability. *Annu. Rev. Genet.* **22**, 521–565.

Felsenstein, J. 1993. *PHYLIP: Phylogeny Inference Package, version 3.5c*. University of Washington.

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

Foulds, L. R. and Graham, R. L. 1982. The steiner problem in phylogeny is NP-complete. *Advances Appl. Math.* **3**, 43–49.

Huelsenbeck, J. P. 1995. The performance of phylogenetic methods in the four-taxon case. *Syst. Biol.* **44**, 17–48.

Kumar, S. 1996. A stepwise algorithm for finding minimum evolutionary trees. *Mol. Biol. Evol.* **13**, 584-583.

Kimura, M. 1980. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.* **16**, 111–120.

Kuhner, M. K. and Felsenstein, J. 1994. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol. Biol. Evol.* **11**, 459–468.

Leitner, T., Escanilla, D., Franzen, C., Uhlen, M. and Albert, J. 1996. Accurate reconstruction of a known HIV-1 transmission history by phylogenetic tree analysis. *Proc. Natl. Acad. Sci. USA,* **93**, 10864–10869.

Maddison, D. R. 1991. African origin of human mitochondria DNA reexamined. *Syst. Zool.* **40**, 355–363.

Penny, D. and Hendy, M. D. 1985. The use of tree comparison metrics. *Syst. Zool.* **34**, 75–82.

Penny, D., Stee, M. A., Waddell, P. J. and Hendy, M. D. 1995. Improved Analyses of Human mtDNA Sequences Support a Recent African Origin for Homo sapiens. *Mol. Biol. Evol.* **12**, 863–882.

Robinson, D. F. and Foulds, L. R. 1981. Comparison of phylogenetic trees. *Math. Biosci.* **53**, 131–147.

Rzhetsky, A. and Nei, M. 1992. A simple method for estimating and testing minimum-evolution trees. *Mol. Biol. Evol.* **9**, 945–967.

Saitou, N. and Imanishi, T. 1989. Relative efficiencies of the Fitch-Margoliash, maximum-parsimony, maximum-likelihood, minimum-evolution, and neighbor-joining methods of phylogenetic tree construction in obtaining the correct tree. *Mol. Biol. Evol.* **6**, 514–525.

Saitou, N. and Nei, M. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.* **4** (4), 406–425.

Steel, M. A. and Hendy, M. D. 1993. Distributions of tree comparison metrics – some new results. *Syst. Biol.* **42**, 126–141.

Studier, J. and Keppler, K. 1988. A note on the neighbor-joining algorithm of Saitou and Nei. *Mol. Biol. Evol.* **5**, 729–731.

Swofford, D. 1996. *PAUP: Phylogenetic Analysis Using Parsimony (and Other Methods), version 4.0 (test version).* Sinauer Associates, Inc., Sunderland, MA.

Swofford, D. L., Olsen, G. J., Waddell, P. J., and Hillis, D. M. 1996. *Molecular Systematics, D.M. Hillis, C. Moritz and B.K. Mable, eds.* Phylogenetic Inference, pp. 407–514. Sinauer Associates, Inc., Sunderland, MA.

Wilson, A. C. E., Zimmer, E. A., Prager, E. M. and Kocher, T. D. 1989. *The Hierarchy of Life*, Restriction Mapping in the Molecular Systematics of Mammals: A Retrospective Salute, pp. 407–419. Elsevier Press, Amsterdam.

Table 1:

Execution times			
Method	8 taxa	16 taxa	32 taxa
NJ	< 0.01	0.01	0.09
FM	0.08	2.4	31.5
GNJ $K = 20$	0.08	0.8	9.8
$K = 50$	0.2	2.1	25.1
$K = 100$	0.5	4.4	52.1
$K = 200$	1.1	8.8	103.7
$K = 500$	3.1	24.2	262.7

Run time (in CPU seconds on a 167-Mhz UltraSparc) for Neighbor-Joining (NJ), Fitch-Margoliash(FM), and GNJ, averaged over 100 type ‘A’ input datasets of various sizes, using the least-squares criterion and $Q = D$.

Figure Legends

Fig. 1 The Neighbor-Joining heuristic (A) An input matrix of size 5. (B) A Neighbor-Joining solution strategy. First, the star tree topology centered at node 6 is formed; next, the closest neighbor pair $\{1, 2\}$ is "joined" into a distinct internal node 7; finally, this new internal node 7, together with one of the leaf nodes 3, are joined into a new internal node 8 to form T_1. (C) An equally-good solution T_2 but with a very different topology, which was not found by NJ.

Fig. 2 The Generalized Neighbor-Joining Method The GNJ heuristic for the data of Figure 1A is shown. Throughout the search, 3 partial solutions are kept ($K = 3$). At each iteration, all possible neighboring taxa pairs are examined; 3 are selected to pass to the next iteration. The dashed lines represent the neighboring pairs that are eliminated during the current iteration. While NJ follows the path on the left, and thus only finds the single tree T_1, the GNJ scheme recovers both equally-good solutions T_1 and T_2 (see Figure 1).

Fig. 3 Distribution of tree costs and diversity The number of trees (left ordinate, ■, □) and maximum topological (partition) distance averaged over 30 datasets (right ordinate, ●,○) are plotted as a function of the fractional cost range. Distributions for the least-squares (LS, ■, ●) and the minimum-evolution (ME, □, ○) criteria are shown. Distributions were determined for 8 taxa from 30 synthetic type 'A1' datasets, 30 synthetic type 'B1' datasets, and 30 datasets from biological dataset 'R1'. The figures show the results determined after an exhaustive search of all 10,395 tree topologies for 8 taxa.

Fig. 4 GNJ solutions—8 taxa The distribution of solutions found by GNJ on 30 type 'A1' synthetic 8 taxa datasets (A, B) or 30 'R1' biological 8 taxa datasets (C, D) are shown. Searches were done with $K = Q + D = 50$. Panels A, C show the average number of different trees with costs within the fractional least-squares cost shown. Panels B, D show the average of the maximum topological distance of the solutions within the fractional cost range.

Fig. 5 GNJ solutions—16 taxa The distribution of solutions found by GNJ on 30 type 'A1' synthetic 16-taxa datasets (A, B) or 30 'R1' biological 16-taxa datasets (C, D) are shown. Searches were done with $K = Q + D = 100$. Panels A, C show the average number of different trees with costs within the fractional minimum-evolution cost shown. Panels B, D show the average of the maximum topological distance of the solutions within the fractional cost range.

Fig. 6 GNJ solutions—32 taxa The distributions of the number of trees (A) and the maximum topological distance averaged over 30 datasets, (B) are shown for the least-squares cost criterion on 32 taxa synthetic type 'A1' data. Searches were done with $K = Q + D = 500$. Panels (C) and (D) show tree numbers and topological distance for the biological 'R1' data.

Fig. 7 GNJ performance—8 taxa The quality and diversity of GNJ solutions with different values of Q and D are compared to the optimal solution set (opt), and Neighbor-Joining NJ and Fitch-Margoliash FM solutions for synthetic type 'A1' data. (A) The fraction of the time a solution was found with a cost < 0.01 of optimal (squares, left axis) using either the least-squares (filled symbols) or the minimum-evolution (open symbols) criterion. The right axis (circles) reports the cost of the best solution found, averaged over the 30 datasets. (B) The diversity of the $K = Q + D$ solutions with cost < 0.01 is shown as the largest topological diversity found among all 30 datasets (squares) and the maximum topological diversity averaged over the 30 datasets. Closed symbols report diversity for least-squares solutions; open symbols report minimum-evolution diversity.

Fig. 8 GNJ performance—16 taxa Results for 16 taxa type 'A1' synthetic data are plotted as in Fig. 7, except that both the fraction of solutions found and the topological distance plot uses a cost threshold of 0.01.

Fig. 9 GNJ performance—biological data Results for 16 taxa from biological dataset ‘R1’ are plotted as in Fig. 8.

Fig. 10 GNJ performance—32 taxa Results for 32 taxa type ‘A1’ synthetic data are plotted as in Fig. 8, except that both the fraction of solutions found and the topological distance plot uses a cost threshold of 0.02.

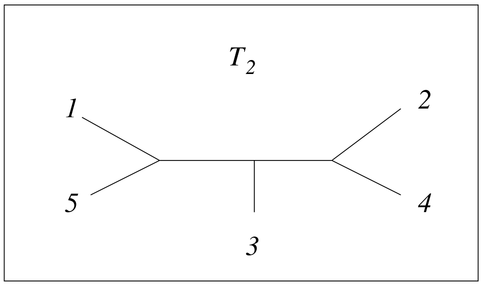
Fig. 11 Number and diversity of post-processed solutions Results for 16-taxa biological data (B, D) and 16-taxa type ‘A2’ data (C, D) are plotted as in Fig. 5. Results shown are for $K = 100$, $Q = D = 50$, with either non-post-processed (squares) or post-processed (circles) solutions. Filled symbols report the distribution of least-squares solutions with fractional cost; open symbols plot minimum-evolution costs.

Fig. 12 Post-processed Neighbor-Joining, Fitch-Margoliash, and GNJ performance Post-processed results for 16-taxa type ‘A2’ data are plotted as in Fig. 8.

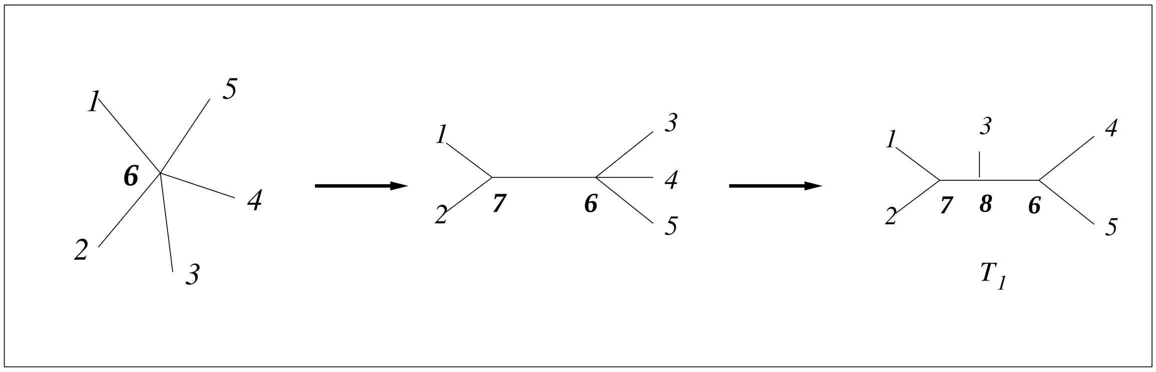
Figure 1: The Neighbor-Joining Heuristic

	s_1	s_2	s_3	s_4	s_5
s_1	0	3	3	4	3
s_2	3	0	3	3	4
s_3	3	3	0	3	3
s_4	4	3	3	0	3
s_5	3	4	3	3	0

(a)



(c)



(b)

Figure 2: Generalized Neighbor-Joining

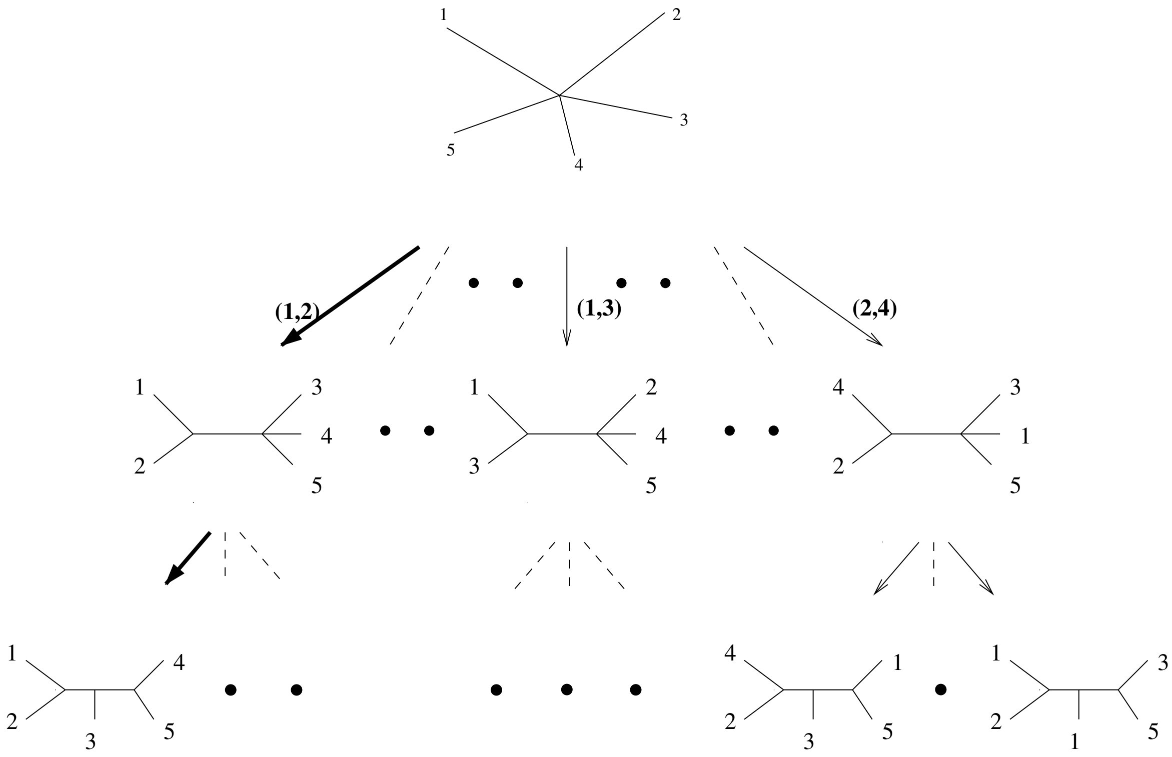
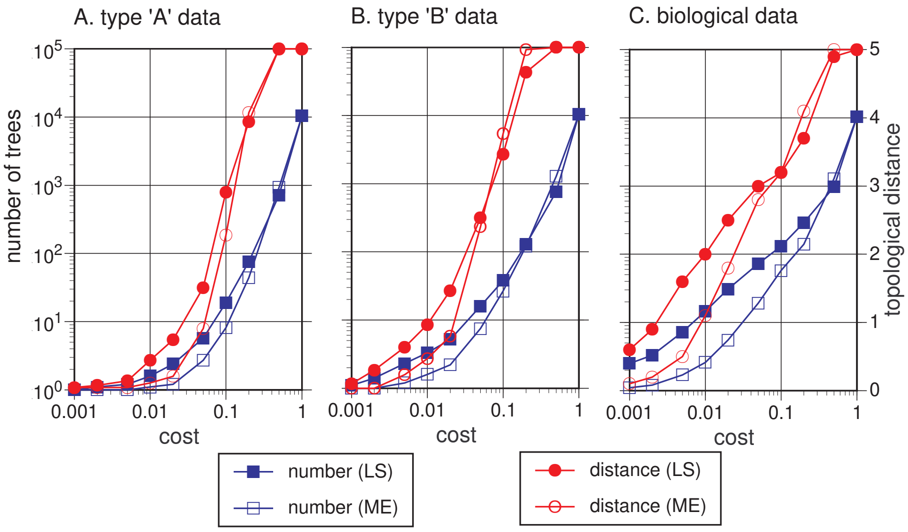
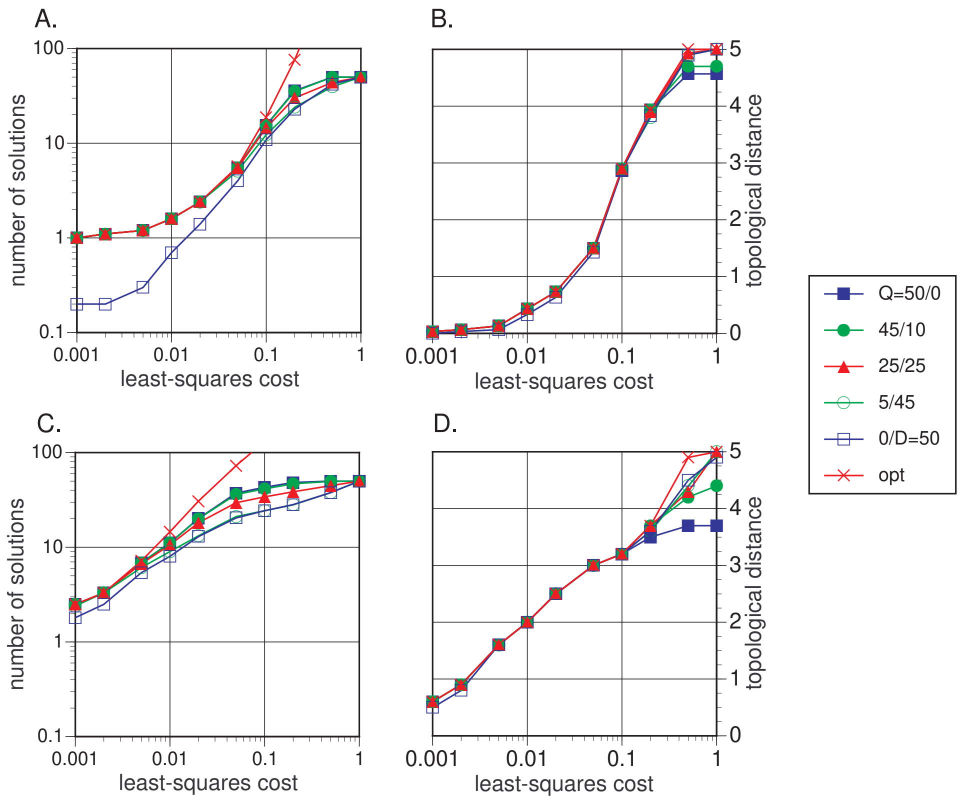


Figure 3:



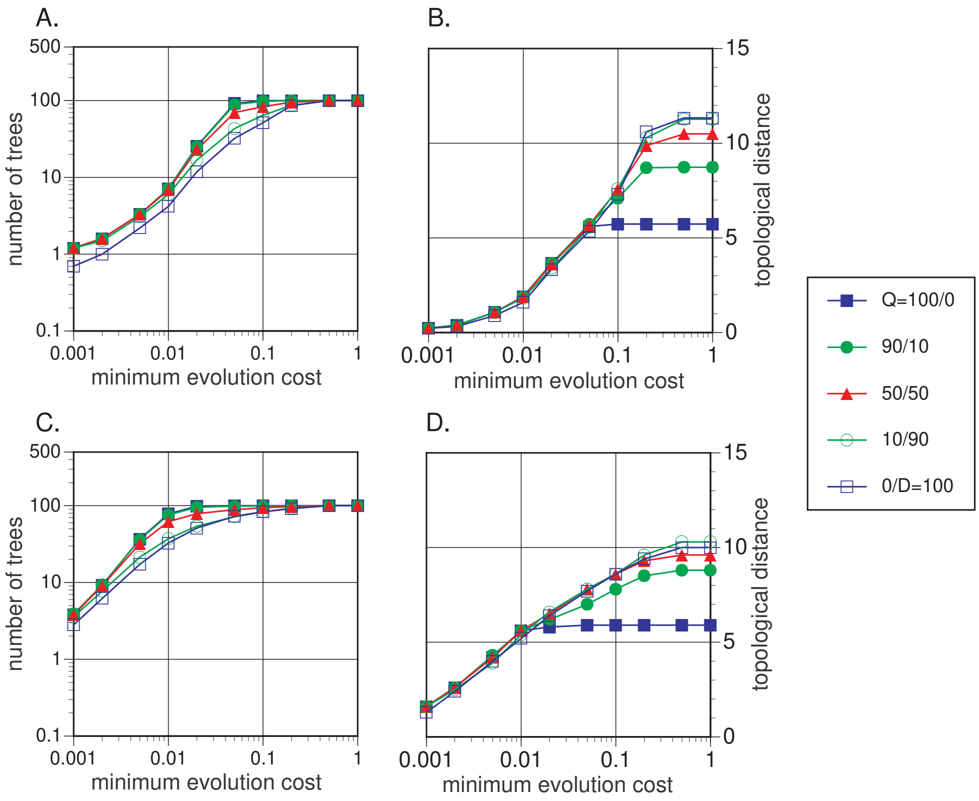
A. type 'A' data
B. type 'B' data
C. biological data
number of trees
topological distance
cost
0.001
0.01
0.1
1
number (LS)
number (ME)
distance (LS)
distance (ME)

Figure 4:



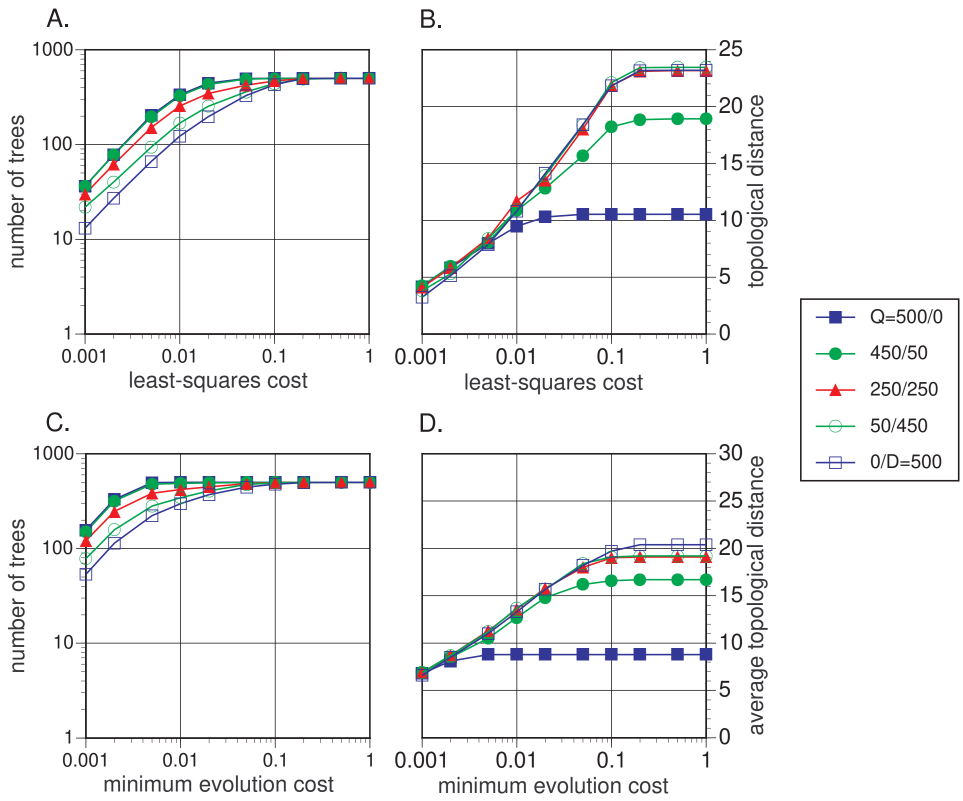
A.
B.
C.
D.
number of solutions
topological distance
least-squares cost
100
10
1
0.1
0.001
0.01
0.1
1
0
1
2
3
4
5
Q=50/0
45/10
25/25
5/45
0/D=50
opt

Figure 5:



A.
number of trees
500
100
10
1
0.1
0.001
0.01
0.1
1
minimum evolution cost
B.
topological distance
15
10
5
0
0.001
0.01
0.1
1
minimum evolution cost
C.
number of trees
500
100
10
1
0.1
0.001
0.01
0.1
1
minimum evolution cost
D.
topological distance
15
10
5
0
0.001
0.01
0.1
1
minimum evolution cost
Q=100/0
90/10
50/50
10/90
0/D=100

Figure 6:



A.
number of trees
1000
100
10
1
0.001
0.01
0.1
1
least-squares cost
B.
topological distance
25
20
15
10
5
0
least-squares cost
C.
number of trees
minimum evolution cost
D.
average topological distance
30
minimum evolution cost
Q=500/0
450/50
250/250
50/450
0/D=500

Figure 7:

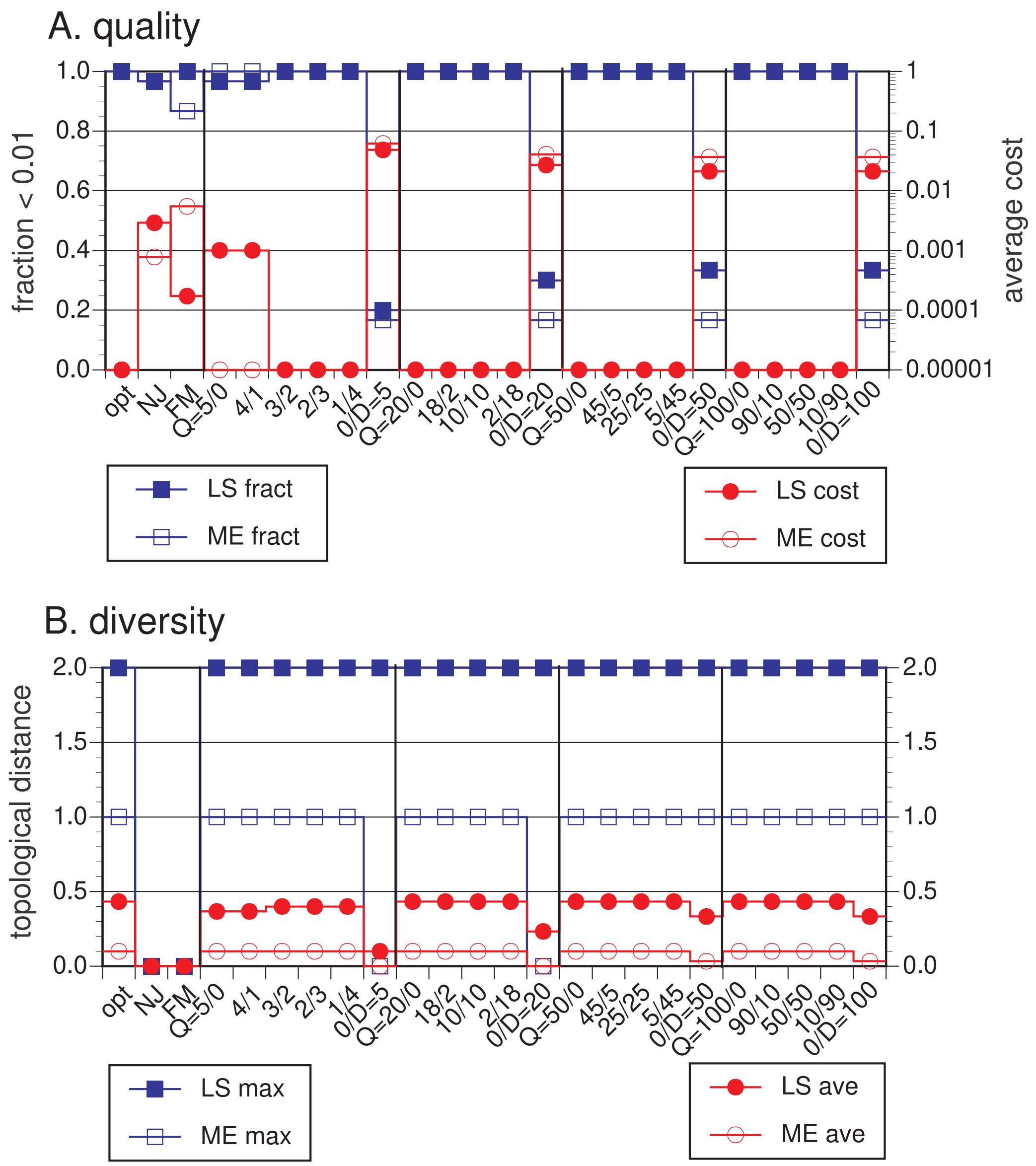

A. quality
fraction < 0.01
average cost
1.0
0.8
0.6
0.4
0.2
0.0
1
0.1
0.01
0.001
0.0001
0.00001
opt
NJ
FM
Q=5/0
4/1
3/2
2/3
1/4
0/D=5
Q=20/0
18/2
10/10
2/18
0/D=20
Q=50/0
45/5
25/25
5/45
0/D=50
Q=100/0
90/10
50/50
10/90
0/D=100
LS fract
ME fract
LS cost
ME cost
B. diversity
topological distance
2.0
1.5
1.0
0.5
0.0
LS max
ME max
LS ave
ME ave

Figure 8:

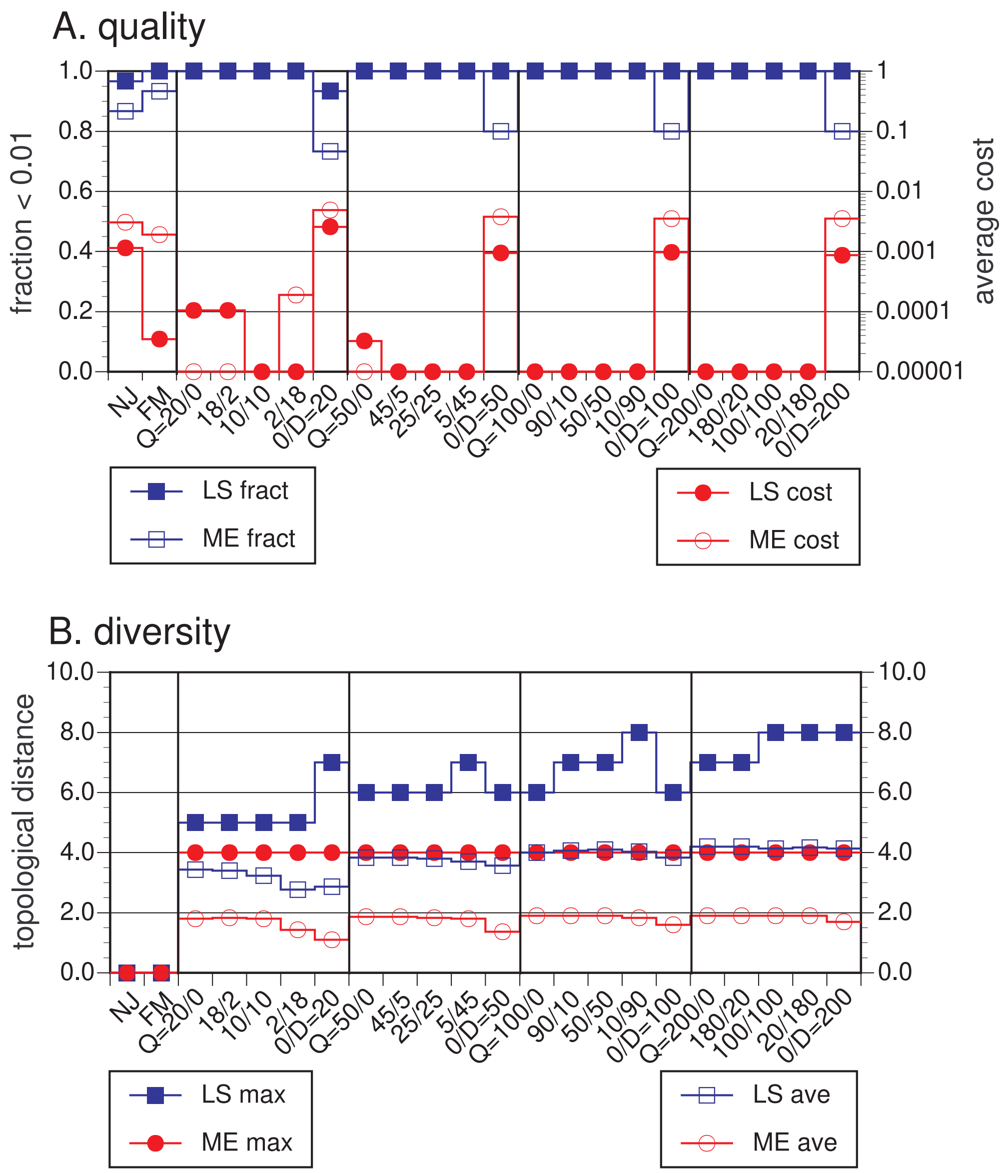

A. quality
fraction < 0.01
average cost
1.0
0.8
0.6
0.4
0.2
0.0
1
0.1
0.01
0.001
0.0001
0.00001
NJ
FM
Q=20/0
18/2
10/10
2/18
0/D=20
Q=50/0
45/5
25/25
5/45
0/D=50
Q=100/0
90/10
50/50
10/90
0/D=100
Q=200/0
180/20
100/100
20/180
0/D=200
LS fract
ME fract
LS cost
ME cost
B. diversity
topological distance
10.0
8.0
6.0
4.0
2.0
0.0
LS max
ME max
LS ave
ME ave

Figure 9:

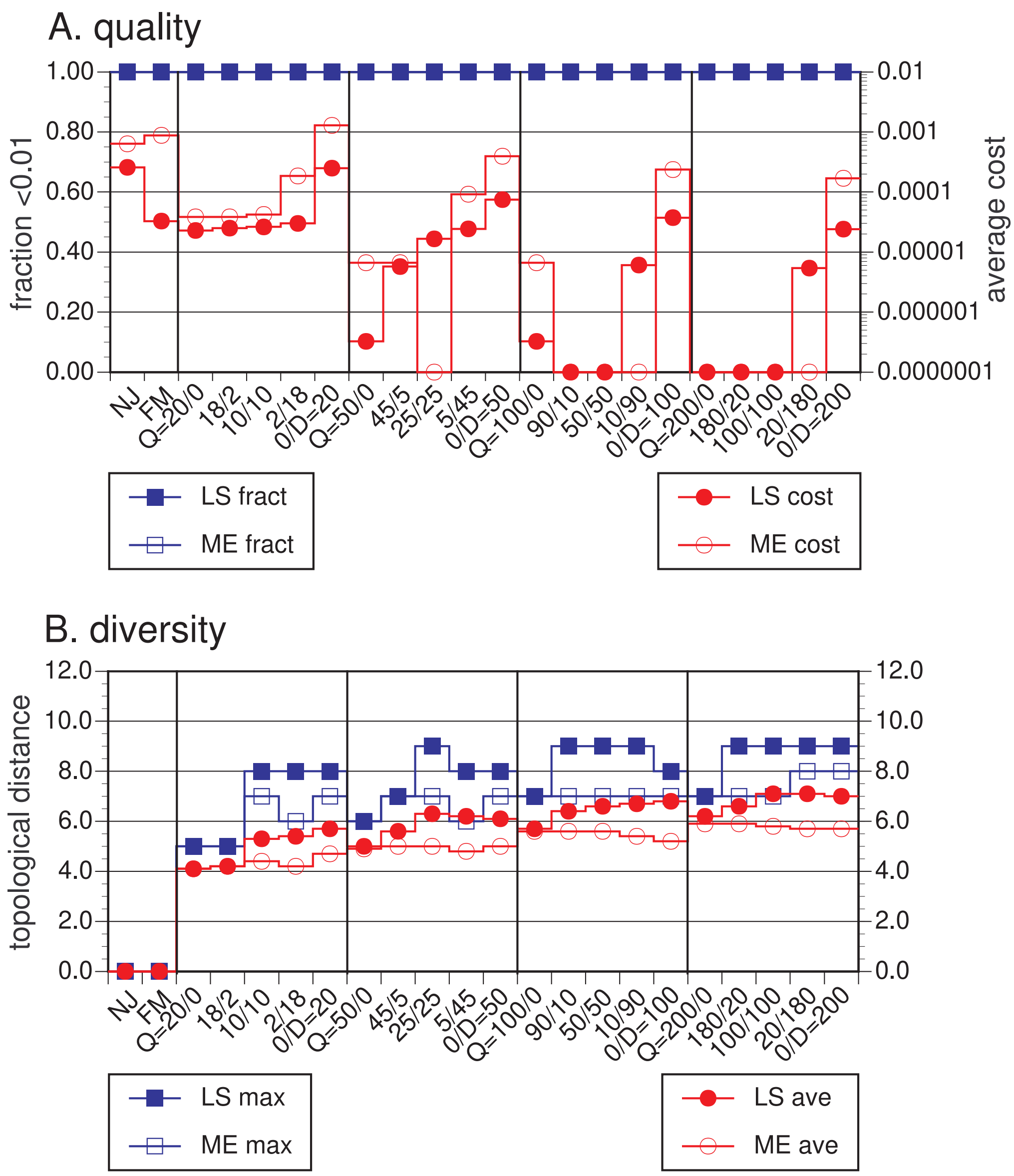
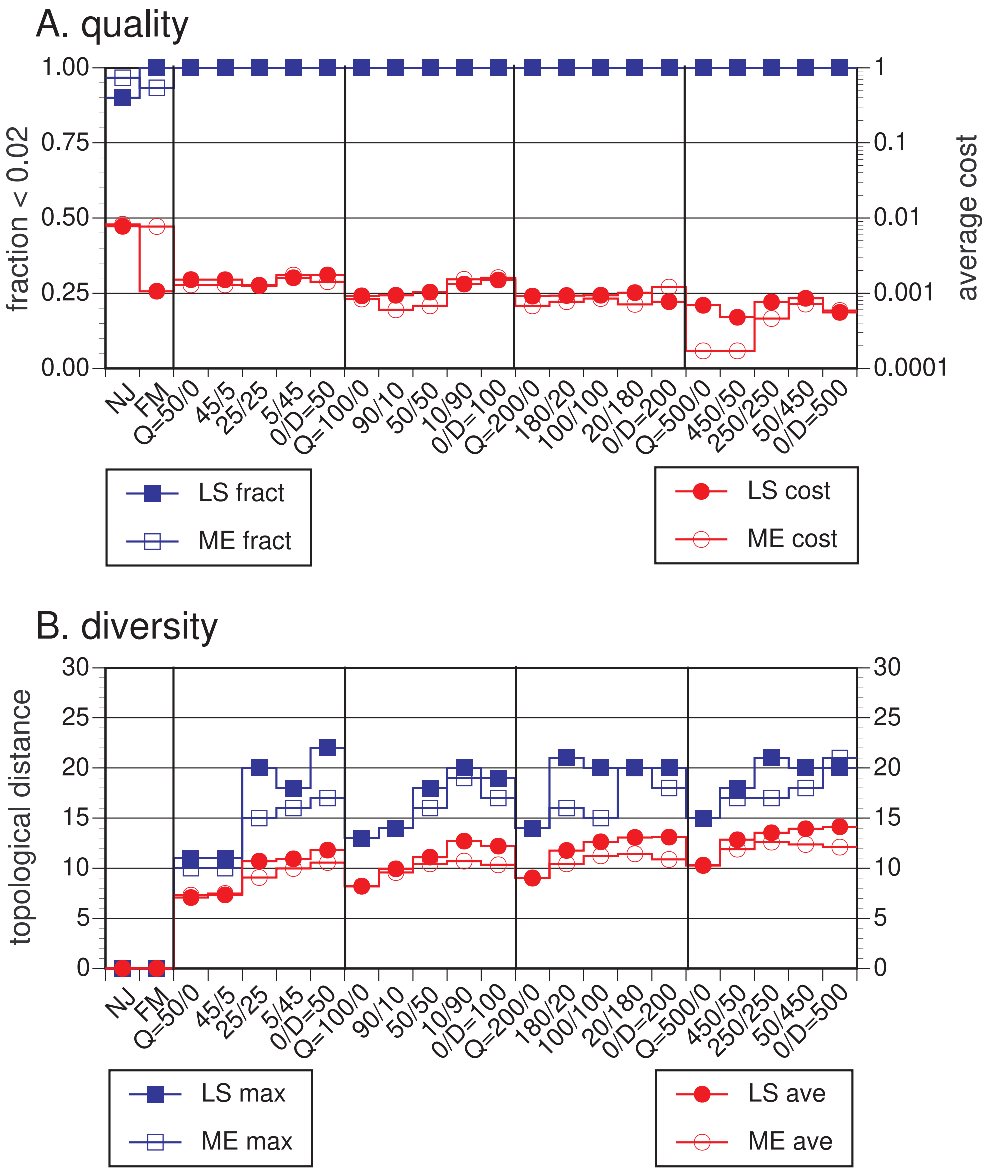
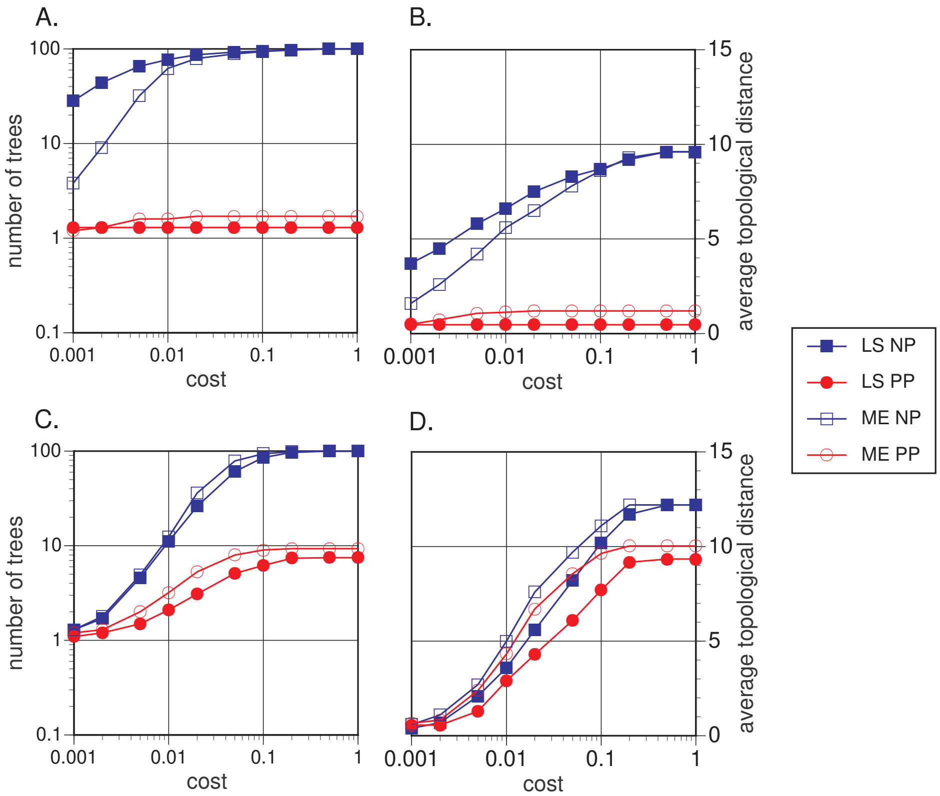

A. quality
fraction <0.01
average cost
1.00
0.80
0.60
0.40
0.20
0.00
0.01
0.001
0.0001
0.00001
0.000001
0.0000001
NJ
FM
Q=20/0
18/2
10/10
2/18
0/D=20
Q=50/0
45/5
25/25
5/45
0/D=50
Q=100/0
90/10
50/50
10/90
0/D=100
Q=200/0
180/20
100/100
20/180
0/D=200
LS fract
ME fract
LS cost
ME cost
B. diversity
topological distance
12.0
10.0
8.0
6.0
4.0
2.0
0.0
LS max
ME max
LS ave
ME ave

Figure 10:



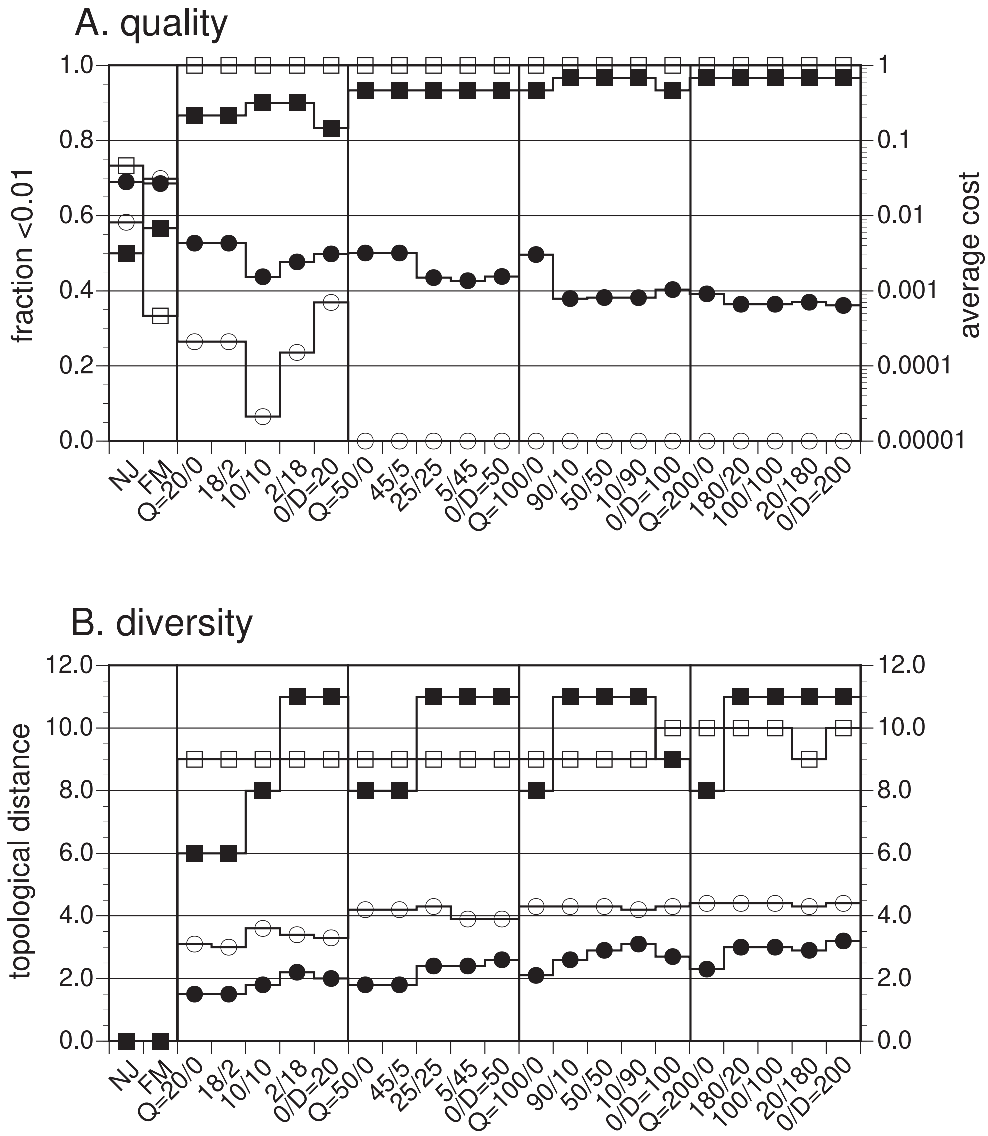
A. quality
fraction < 0.02
average cost
1.00
0.75
0.50
0.25
0.00
1
0.1
0.01
0.001
0.0001
NJ
FM
Q=50/0
45/5
25/25
5/45
0/D=50
Q=100/0
90/10
50/50
10/90
0/D=100
Q=200/0
180/20
100/100
20/180
0/D=200
Q=500/0
450/50
250/250
50/450
0/D=500
LS fract
ME fract
LS cost
ME cost
B. diversity
topological distance
30
25
20
15
10
5
0
LS max
ME max
LS ave
ME ave

Figure 11:



A.
B.
C.
D.
100
10
1
0.1
0.001
0.01
0.1
1
cost
number of trees
15
10
5
0
average topological distance
LS NP
LS PP
ME NP
ME PP

Figure 12:



A. quality
fraction <0.01
average cost
1.0
0.8
0.6
0.4
0.2
0.0
1
0.1
0.01
0.001
0.0001
0.00001
NJ
FM
Q=20/0
18/2
10/10
2/18
0/D=20
Q=50/0
45/5
25/25
5/45
0/D=50
Q=100/0
90/10
50/50
10/90
0/D=100
Q=200/0
180/20
100/100
20/180
0/D=200
B. diversity
topological distance
12.0
10.0
8.0
6.0
4.0
2.0
0.0
NJ
FM
Q=20/0
18/2
10/10
2/18
0/D=20
Q=50/0
45/5
25/25
5/45
0/D=50
Q=100/0
90/10
50/50
10/90
0/D=100
Q=200/0
180/20
100/100
20/180
0/D=200