Optimizing the Order of Taxon Addition in Phylogenetic Tree Construction Using Genetic Algorithm

Yong-Hyuk Kim[1], Seung-Kyu Lee[2], and Byung-Ro Moon[1]

[1] School of Computer Science & Engineering, Seoul National University
Shilim-dong, Kwanak-gu, Seoul, 151-742 Korea
{yhdfly, moon}@soar.snu.ac.kr
[2] NHN Corp., 7th floor, Startower,
737 Yoksam-dong, Kangnam-gu, Seoul, Korea
spin30@soar.snu.ac.kr

Abstract. Phylogenetics has gained in public favor for the analysis of DNA sequence data as molecular biology has advanced. Among a number of algorithms for phylogenetics, the fastDNAml is considered to have reasonable computational cost and performance. However, it has a defect that its performance is likely to be significantly affected by the order of taxon addition. In this paper, we propose a genetic algorithm for optimizing the order of taxon addition in the fastDNAml. Experimental results show that the fastDNAml with the optimized order of taxon addition constructs more probable evolutionary trees in terms of the maximum likelihood.

1 Introduction

As the revolutions in molecular biology have produced a huge amount of DNA sequence data, extracting useful information from them has been considered to be of paramount importance. One of the most important issues includes *phylogenetics*.

Phylogenetics [27] [18] [24] is to infer the most probable evolutionary relationships among species from DNA sequence data. The inferred relationships among species are typically represented by a tree, also called *phylogeny*, which consists of nodes and branches connecting nodes; each node represents a species and each branch represents the amount of genetic variation between two species. It is known that constructing the most probable phylogenetic tree is NP-complete [5] [12]. We are usually interested in the most probable tree in terms of both tree topology and branch lengths.

A number of algorithms for constructing evolutionary trees have been proposed. Parsimony [6] [7] [1] is one of the most popular methods. However, it has a severe problem in that it constructs an inconsistent evolutionary tree when the amounts of genetic changes in different lineages are sufficiently unequal [8].

In contrast to the parsimony which make full use of the information available in the DNA sequence, there have been simpler approaches that exploit only the

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2724, pp. 2168–2178, 2003.

pairwise similarity between DNA sequences. The least-squares [3] is a popular method among them. While the least-squares has explicit statistical justification, it also constructs an inconsistent tree if the rates of evolution are sufficiently unequal in different lineages [4] [8], as in the parsimony.

To estimate more consistent and probable trees, statistical methods using a probabilistic model of evolution are proposed. One of the most robust method is considered to be the maximum likelihood [8], motivated from the earlier probabilistic models of evolution [25]. The approaches using the maximum likelihood can be classified into two categories: constructive approach and non-constructive one.

The constructive approach, which is more popular, builds an evolutionary tree by adding one taxon at a time, starting at an empty tree, with some heuristic information. DNAml [9] and its improved variant, fastDNAml [26], are the representative of them. Although fastDNAml is one of the most widely used method in the phylogenetics literature, its performance is limited due to its incremental nature in constructing trees. In particular, the performance of fastDNAml is notably affected by the order of taxon addition.

As an alternative, non-constructive approaches have also been applied for phylogeny reconstruction. They include all the algorithms without the explicit taxon addition. Recently, evolutionary algorithms such as genetic algorithms [15] [11] [23] have been proposed for constructing evolutionary trees [20] [17] [16] [21]. However, most of them conducted experiments with limited data sets and required considerably high computational cost compared with the constructive approach. They need to be more elaborate to be useful as practical algorithms with reasonable performance.

In this paper, we propose a genetic algorithm for optimizing the order of taxon addition in fastDNAml. The rest of the paper is organized as follows. In Section 2, we describe the maximum likelihood and the fastDNAml. In Section 3, we explain our genetic algorithm in detail and present our experimental results in Section 4. Finally, we make our conclusions in Section 5.

2 Preliminaries

2.1 Maximum Likelihood

Maximum likelihood method [8] is a method for reconstructing phylogenetic trees, or evolutionary trees. Its distinctive feature is that it requires a model of sequence evolution which designates how the sequence evolves. The maximum likelihood method consists of three elements: an evolutionary model, a tree, and the observed sequence.

The maximum likelihood method computes the likelihood of obtaining the observed sequence with a given tree topology, assigned branch lengths, and a given evolutionary model. Since the likelihood is mostly very small, we usually work with log-likelihood rather than the likelihood itself. The log likelihood of obtaining the observed sequence is defined by:

$$\ln L = \sum_{i=1}^{k} \ln L_i$$

where k is the number of sites and L_i is the likelihood of obtaining the nucleotide, one of $\{A, C, G, T\}$, at site i. Based on the maximum likelihood, trees with higher log-likelihoods are considered better.

2.2 fastDNAml

The fastDNAml [26] is one of the most popular programs with reasonable performance and running time. It is an improved version of its predecessor, DNAml [9], in terms of both performance and running time.

The main motivation for the fastDNAml was to reduce the computational cost of DNAml. The DNAml was effective in reconstructing phylogenetic trees with high likelihoods but it required considerably long time to find the trees. To alleviate the cost, fastDNAml uses Newton-Raphson method for finding optimal branch lengths and limits the effort concerning to the branch length optimization. With the two alterations, fastDNAml considerably outperformed the DNAml in terms of both performance and running time.

Figure 1 shows the outline of tree construction in fastDNAml. Note that the phylogenetic tree with three taxa has only one topology. Details for partial tree check and full tree check are described in [26].

The fastDNAml is the representative of the constructive approaches, which build an evolutionary tree by adding one taxon at a time, starting at an empty tree. The performance of fastDNAml is greatly affected by the order of taxon addition. Figure 2 shows two example phylogenetic trees with different orders of taxon addition for an instance with eight taxa (instance *algae*). It suggests that the order of taxon addition can greatly affect the qualities of the resultant trees.

3 A Genetic Algorithm

We propose a genetic algorithm (GA) for finding an optimal order of taxa addition. It conducts a search using an evaluation function related with distance between taxa. The order can be found by enumerating and testing all possibilities. The search space with n taxa has $n!$ elements if all possibilities are considered. Our GA provides an alternative search method to find a good order of taxa addition.

A genetic algorithm hybridized with local optimizations is called a hybrid GA. A considerable number of studies about hybridization of GAs [30] [29] [19] have been proposed. Figure 3 shows a typical steady-state hybrid genetic algorithm. In the next subsection, we describe each part of the hybrid GA that we used for this work.

```
fastDNAml()
// n : the final number of taxa
// i : the number of taxa in the current tree
// A : the next taxon to be inserted
// T_i : the current estimate of the best tree size i
// T_p : the tree after partial tree check (minor changes)
// T_f : the tree after full tree check (greater changes)
// P_i : the set of all the possible tree topologies by adding A to T_i

Compute the optimal tree T_3;
i ← 3;
do {
    Pick the next taxon A;
    Construct the set P_i;
    for each tree in P_i
        { Compute the optimal branch lengths and corresponding likelihood; }
    Set T_{i+1} to be the best tree in P_i;
    i ← i + 1;
    do
        Generate a modified tree T_p from T_i (partial tree check);
    until (none of T_p's is better than T_i);
    T_i ← the best among T_p's;
} until (i = n);
do
    Generate a modified tree T_f from T_n (full tree check);
until (none of T_f's is better than T_n);
T_n ← the best among T_f's;
return T_n;
```

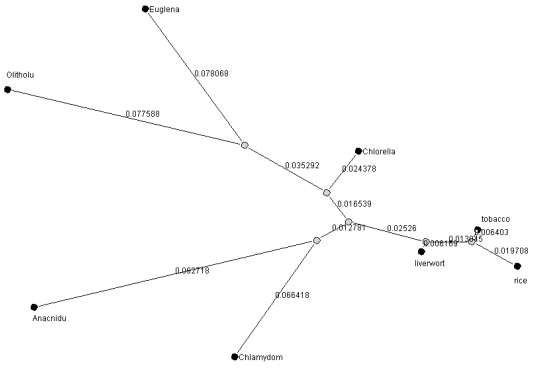
Fig. 1. The outline of tree construction in fastDNAml

3.1 Genetic Operators

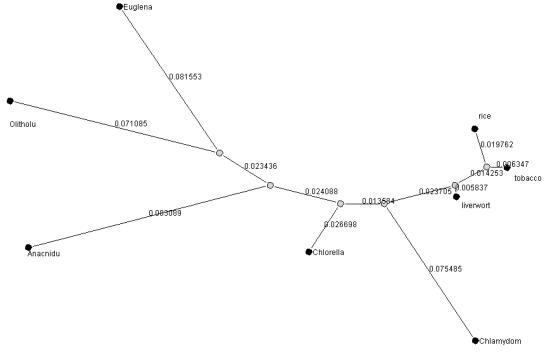
- *Encoding*: A chromosome corresponds to an order of taxon addition. The number of genes in the chromosome is equal to the number of taxa. Each gene corresponds to a taxon.
- *Initialization*: All the chromosomes are created at random. Any valid permutation of order is allowed. We set the population size to be 50 in our algorithm.
- *Selection*: The roulette-wheel-based *proportional selection* is used. The fitness value F_i of a chromosome i is calculated as follows:

$$F_i = (O_w - O_i) + (O_w - O_b)/3$$

where O_w is object value of the worst chromosome in the population, O_b is object value of the best chromosome in the population, O_i is object value of chromosome i. Each chromosome is selected as a parent with a probability proportional to its fitness value.



$$\ln L = -3166.91330$$



$$\ln L = -3158.20339$$

Fig. 2. Two example phylogenetic trees with different orders of taxon addition for *algae*

Create initial population of fixed size;
do {
    Choose parent1 and parent2 from population;
    offspring ← crossover(parent1, parent2);
    mutation(offspring);
    local-optimization(offspring);
    if suited(offspring) **then** replace(population, offspring);
}**until** (stopping condition);
return the best answer;

Fig. 3. A typical steady-state hybrid genetic algorithm

```
iterative-improvement()
// c_i : i^th gene of chromosome C
// f_C : fitness of chromosome C

prev ← f_C;
do {
    flag ← false;
    for all i, j pairs (i < j)
    {
        Swap c_i and c_j;
        current ← f_C;
        gain ← prev − current;
        if (gain < 0) then Swap c_i and c_j; // undo swapping
        else {
            flag ← true;
            prev ← current;
        }
    }
} until (flag = false);
```

Fig. 4. An iterative improvement heuristic

- *Crossover*: Since a chromosome designates an order, an order-based crossover is a natural choice. We use the PMX (Partially Matched Crossover) [10], one of the most popular order-based crossovers. PMX proceeds as follows. 1) Two chromosomes are aligned. 2) Two crossing points are selected at random along the chromosomes, defining a matching section. 3) The genes in the matching section are exchanged. 4) Repair for a valid permutation is performed.
- *Mutation*: Two genes are randomly chosen and swapped. The swaps are repeated for a predetermined times.
- *Local Optimization*: Hybrid genetic algorithms have been considered natural in solving a difficult problem to get desirable performance since genetic algorithms are not so good at fine tuning near local optima. In this study, we use an iterative improvement heuristic for local minimization and it is applied to the offspring after mutation. Figure 4 shows the iterative heuristic.
- *Replacement*: The preselection [2] is used. The offspring replaces the worse parent. The preselection is advantageous in maintaining the diversity of the population.
- *Stopping Criterion*: Our GA stops when one of the two conditions is satisfied: i) the number of generations reaches 5,000, ii) when the fitness of the worst chromosome is equal to the fitness of the best one.

Table 1. Comparison of Two Addition Orders

	Max-relation order	Min-relation order
HIVenvSweden	−1159.93528	−1159.68256
algae	−3159.07438	−3158.20339
hasegawa5	−2682.76961	−2682.75376
exampleTipDate	−3869.25646	−3869.25645

3.2 Evaluation Function

It is ideal to use fastDNAml itself for the fitness evaluation of the GA. However, because of the serious time requirement of fastDNAml, we use a heuristic method for the fitness evaluation of a taxon order.

We performed some experiments to get insights on good orders of taxon addition. Firstly, we tried to iteratively add a taxon that highly relates with previously added taxa. We call this order "Max-relation order." On the other hand, we also tried the opposite. In this heuristic, we prefer a taxon most *unrelated* with previously added taxa, the order is called "Min-relation order." Table 1 shows the ln$L$ scores for some instances by the two addition orders. The results of "Min-relation order" were better than those of "Max-relation order." This result is contrary to our expectation. The performance of "Max-relation order" seems to be limited in that it is likely to form too strong a shape in the early stage of tree construction.

We attempt to find a Min-relation order. We suspect that such an order first makes a global sketch of the tree topology and then adjusts the details. Our GA minimizes the following formula:

$$object\ function = \sum_{i>j} \left(D_{ij} - w \cdot ((n-1) - (i-j))\right)^2$$

where D_{ij} is the gene distance between taxa i and j, n is the number of taxa, and the balancing factor $w = \sum_{i>j} D_{ij} / \sum_{i>j} (i-j)$.

4 Experimental Results

4.1 Data Sets

Nine instances were tested. Table 2 shows the number of taxa and the number of sites for each instance. The number of taxa ranges from 7 to 55. The number of sites ranges from 232 up through 1,485. Brief descriptions about the instances are in the following.

- *HIVenvSweden*: HIV-1 sample of 136 patients from Sweden envelope glycoprotein (*env*) gene, V3 region. Thirteen HIV *env* genes used by Yang *et al.* [35] in developing models of variable selective pressures among sites (the NSsites models).

Table 2. Test Sets

	# of taxa	# of sites
HIVenvSweden	13	273
algae	8	900
hummt25	25	601
green	12	1314
rbcl55	55	1314
hasegawa5	14	232
mtprim9	9	888
exampleTipDate	17	1485
lysozymeSmall	7	390

- *algae*: 16s rDNA data.
- *hummt25*: Twenty five human D-loop sequences used in [34].
- *rbcl55*: Large subunit of RuBisCO gene from chloroplasts. Sequences of the chloroplast gene rbcL from a diversity of green plants, used in [17]. *green* extracted from rbcl55 consists of first 12 taxa of rbcl55.
- *hasegawa5*: Used by Hasegawa *et al.* [13].
- *mtprim9*: mtDNA primate dataset. A mitochondrial segment consisting 888 aligned sites from nine primate species [14], used by Yang [31] to test the discrete-gamma model and Yang [32] to test the auto-discrete-gamma models.
- *exampleTipDate*: Data set of 17 dengo viral strains sequenced at different dates from Andrew Rambaut's TipDate program. This was used for testing the TipDate models of [28].
- *lysozymeSmall*: Primate lysozyme genes of [22], used by Yang [33] in developing tests of positive selection along lineages. This is the "small data set" analyzed in that paper.

4.2 Performance

The main results are given in Table 3. The column "Basic order" shows the $\ln L$ scores by the usual random addition order, and the column "New order" shows the $\ln L$ scores by the addition order obtained by our GA. One can see that the results by "New order" significantly better than those of "Basic order."

Finally, we examine the effectiveness of the object function of Section 3.2. Since the fastDNAml itself requires rather high computational cost, it is impractical to use fastDNAml for fitness evaluation in GA. Although impractical, we replaced the object function by the $\ln L$ score of fastDNAml. This means that we run fastDNAml for evaluation whenever an offspring is created. Table 4 shows the $\ln L$ scores by the best and worst addition orders found by GA. Although some instances were independent of the addition orders (*mtprim9* and *lysozymeSmall*), the results overall shows that the order of addition greatly affects the qualities

Table 3. Comparison of Results

	Basic order	New order
HIVenvSweden	−1160.40239	**−1159.27680**
algae	−3159.07438	**−3158.20339**
hummt25	−1710.83504	**−1706.99035**
green	−8808.40925	**−8800.40369**
rbcl55	−28586.07304	**−28575.65960**
hasegawa5	−2682.91642	**−2682.67452**
mtprim9	−5243.41821	−5243.41821
exampleTipDate	−3869.32158	**−3869.25645**
lysozymeSmall	−924.97205	−924.97205

- The figures in the table are the ln$L$ scores with HKY evolutionary model [13].

Table 4. Results of the Best and Worst Addition Orders

	Worst order	Best order
HIVenvSweden	−1164.16258	−1159.27680
algae	−3166.91330	−3158.20339
hummt25	−1754.03450	−1706.99035
green	−8840.43147	−8800.40369
rbcl55	−28662.63835	−28571.70704
hasegawa5	−2692.47175	−2682.67452
mtprim9	−5243.41821	−5243.41821
exampleTipDate	−3869.52773	−3869.25645
lysozymeSmall	−924.97205	−924.97205

of the resultant trees. It is surprising that except for one instance, *rbcl55*, the ln$L$ scores by the "Best order" are the same as those by the "New Order" in Table 3. This supports the effectiveness of the Min-relation order; we suggest to use it practically. The results of fastDNAml could be improved in this way by the proposed GA.

5 Conclusions

We proposed a hybrid genetic algorithm for optimizing the order of taxon addition in the fastDNAml. Since the performance of the fastDNAml is dependent on the order of taxon addition, we attempted to optimize the order using a genetic algorithm.

Although we improved the fastDNAml with attractive orders, there is still room for improvements. First of all, we need to study more about the relation between the distance among taxa and the taxon addition order.

It is also necessary to incorporate more problem-specific information into the local optimizations. Since a phylogenetic tree with high ln$L$ score often re-

veals new relationship between taxa, it is practically valuable to improve the algorithm. It is left for further study.

Acknowledgments. This work was partly supported by Optus Inc. and Brain Korea 21 Project. The RIACT at Seoul National University provided research facilities for this study.

References

1. J. H. Camin and R. R. Sokal. A method for deducing branching sequences in phylogeny. *Evolution*, 19:311–326, 1965.
2. D. Cavicchio. *Adaptive Search Using Simulated Evolution.* PhD thesis, University of Michigan, Ann Arbor, MI, 1970.
3. R. Chakraborty. Estimation of time of divergence from phylogenetic studies. *Canadian Journal of Genetics and Cytology*, 19:217–223, 1977.
4. D. H. Colless. The phylogenetic fallacy. *Systematic Zoology*, 16:289–295, 1967.
5. W. H. E. Day. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bulletin of Mathematical Biology*, 49(4):461–467, 1987.
6. A. W. F. Edwards. The reconstruction of evolution. *Heredity*, 18:553, 1963.
7. A. W. F. Edwards and L. L. Cavalli-Sforza. Reconstruction of evolutionary trees. *Phenetic and Phylogenetic Classification*, pages 67–76, 1964.
8. J. Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, 17:368–376, 1981.
9. J. Felsenstein. PHYLIP - phylogeny inference package (version 3.2). *Cladstics*, 5:164–166, 1989.
10. D. Goldberg and R. Lingle. Alleles, loci and the traveling salesman problem. In *International Conference on Genetic Algorithms*, pages 154–159, 1985.
11. D. E. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley, Reading, MA, 1989.
12. R. L. Graham and L. R. Foulds. Unlikelihood that minimal phylogenetics for a realistic biological study can be constructed in reasonable computational time. *Mathematical Biosciences*, 60:133–142, 1982.
13. M. Hasegawa, H. Kishino, and T. Yano. Dating the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of Molecular Evolution*, 22:160–174, 1985.
14. K. Hayasaka, T. Gojobori, and S. Horai. Molecular phylogeny and evolution of primate mitochondrial DNA. *Molecular Biology and Evolution*, 5:626–644, 1988.
15. J. Holland. *Adaptation in natural and artificial systems.* University of Michigan Press, Ann Arbor, 1975.
16. K. Katoh, K. Kuma, and T. Miyata. Genetic algorithm-based maximum-likelihood analysis for molecular phylogeny. *Journal of Molecular Evolution*, 53:477–484, 2001.
17. P. O. Lewis. A genetic algorithm for maximum likelihood phylogeny inference using nucleotide sequence data. *Molecular Biology and Evolution*, 15(3):277–283, 1998.
18. W. H. Li. *Molecular Evolution.* Sinauer Associates, Sunderland MA, 1997.
19. F. G. Lobo and D. E. Goldberg. Decision making in a hybrid genetic algorithm. In *IEEE International Conference on Evolutionary Computation*, pages 121–125, 1997.

20. H. Matsuda. Protein phylogenetic inference using maximum likelihood with a genetic algorithm. In *Pacific Symposium on Biocomputing '96*, pages 512–523, 1996.
21. A. Meade, D. Corne, M. Pagel, and R. Sibly. Using evolutionary algorithms to estimate transition rates of discrete characteristics in phylogenetic trees. In *Congress on Evolutionary Computation*, pages 1170–1177, 2001.
22. W. Messier and C.-B. Stewart. Episodic adaptive evolution of primate lysozymes. *Nature*, 385:151–154, 1997.
23. M. Mitchell. *An introduction to genetic algorithms.* MIT Press, London, 1996.
24. M. Nei and S. Kumar. *Molecular Evolution and Phylogenetics.* Oxford University Press, New York, 2000.
25. J. Neyman. Molecular studies of evolution: a source of novel statistical problems. In *Statistical Decision Theory and Related Topics, ed. S. S. Gupta and J. Yackel. New York: Academic Press*, pages 1–27, 1971.
26. G. J. Olsen, H. Matsuda, R. Hagstrom, and R. Overbeek. fastDNAml: a tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. *Computer Applications in the Biosciences*, 10(1):41–48, 1994.
27. R. D. M. Page and E. C. Holmes. *Molecular Evolution: A Phylogenetic Approach.* Blackwell Science, 1998.
28. A. Rambaut. Estimating the rate of molecular evolution: incorporating non-contemporaneous sequences into maximum likelihood phylogenies. *Bioinformatics*, 16(4):395–399, 2000.
29. J. M. Renders and H. Bersini. Hybridizing genetic algorithms with hill-climbing methods for global optimization: two possible ways. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 312–317, 1994.
30. D. Whitley, V. Gordon, and K. Mathias. Larmarckian evolution, the Baldwin effect and function optimization. In *International Conference on Evolutionary Computation*, Oct. 1994. *Lecture Notes in Computer Science*, 866:6-15, Springer-Verlag.
31. Z. Yang. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *Journal of Molecular Evolution*, 39:306–314, 1994.
32. Z. Yang. A space-time process model for the evolution of DNA sequences. *Genetics*, 139:993–1005, 1995.
33. Z. Yang. Likelihood ratio tests for detecting positive selection and application to primate lysozyme evolution. *Molecular Biology and Evolution*, 15:568–573, 1998.
34. Z. Yang and S. Kumar. New parsimony-based methods for estimating the pattern of nucleotide substitution and the variation of substitution rates among sites and comparison with likelihood methods. *Molecular Biology and Evolution*, 13:650–659, 1996.
35. Z. Yang, R. Nielsen, N. Goldman, and A.-M. K. Pedersen. Codon-substitution models for variable selection pressure at amino acid sites. *Genetics*, 155:431–449, 2000.