# A Multi-Stack Based Phylogenetic Tree Building Method

Róbert Busa-Fekete[1], András Kocsor[1], and Csaba Bagyinka[2]

[1] Research Group on Artificial Intelligence of the Hungarian Academy of Sciences and University of Szeged, H-6720 Szeged, Aradi vértanúk tere 1., Hungary
{busarobi,kocsor}@inf.u-szeged.hu
[2] Institute of Biophysics, Biological Research Center of the Hung. Acad. Sci. H-6701 Szeged, P. O. 521., Hungary
csaba@nucleus.szbk.u-szeged.hu

**Abstract.** Here we introduce a new Multi-Stack (MS) based phylogenetic tree building method. The Multi-Stack approach organizes the candidate subtrees (i.e. those having same number of leaves) into limited priority queues, always selecting the $K$-best subtrees, according to their distance estimation error. Using the $K$-best subtrees our method iteratively applies a novel subtree joining strategy to generate candidate higher level subtrees from the existing low-level ones. This new MS method uses the Constrained Least Squares Criteria (CLSC) which guarantees the non-negativity of the edge weights.

The method was evaluated on real-life datasets as well as on artificial data. Our empirical study consists of three very different biological domains, and the artificial tests were carried out by applying a proper model population generator which evolves the sequences according to the predetermined branching pattern of a randomly generated model tree. The MS method was compared with the Unweighted Pair Group Method (UPGMA), Neighbor-Joining (NJ), Maximum Likelihood (ML) and Fitch-Margoliash (FM) methods in terms of Branch Score Distance (BSD) and Distance Estimation Error (DEE). The results show clearly that the MS method can achieve improvements in building phylogenetic trees.

**Keywords:** Phylogenetics – tree estimation – Multi-Stack – tree-joining operator.

## 1 Introduction

The reliable reconstruction of a tree topology from a set of homologous sequence data is one of the most important goals in system biology. A major family of the phylogenetic tree building methods is the *distance-based* or *distance matrix methods*. The general idea behind them is to calculate a measure for the distance between each pair of taxa, and then find a tree that predicts the observed set of distances as closely as possible. There are quite a few heuristic distance-based algorithms with a fixed criterion available for estimating phylogeny, and their strengths and weaknesses are familiar to everyone in the field. The distance-based

I. Mandoiu and A. Zelikovsky (Eds.): ISBRA 2007, LNBI 4463, pp. 49–60, 2007.

methods, like the Unweighted Pair-Group Method using Arithmetic averages (UPGMA) [1] and Neighbor-Joining (NJ) [2], work similarly: they iteratively form clusters, always choosing the best possibility based on a given criterion. We can call these methods greedy in a certain sense, because they always work on the current best candidate subtrees. The NJ method produces additive trees, while UPGMA assumes that the evolutionary process can be represented by an ultrametric tree. These restrictions may then interfere with the correct estimation of the evolutionary process.

The chief aim of this paper is to develop a good distance-based method that closely approximates to the true tree for any available evolutionary (not just for ultrametric or additive) distance. To achieve this we apply a special form of the Least Square Criteria (LSC) to phylogenetic trees [4]. The LSC will guarantee a minimal deviation between the evolutionary distances and the leaf distances in the phylogenetic tree. It is fortunate that the LSC weighting for a phylogenetic tree can be computed in $O\left(n^2\right)$ time. The original LSC was introduced by Fitch and Margoliash, and nowadays several forms of it are in use in the literature, like the Weighted LSC [5], Unweighted and Generalized LSC [6]. We applied the constrained version of LSC (CLSC) here to evaluate phylogenetic trees because the weights of the edges have to be non-negative. The solution of the problem retains its simplicity because the Constrained LSC can easily be handled by the Levenberg-Marquardt method [7].

Since finding the least squares tree (whether it is constrained or not) is an NP-complete problem [8], a polynomial-time algorithm to solve it is unlikely to exist. Many meta-heuristics have been applied in phylogenetic tree-building. We now propose a novel heuristic, based on the so-called Multi-Stack (MS) construction [10]. The MS heuristic organizes the candidate subtrees having the same number of leaves into a priority queue according to their distance estimation error, and generates newer candidate trees by joining the existing trees via a novel tree joining strategy. It may happen however that there are many trees within a priority queue that have a non-disjunct set of leaves, and it is not possible to join them. The Closest-Neighbourhood Tree Joining (CNTJ) strategy introduced here always provides a tree topology based on all of the subtrees, swapping their common taxa with their closest neighbour. Our method was tested on artificial as well as on real-life datasets.

## 2 Background

### 2.1 Phylogenetic Trees

A tree is a connected acyclic graph. First we denote the vertex set and the edge set of a tree $T$ by $V(T)$ and $E(T)$, respectively. Furthermore, let us denote the non-negative weights of the edges by $w : E(T) \rightarrow \mathbb{R}_{\geq 0}$. A weighted tree assigns a distance for each pair of leaves (which can be calculated by summing the weight of the edges on the path between them) that is called the *leaf* distance of $T$, and will be denoted by $D^T$. A *phylogenetic tree* is represented as a leaf-labelled weighted binary tree. The labels of the leaves of phylogenetic tree $T$ correspond to

the set of taxa $\mathcal{T}_T$. The inner nodes represent the hypothetical ancestors, and the weighting of the phylogenetic tree represents the evolutionary distance defined by $T$. If we regard $T$ as a rooted tree, then there is only one internal node of degree 2; the degrees of the other internal nodes are 3. Here we will only deal with rooted phylogenetic trees. The subset of the descendants of an internal node is called a cluster, and the internal nodes are the most recent *common ancestors* of the *monophyletic group* or *cluster*. Thus the internal nodes of a phylogenetic tree and clusters are equivalent concepts. This way each phylogenetic tree corresponds to a set of compatible clusters $\mathcal{C}$ (i.e. for all $A, B \in \mathcal{C}$ either $A \subseteq B$, or $B \subseteq A$, or $A \cap B = \emptyset$). This construction is also called a *Linnean Hierarchy*. Here we will denote the clusters of $T$ by $T^{\mathcal{C}}$.

The Robinson-Foulds (RF) distance or *symmetric difference* for rooted trees [11] is based on this approach as well. Because the RF distance of two rooted phylogenetic trees $T_1$ and $T_2$ is the cardinality of the symmetric difference of their cluster sets, $T_1^{\mathcal{C}} \triangle T_2^{\mathcal{C}} = \left(T_1^{\mathcal{C}} \setminus T_2^{\mathcal{C}}\right) \cup \left(T_2^{\mathcal{C}} \setminus T_1^{\mathcal{C}}\right)$. There is also an extension of the RF distance introduced by Kuhner and Felsenstein [12], and it is known as the Branch Score Distance (BSD).

### 2.2 Constrained Least Squares Criterion

There are many criteria in use for phylogenetic trees that can be applied to the distance data, like Minimum Evolution Length and Least Squares (LSC) Criterion. There are many forms of Least Squares Criteria available, and all of them require the optimization of a quadratic function. Before we formally describe these criteria, let us denote the *path-edge incidence* or *topology matrix* of a phylogenetic tree $T$ by $P_T$. The matrix $P_T$ is a binary matrix whose columns correspond to the edges of $T$, while the rows correspond to the paths between the leaves of $T$. This representation of a tree requires a space of $O\left(n^3\right)$ because it has $n-1$ columns and $\binom{n}{2}$ rows, even though it has just a few non-zero elements. Hence it is worth exploiting the sparsity of the topology matrix for an efficient implementation.

Next, we will denote a distance matrix by $D$ defined on the taxon set of $\mathcal{T}$. We can rewrite $D$ using its vector form $\mathbf{d}$ (i.e. turning the upper triangular of $D$ into a vector). The arrangement according to the topology matrix $P_T$ determines an unambiguous ordering among the $\binom{n}{2}$ entries of the vector $\mathbf{d}$. Introducing the necessary notations, we can write, in a simple way, the Unweighted Least Square Criteria (LSC) for a given $T$ phylogenetic tree. The edge weighting of a tree $T$ satisfies the LSC criteria if it satisfies the following optimisation task:

$$\min_{\mathbf{x} \in \mathbb{R}^{\mathbf{n-1}}} \| \left(P_T \mathbf{x} - \mathbf{d}\right) \| \tag{1}$$

where the elements of $\mathbf{x}$ may have any real value, including zero or negative values. The solution of the problem defined by Eq. (1) results in an optimal edge weighting for a phylogenetic tree $T$ and a minimal Frobenius norm for $\|D_T - D\|_F$. This means that the deviation between the calculated weighting $D_T$ and $D$ is minimal. The problem can be solved in $O\left(n^2\right)$ time for a given phylogenetic tree [5]. We also require that a weighting always be positive because

a negative evolutionary distance has no physical sense. That is why we will restrict ourselves here to the following minimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^{\mathbf{n-1}}} \| (P_T\mathbf{x} - \mathbf{d}) \| \quad s.t. \quad \mathbf{0} \leq \mathbf{x} \tag{2}$$

The Constrained Least Squares Criteria (CLSC) defined above results in a non-negative weighting for a phylogenetic tree $T$. CLSC also retains the property of the original LSC that it can be solved in $O\left(n^2\right)$ time, because the algorithm introduced by Bryant & Waddell [5] can handle the Levenberg-Marquardt method as well. Here $\|D_T - D\|_F$ is the *distance estimation error (DEE)*, and

$$\frac{\|D_T - D\|_F}{\binom{n}{2}}$$

is the *normalized distance estimation error (NDEE)* of the phylogenetic tree and will be denoted by $e_T$.

## 3 Materials and Methods

### 3.1 Multi-Stack Approach

To solve problems which have enormous solution spaces we need to apply an efficient search technique. That is why we decided to adopt a heuristic approach for phylogenetic tree-building which is also used in speech recognition [13]. To describe the method we first have to give a definition. A *stack* is a structure for keeping candidate solutions in. Furthermore, we use limited-sized stacks: if there are too many candidates in a stack, we prune the ones with the highest fitness value. In the MS algorithm we assign a separate stack to the trees having the same number of leaves and store the K-best candidate subtrees in the stack according to their DEEs. In the initial step the algorithm generates the lower level subtrees only, and then it pops each pair of candidate subtree from the stacks, joins them in every possible way, and afterwards puts the new candidate subtrees into the stack according their leaf numbers associated with their new DEE. Applying this heuristic to phylogenetic tree building, we obtain an iterative tree-building procedure.

The pseudocode of the MS algorithm is presented in Table 1, where the $Q_n$ elements denote the limited priority queue which contains at most $K$ trees, and each tree has exactly $n$ leaves in it. The initial step of the method includes the exploration of all tree topologies with at most three leaves. This step makes sense because there are only $n + \binom{n}{2} + \binom{n}{3}$ phylogenetic trees when $|\mathcal{T}| = n$, so during this initial step we can explore the whole space of trees. In the next steps MS generates the possible subtrees, and it always keeps the best $K$ subtrees based on their distance estimation error.

The complexity of the MS method naturally depends on the variable $K$ because we are exploring an $n^2K^2$ tree topology. Since CLSC requires a quadratic

**Table 1.** The Multi-Stack algorithm

| Input: | D distance matrix, $K$ size of priority queues |
|---|---|
| 1 | Initial step: fill up $Q_1, Q_2$ and $Q_3$ |
| 2 | for $i = 3 : n$ |
| 3 | for $j = 1 : min(n - i, i)$ |
| 5 | Generate all of the trees joining the elements of $Q_i$ and $Q_j$ |
| 6 | Add them to the priority queue $Q_{i+j}$ according their DEE |
| 7 | endfor |
| 8 | endfor |
| 9 | return to first element of $Q_n^K$ |
| Output: | Rooted phylogenetic tree $T$ with $n$ leaves |

time complexity ($O\left(n^2\right)$), it becomes the most time-consuming step of the MS. Due to this features the MS tree building method has a time complexity of $O\left(K^2n^4\right)$ overall. This computation also includes the time requirements of the joining step which will be introduced in the next section.

### 3.2 Closest Neighborhood Tree Joining Operator

With the Multi-Stack tree building approach it may happen that we want to join two candidate trees that have some common taxa. The simplest idea is the naive approach: let us replace the common taxon set of the candidate trees that interferes the tree joining in every possible way with those taxa that do not occur in the taxon set of candidate trees. After we have carried out and evaluated all possible replacements, let us choose the best replacement. But it can be easily seen that this will lead to a very high computational burden, because the number of possible replacement grows exponentially with the number of the common taxon set. Instead here we suggest a tree joining strategy as a way of avoiding this problem.

We need to join two candidate subtrees $T_1, T_2$ having $n_1$ and $n_2$ leaves respectively, and we need to determine a strategy for the elimination of the duplicated taxa of the candidate trees: $|\mathcal{T}_{T_1} \cap \mathcal{T}_{T_2}| = k$. From the solution of this problem we also require that the distance estimation errors $e_{T_1}$ and $e_{T_2}$ with respect to the applied distance matrix $D$ remain or grow as little as possible. Thus the goal here is to determine a strategy for the replacement of common taxa that produce the least variation in the tree estimation errors of the candidate trees in question.

For the formal description let us denote the cost of the replacement for a taxon $t \in \mathcal{T}_T$ by $c(t, t^{'})$, where $t^{'} \in \mathcal{T} - \mathcal{T}_T$. This leads to a change in the $e_T$ value after the replacement, which can be a negative real number as well. We can readily determine an upper bound for this cost, because using the weights of $T$ before the replacement, the following proposition always hold.

**Proposition 1.** *Let $T$ be a phylogenetic tree with a taxon set $\mathcal{T}_T \subset \mathcal{T}$, and let $e_T$ be its distance estimation error. A distance on $\mathcal{T}$ will be denoted by $D$,*

*and the leaf distance will be denoted by* $D^T$. *Now let* $t \in \mathcal{T}_T$ *and* $t^{'} \in \mathcal{T} - \mathcal{T}_T$ *be two taxons. Then the following inequality will hold for the* $c(t, t^{'})$ *cost of the replacement:*

$$c(t, t^{'}) \leq \sum_{t^{''} \in \mathcal{T}_T} b_{t^{''}}(t, t) - b_{t^{''}}(t^{'}, t) \tag{3}$$

*where* $b^1_{t^{''}}(t_1, t_2) = \left| D(t_1, t^{''}) - D^T(t_2, t^{''}) \right|$

*Proof.* If we use the CLSC for $T$, then we get an optimal edge weighting $w$ using the taxon set $\mathcal{T}_T$ and distance matrix $D$. Equation 3 corresponds to the rows in Equation 2, that is, it is represents the path between $t$ and $\mathcal{T} - t$. Thus if we replace this taxon, the change of the optima will vary according to the magnitude when we use the weighting $w$. So Equation 3 will hold apart from the choice of $t^{'} \in \mathcal{T}$.

Summarizing the above points, Proposition 1 allows us to determine an upper bound for a replacement of a taxon. That is why we suggest here that the common taxon set $|\mathcal{T}_{T_1} \cap \mathcal{T}_{T_2}|$ should be replaced iteratively, taxon by taxon, always choosing the pair of taxons that have the lowest bound in accordance with Proposition 1.

### 3.3 Distances and Similarities

**Evolutionary distances.** The global alignment of protein sequences can be performed using the well-known Needlemann-Wunsch [14] algorithm with the BLOSUM70 [15] matrix. The simplest evolutionary distance between a pair of aligned sequences is usually measured by the number of sites where a substitution occurs. Many models have been proposed to describe the true evolutionary process. There are many corrections of this measure which try to fine tune the evolutionary rate. Some of them were used here when we performed our tests on different real-life datasets. These include the Gamma, Poisson and Jukes-Cantor corrections [16].

**Compression-based similarity measures.** The *information theoretical distance* functions are based on a comparison of how many information sequences there are relative to each other. This approach originated from Kolmogorov-complexity theory. The Conditional Kolmogorov complexity $K(X|Y)$ is defined as the length of the shortest program computing $X$ on an input $Y$ [19]. The Kolmogorov complexity $K(X)$ of a sequence $X$ is a shorthand notation for $K(X|\lambda)$, where $\lambda$ is an empty sequence. The corresponding distance function uses the relative decrease in complexity or conditional complexity as a measure of sequence similarity, that is

$$d(X, Y) = \frac{max\{K(X|Y), K(Y|X)\}}{K(YX)} \tag{4}$$

Kolmogorov complexity is a non-computable notion, so in practical applications it is approximated by the length of a compressed sequence calculated with

a compression algorithms like LZW [20] or Sequitur [21]. The formula for calculating compression-based similarity measures (CBM) using the length values of compressed sequences can be derived from Equation 4. It takes the form

$$d_{CBM}(X,Y) = \frac{C(XY) - min\{C(X), C(Y)\}}{max\{C(X), C(Y)\}} \tag{5}$$

where $C(.)$ denotes the length of a compressed sequence, compressed by a particular compressor C. We will focus on two well-known compressor algorithm in our experiments, namely LZW [20] and Sequitur [21].

### 3.4 Generation of Model Populations

Since the correct phylogeny for a set of taxa is usually unknown, we first carried out our tests on randomly generated model populations having $10-20-30-40$ members. For each population 100 independent and identically-distributed and and non ultrametric model trees were generated from the tree-space. In order to calculate the leaves of these trees, pseudo random sequences of 600 amino acids were used as ancestor sequences. The sequence was then assumed to evolve according to the predetermined branching pattern of the randomly generated model tree. The edge lengths of the generated tree correspond to the expected number of amino acid substitutions per site. We varied this value between $0-0.1$, and the number of amino acid substitutions at each site was assumed to have a Poisson distribution [17,18], were also used to mimic the mutations. Using this rate we carried out point mutations according to the BLOSUM70 matrix. Hundred different set of sequences (model populations) were generated for each (10-20-30-40) member number.

### 3.5 Description of Real-Life Datasets

We utilized three different datasets of various size to compare and test the methods. Primates consist of mitochondrial DNA, while hydrogenases and myoglobins are distinct protein families, hence they are very suitable objects for statistically testing different tree building and distance (similarity) calculating procedures.

The set of *primates* is quite small (12 sequences), and it was borrowed from Ovchinnikov et al. [22]. This dataset contains the mitochondrial DNA of two Neanderthals, the modern human species and other vertebrates. The second set we used for testing is a typical set of sequences of *myoglobins*. It contains 27 proteins from different organisms. The third set is the group of 75 [NiFe] *hydrogenases*. Hydrogenases are metalloenzymes that catalyze the reaction $H_2 \rightleftharpoons 2H^+ + 2e^-$. They can be found in bacteria, archae and cyanobacteria. The [NiFe] hydrogenases are usually placed into 4 different taxonomic groups [23]. In the rest of the paper these datasets will be called *primates, myoglobins and hydrogenases* respectively.

# 4 Experiments

## 4.1 Evaluation of the Model Populations

The evolutionary distances (Poisson distance and CBM similarity measure with the Sequitur compressor method [21]) were calculated for the model populations, and phylogenetic trees were built over these model populations using four different tree-building methods: Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [1], Neighbour-Joining (NJ) [2], Maximum Likelihood method for proteins (ML) [24] and the Fitch-Margoliash (FM) [4] method, all of which were implemented in the Phylip package [25], and our newly developed Multi-Stack method (MS). The parameter $K$ for the MS method was set to 20 for the populations having 10 and 20 members (leaves) and to 40 for the populations with 30 and 40 members (leaves).

The BSD distance between the randomly-generated model tree and the built phylogenetic tree along with the distance estimation error (DEE) were calculated after building the phylogenetic tree. The test was repeated 100 times on 100 similar model populations and the average of BSD distance and DEE was calculated. The results of this are summarized in Table 2. It is striking that the MS method is superior to all other methods tested. Both the BSD and the DEE values are smaller in every case when the MS method was applied. The UPGMA approach in contrast proved to be the least efficient method in reconstructing phylogenetic trees. The performances of the NJ and the FM method are quite similar, and the means of the BSD distance are equal to each other in many test cases. The mean of the Normalized DEE and BSD distances for NJ and FM only lags behind the results of the MS method by a small amount when the leaf number is set to 40.

**Table 2.** The performance of the test on randomly generated model trees. The values in bold show the minimal value in each row.

| | No | Length of ancestor = 600 | | | | |
|---|---|---|---|---|---|---|
| | leaves | UPGMA | NJ | FM | MS | ML |
| Poisson-Poisson DEE(*$10^3$) | 10 | 60.83 | 24.69 | 25.13 | **7.39** ($K = 20$) | 55.6 |
| | 20 | 41.10 | 16.62 | 16.58 | **10.33** ($K = 20$) | 37.5 |
| | 30 | 30.45 | 11.85 | 11.97 | **6.85** ($K = 40$) | 29.9 |
| | 40 | 24.86 | 9.37 | 9.35 | **5.12** ($K = 40$) | 21.6 |
| Poisson-Sequitur DEE(*$10^3$) | 10 | 122.55 | 34.79 | 35.90 | **17.49** ($K = 20$) | 191.4 |
| | 20 | 70.58 | 16.58 | 16.32 | **11.05** ($K = 20$) | 101.3 |
| | 30 | 48.45 | 10.31 | 10.39 | **6.61** ($K = 40$) | 67.8 |
| | 40 | 37.32 | 6.16 | 6.06 | **5.50** ($K = 40$) | 69.1 |
| Poisson-Poisson BSD distance | 10 | 0.32 | 0.21 | 0.22 | **0.20** ($K = 20$) | 0.28 |
| | 20 | 0.50 | 0.33 | 0.34 | **0.28** ($K = 20$) | 0.33 |
| | 30 | 0.63 | 0.41 | 0.41 | **0.32** ($K = 40$) | 0.56 |
| | 40 | 0.73 | 0.48 | 0.48 | **0.38** ($K = 40$) | 0.62 |
| Poisson-Sequitur BSD distance | 10 | 0.49 | 0.27 | 0.29 | **0.29** ($K = 20$) | 0.34 |
| | 20 | 0.79 | 0.44 | 0.44 | **0.32** ($K = 20$) | 0.39 |
| | 30 | 0.95 | 0.54 | 0.54 | **0.35** ($K = 40$) | 0.59 |
| | 40 | 1.15 | 0.63 | 0.62 | **0.39** ($K = 40$) | 0.62 |

### 4.2 Real-Life Datasets

The newly developed MS method was tested on real-life datasets as well. To evaluate the trees we used the distance estimation error values (DEE). The properties of the MS method were investigated and the results were again compared with other tree building algorithms. In this case we applied them on six evolutionary distances (similarities) (Jukes-Cantor, Gamma, Poisson, LZW, Sequitur and alignment score).

The only tunable parameter for the MS method is $K$ (the size of the limited priority queue). When evaluating our MS trees we always chose the best tree in the last stack, i.e. the one which had the lowest DEE value. It is interesting to see the "goodness" of different trees in the last stack i.e. how much the "best tree" was better than the others. We set the value of parameter $K$ to 30 and plotted the DEE value of the trees in the last stack (Figure 1) for primates, myoglobins and hydrogenases. The DEE for the trees grew slightly at the beginning of the stack, but the first few trees were almost as good. There was a pronounced jump after this nearly constant level. The position of the jump depends on the evolutionary distance used, but correlates with the number of leaves on the tree when the number of leaves is small ($< 30$). For hydrogenases the jump was around $K = 30$. In order to investigate the effect of the $K$ parameter on the "goodness" of trees we also built trees for each dataset using different $K$ values. The DEE value of the best tree (which has the smallest DEE) in the last queue was plotted against the $K$ in Figure 2 for different phylogenetic distances and datasets. As can be seen, the DEE decreased while the limits of the priority queues rose to 60. Moreover there is threshold (about $30 - 40$), after which the DEE of the best trees remains practically constant.

As a rule of thumb these points give us a good estimation of what the parameter setting for $K$ should be. According to this rule, $K$ should be around the number of leaves if it is smaller than 30. For bigger trees $K = 40$ seems to be a good estimate. Applying this rule we built trees with different tree building methods using various distances on the three real-life datasets. The results are summarized in Table 3. It is evident that DEE in most cases is the smallest for our new MS based tree building method. The performance of the UPGMA was not as good as the others when compared with the model populations, but the

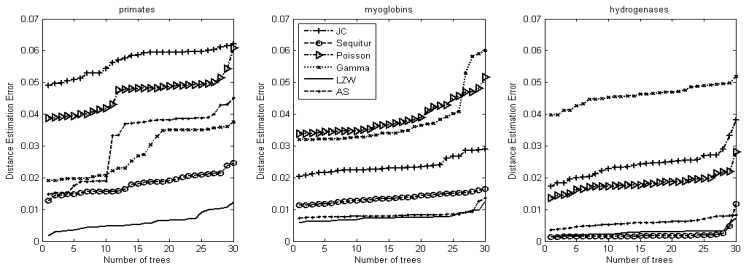


**Fig. 1.** The normalized DEE of the MS trees in the last priority queue ($K = 30$)

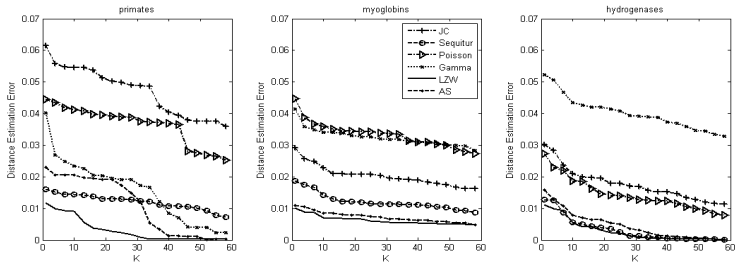


**Fig. 2.** The dependence of the normalized DEE of the best tree in the priority queue on the parameter $K$

**Table 3.** The normalized distance estimation error of different tree building methods using distinct similarity measures on the datasets. The values in bold show the minimal value in each row. Normalized distance estimation errors were multiplied by 1000. The value $K$ for MS was set to 30 for primates and Myoglobins and 40 for hydrogenases. In this table UP means the UPGMA method, and A-S the Alignment-Score.

| | Primates ($N = 12$) | | | | Myoglobins ($N = 27$) | | | | Hydrogenases ($N = 75$) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | UP | NJ | FM | MS | UP | NJ | FM | MS | UP | NJ | FM | MS |
| JC | 96.22 | 60.42 | 49.93 | **45.17** | 88.35 | 62.77 | 62.12 | **19.70** | 40.80 | 11.69 | **10.58** | 15.17 |
| Gamma | 112.87 | 76.36 | 63.08 | **17.62** | 174.56 | 90.69 | 81.43 | **30.80** | 56.44 | 18.29 | **16.97** | 37.48 |
| Poisson | 93.85 | 53.99 | 47.97 | **37.06** | 115.63 | 57.34 | 59.53 | **32.58** | 37.86 | 10.33 | **8.76** | 12.36 |
| LZW | 68.95 | 12.31 | 12.31 | **1.68** | 33.10 | 8.63 | 62.97 | **5.60** | 15.33 | 2.91 | 1.85 | **0.50** |
| Sequ. | 30.49 | 30.49 | 30.49 | **11.89** | 69.20 | 23.10 | 48.76 | **10.99** | 22.86 | 2.13 | 2.14 | **0.52** |
| A-S | 88.25 | 78.32 | 65.59 | **13.71** | 65.66 | 18.21 | 48.05 | **7.03** | 26.71 | 4.07 | 5.32 | **1.27** |

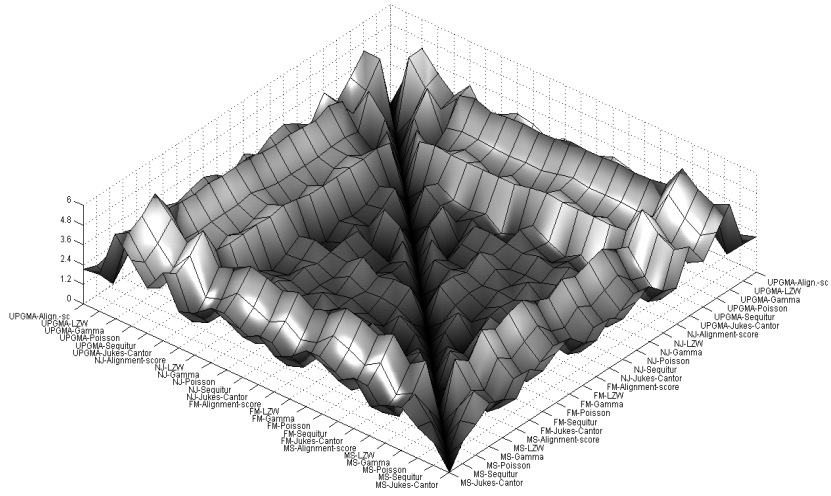

**Fig. 3.** The BSD distance of the trees with the myoglobin dataset

Normalized DEE for the FM and NJ methods are very similar here. Interestingly these two methods (NJ, FM) outperform the MS method in terms of a Normalized DEE when we used alignment-based evolutionary distances. Otherwise the MS method achieved better results. We also compared the methods in terms of their BSD values. In Figure 3 the labels along the axis represent the tree building methods and the evolutionary distance/similarity measure we applied and are separated by a hyphen. Comparing the tree topologies of different trees that employ the BSD, we see that the performance of the MS method is very similar for all datasets (note the plateau in Figure 3). It is also apparent from the evaluations that distance-based methods (UPGMA, NJ and FM ) produce trees with very similar topologies (note the wide valley in the middle of the diagrams). The topology of the trees built by the MS method are different for these trees (notice the higher regions of the diagram in Figure 3). The MS method, however, produced similar topologies regardless of the evolution distance used for tree building, but we can still say that the MS trees brought an improvement in the Normalized DEE for the trees as Table 3 quite clearly indicates.

## 5 Conclusion

In this paper we have presented a novel distance-based and iterative tree building algorithm for analysing the lineage of taxa in structural biology and then compared it with other tree building methods using a new phylogenetic benchmark. Next we showed for simulated, model datasets and for three distinct real-life datasets that it is an efficient tool for building phylogenetic trees. The new method is superior on distance estimation and produces robust trees as the tests on model trees shows. The "goodness" of the resultant trees, however, strongly depends on the parameter $K$. As it follows from the nature of the method, if $K$ is big enough the MS method approximates the exhaustive search. On choosing a proper $K$ value, MS successfully and quickly searches in a previously unexplored region of the possible tree topologies, hence it produces slightly different topologies than those with the algorithms used previously. This allows us to gain a deeper insight into protein and DNA evolution, relationships and lineage, and we hope that the MS method and the phylogenetic benchmarking we introduced here will become widely used tools for tackling phylogenetic problems.

## References

1. Rohlf F. J. (1963) Classification of Aedes by numerical taxonomic methods (Diptera: Culicidae). Ann Entomol Soc Am 56:798804.
2. Saitou N., Nei M. (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol Biol Evol. Jul;4(4):406-25.
3. Atteson K. (1999): The performance of neighbor-joining methods of phylogenetic reconstruction. Algorithmica, 25.
4. Fitch, W. M., and E. Margoliash. (1967): Construction of phylogenetic trees. Science 155:279284.

5. Bryant D. and Waddell P. (1998): Rapid Evaluation of Least-Squares and Minimum-Evolution Criteria on Phylogenetic Trees. Mol. Biol. Evol. 15(10):1346-1359.
6. Cavalli-Sforza, L., and Edwards. A. (1967): Phylogenetic analysis models and estimation procedures. Evolution 32: 550570.
7. Levenberg-Marquardt nonlinear least squares algorithms in C/C++ `http://www.ics.forth.gr/~lourakis/levmar/`
8. Day, W.H.E. (1986): Computational complexity of inferring phylogenies from dissimilarity matrices. Bulletin of Mathematical Biology 49:461-467.
9. Goloboff, P., A. (1999): Analysing large data sets in reasonable times: Solutions for composite optima. Cladistics 15:415-428, 1999.
10. Bahl L.R., Gopalakrishnan P.S. and Mercer R.L. (1993): Search Issues in Large Vocabulary Speech Recognition, Proceedings of the 1993 IEEE Workshop on Automatic Speech Recognition, Snowbird, UT.
11. Robinson, D.F., Foulds, L.R. (1981): Comparison of phylogenetic trees. Math. Biosci. 53, 131–147.
12. Kuhner M. K., Felsenstein J. (1995): A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. Mol Biol Evol May;12(3):525.
13. Gosztolya G., Kocsor A. (2003): Improving the Multi-stack Decoding Algorithm in a Segment-Based Speech Recognizer. IEA/AIE 2003: 744-749
14. Needleman, S. B., Wunsch, C. D. (1970): A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. 48:443-453.
15. Henikoff S., Henikoff JG. (1992): Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci. USA. 15;89(22):10915-9.
16. Jukes T. H., Cantor C. R. (1969): Evolution of protein molecules, pp. 21132 in Mammalian Protein Metabolism, edited by H. N. MUNRO. Academic Press, New York.
17. Zuckerland, E., and L. Pauling, (1965): Molecular disease, evolution, and genetic heterogeneity, pp. 189225 in Horizons in Biochemistry, edited by M. KASHA and B. PULLMAN. Academic Press, New York.
18. Dickerson, R. E. (1971): The structures of cytochrome c and the rates of molecular evolution. J. Mol. Evol. 1:26-45.
19. Cilibrasi R., Vitányi P. (2004): Clustering by compression, IEEE Transactions on Infomation Theory.
20. Ziv J., Lempel A. (1977): A universal algorithm for sequential data compression, IEEE Trans. on Inf. Th. IT-23 337-343.
21. Nevill-Manning C. G., Witten I. H. (1997): Compression and explanation using hierarchical grammars. Computer Journal, 40(2/3):103-116.
22. Ovchinnikov I., et al. (2000). Molecular analysis of Neanderthal DNA from the northern Caucasus, Nature 404(6777):490-493.
23. Vignais P. M., Billoud B., Meyer J. (2001): Classification and phylogeny of hydrogenases. FEMS Microbiology Reviews 25 455-501.
24. Felsenstein, J. (1981): Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. 17:368376.
25. Phylip program package `http://evolution.genetics.washington.edu`