PHYLOGENETIC TREE BUILDING USING A NOVEL COMPRESSION-BASED NON-SYMMETRIC DISSIMILARITY MEASURE

R. BUSA-FEKETE[1] – A. KOCSOR[1]*– CS. 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
(phone: +36-62-524-140; fax: +36-62-425-508)*

**Corresponding author*
e-mail: busarobi.kocsor@inf.u-szeged.hu

*[2]MTA SZBK Institute of Biophysics
H-6701 Szeged, POBox. 521., Hungary
(phone: +36-62-599-605; fax: +36-62-433-133)
e-mail: csaba@nucleus.szbk.u-szeged.hu*

(Received 10th Sep 2005, accepted 10th Oct 2006)

Abstract. An approach of building phylogenetic trees is to define a distance function based on amino acid sequences of distinct proteins. The aim of this approach is to determine a weighted tree topology that approximates the entire functional similarity relations between proteins, defined by a distance function. In this case – according to the definition - the similarity relation has a symmetric property. However, the assumption of symmetry is not always appropriate, because non-symmetric similarity relations between proteins might have a biological significance. This notion inspired us to define a novel, compression-based, non-symmetric dissimilarity measure and to modify the ubiquitous *'Unweighted Pair Group Method with Arithmetic Mean'* (UPGMA)-based tree-building algorithm so that the new measure can be applied.

Keywords: *phylogenetic trees, non-symmetric dissimilarity measures, Unweighted Pair Group Method with Arithmetic Mean, compression–based similarity measure, hydrogenase proteins*

Introduction

Over the past decades researchers have studied proteins in order to learn more about their functionality. Now work has moved on to the systematisation of proteins isolated from distinct species that have similar functionality, and this has become one of key question in phylogenetics. This methodology allows us to examine the proteins by comparing their AA (amino acid) sequences. There are several distance functions available in the literature for calculating distance values between aligned AA sequences. The best-known ones are the Jukes-Cantor distance function [8], and the Kimura 2-parameter model [7]. One drawback of these models, however, is that the AA sequences in question first have to be aligned, which might require an algorithm that makes use of dynamic programming [14], [13].

When examining the proteins we should also ask whether a symmetric similarity measure (e.g. a distance function) describes protein relations in the right way. In the real world a similarity relation is not necessarily symmetric, as one of the proteins being compared might contain more information than the other. That is why we decided to develop a mathematical model that employs a non-symmetric dissimilarity measure (NSDM) in order to learn how dissimilar these proteins are. This model might have interesting biological applications.

The tree-building algorithms can usually handle only a symmetric measure. One of the methods used is the traditional '*Unweighted Pair Group Method with Arithmetic Mean*' (UPGMA), which is a bottom-up algorithm (agglomerative). Here we extend the UPGMA procedure using the symmetrised NSDM for building phylogenetic tree structures, and we use the original asymmetric dissimilarity measure when we calculate the dissimilarity measures of individual nodes and leaves by UPGMA.

The biological example used in this study is the hydrogenase enzyme group. Hydrogenases (H2ases) catalyze the reversible oxidation of molecular hydrogen and play a central role in microbial energy metabolism. Most of these enzymes are found in *Archaea* and *Bacteria*, but a few are present in *Eucarya* as well. They can be placed into three classes: the [Fe]-H2ases, the [NiFe]-H2ases, and the metal-free H2ases. The vast majority of known H2ases belong to the first two classes, and over 100 of these enzymes have been characterized genetically and/or biochemically. 73 sequences of different [NiFe] hydrogenases from various microorganisms were chosen for our study (tree building and classification of this enzyme family into different classes). The sample set of sequences were taken from [15].

In Section 2, we describe in detail our novel non-symmetric dissimilarity measure and its mathematical background. In Section 3 an extension of the UPGMA algorithm is presented, which calculates the distances between the nodes and leaves using NSDM defined in Section 2. In the next section we test our new method and present the results obtained in the case of the [NiFe] hydrogenase enzyme group. Finally, in Section 5, we summarise our findings and draw some conclusions about the practical usefulness of our measure.

A compression-based novel non-symmetric dissimilarity measure

Before introducing our compression-based non-symmetric dissimilarity measure we would like to clarify the relationship between the concept of distance functions, metrics and non-symmetric dissimilarity measures.

Distance function, metric and non-symmetric dissimilarity measure

Let us assume that there are a given set of objects (in our case protein sequences). Then it is necessary to define what we mean by 'similarity' and 'dissimilarity'. Furthermore, we need to express this value in numerical terms. If we already have a formal definition of 'similarity' or 'dissimilarity', then the other definition could be calculated by carrying out a monotone decreasing transformation of the original definition.

In order to define dissimilarity it is customary to apply a distance function, or a metric. The distance function is a function $f(x,y)$ over a given set Ω that has the following properties:

a) the value of the distance function is 0, if and only if the two elements coincide for any $x, y \in \Omega$;
b) the function is called *symmetric* when $d(x,y) = d(y,x)$ for any $x, y \in \Omega$. Otherwise it is called non-symmetric.

If a distance function also satisfies the triangle inequality, it will be a metric.

A metric and a distance function usually have symmetric properties, but we have a feeling that the concept of similarity and dissimilarity does not categorically determine a symmetric 'relation' between objects. Hence it is worth defining the non-symmetric versions of the definitions stated above, which we could obtain by abandoning the symmetric property requirement. Henceforth, instead of talking about a 'non-symmetric distance function' and a 'non-symmetric dissimilarity metric' we shall use a more general term, that of 'non-symmetric dissimilarity measure' (NSDM).

Compression based non-symmetric dissimilarity measure

When solving many problems in bioinformatics we may successfully apply an information theoretical similarity measure. The information theoretical distance functions are based on a comparison of how much information objects contain relative to each other [3]. Here we shall introduce a non-symmetric compression-based version of it. But first of all we need to review the concept of the Universal Turing Machine and Kolmogorov-complexity.

The Universal Turing Machine (UTM) is roughly a mathematical model of present-day computers [1], since it is necessary to input a word x, say, and a program M, say, over the same alphabet, Σ, for it to work. Then the UTM simulates the operation of the Turing Machine coded in the program M with the input word x. (In general it is not a problem if we further assume that the alphabet is binary as we can always have an equivalent subset of positive numbers for any set, where the numbers are represented in binary form.)

The language recognised by UTM is a recursively enumerable set, but is not recursive; hence it is not computable 'algorithmically'. This is an important fact in our approach, because the Kolmogorov-complexity of a given word x over an alphabet Σ means the length of the shortest program, so the UTM has to count the word x with an empty input [2]. In other words, the Kolmogorov-complexity of a given word is also not computable algorithmically, so we need to use approximation methods to determine it.

In another approach, the Kolmogorov-complexity means that we give the shortest representation of a word, from which we can restore the original word without loss of information. That is why it seems to be worth approximating the Kolmogorov-complexity with an efficient compressor method in the following way: given a compressor algorithm, we can calculate the length of the sequence after compression, and then assign this value to the word in question. Let *C(x)* denote the length of the sequence we get after compressing a given word x using a given compressor algorithm. Next, let us define the following non-symmetric dissimilarity measure:

Definition: Let n(x,y) be a non-symmetric dissimilarity measure that associates a non-negative real number with any two sequences x,y over Σ defined by:

$$n(x,y) = \frac{C(xy) - C(x)}{C(xy)}, \qquad \text{(Eq. 1)}$$

where xy is the concatenation of words x and y.

Phylogenetic tree building using a non-symmetric dissimilarity measure

Phylogenetic tree-building methods usually construct a tree with a symmetric distance matrix, whose leaves represent proteins. By distance, we mean that the distance between two proteins in the tree is the sum of the weights of edges that connect the two proteins in a unique way. There are two sorts of phylogenetic tree-building algorithms: the so-called 1-stage methods, which determine the weights of edges during the tree-building process, and the 2-stage methods, whose algorithms first build up a tree, and then they calculate the optimal weighting of edges based on a target function.

The problem of phylogenetic tree-building will still be NP-complete if we consider only a subclass of the original problem set [11] (Maximum Parsimony Principle [10]). So it is necessary to use a heuristic to look for the right tree topology.

In this section we will describe a phylogenetic tree-building method that employs a non-symmetric dissimilarity measure which we defined above. This is a 2-stage method, so it must first look for an appropriate tree topology, and it must then assign optimal weights to edges (between i and j in both directions) of the tree based on the least square procedure.

Least Squares Unweighted Pair Group Method with Arithmetic Mean (LS-UPGMA)

The UPGMA method [9] is a hierarchical agglomerative procedure. It is a bottom-up algorithm, so it first considers all points as subtrees, and then it iteratively joins a pair of subtrees which seem most appropriate for the heuristic used.

It is straightforward to extend the UPGMA method to one which has a non-symmetric dissimilarity measure. The method we developed determines a tree topology using a modified UPGMA, and then it applies a numerical algorithm to optimise the tree weighting. The framework of the procedure can be seen in the following:

1. Algorithm: *Least Square Unweighted Pair Group Method with Arithmetic Mean* (LS-UPGMA):

```
    Input: $D^{nxn}$ (non-symmetric) similarity matrix
    Output: $C^{(2n-1)x(2n-1)}$ incidence matrix, W optimal weights of edges
        C: = MUPGMA(D)      % seeking a tree topology
        W:= LS(C,D)         % determine the optimal edge weighting
```

The *Modified Unweighted Pair Group Method with Arithmetic Mean (MUPGMA)* differs from the original one as it only determines a tree topology. The edge-weighting task is performed by another algorithm. Furthermore, we must not forget the fact that when choosing and joining sub-trees, the dissimilarity measure used is not symmetric. The algorithm needed for this is given below:

2. Algorithm: (MUPGMA):
Input: D^{nxn} *dissimiliarity matrix*
Output: $C^{(2n-1)x(2n-1)}$ *incidence matrix*
Given a D=[d_{ij}*] (non-symmetric) dissimilarity matrix between n elements. After we consider all the elements as different clusters, and carry out the following steps when it has only one cluster left (n-1 times):*

1. *Find the (i,j) indices so that* $\max(d_{ij}, d_{ji})$ *is minimal for any cluster*
2. *Create a new u cluster joining the i and j cluster*

3. Determine the u cluster's distance from other clusters in the following way:

$$d_{k,u} = \frac{n_i d_{k,i} + n_j d_{k,j}}{n_i + n_j} \tag{Eq. 2}$$

$$\textit{and}\ d_{u,k} = \frac{n_i d_{i,k} + n_j d_{j,k}}{n_i + n_j} \tag{Eq. 3}$$

where n_i and n_j denote the number of elements in cluster i and cluster j
4. Delete clusters i and j

After running the MUPGMA method we get a tree-topology, where each leaf represents a protein. We would like to calculate the optimal weighting of edges so that the mean squared error between the distances of leaves is found by the tree and the dissimilarity measure is given by the D matrix above. Actually, we can express this task as a constrained least squares problem, as we shall now see.

Let us consider the tree which is built up using the MUPGMA method. The path-edge incidence matrix P to this tree could be constructed, whose rows correspond to the paths between the leaves, and whose columns correspond to the edges in a fixed order. The element with row index *(i,j)* and column index *k* in matrix P has the value *1*, if the edge *k* can be found on the path between the points *i* and *j;* otherwise it has the value *0.*

Furthermore, let *v* denote the vector whose distances between the leaves are given by the elements of the matrix D, and let y denote the vector that contains the unknown edge weightings, whose order are determined by columns in the matrix P. Next, we need to minimise the following expression for the optimal weights of edges:

$$\begin{gathered} y^* = \arg\min_{y \in \mathbb{R}^{2n-1}} \|Py - v\| \\ s.t. \quad 0 \le y \end{gathered} \tag{Eq. 4}$$

This algorithm can be expressed in a compact way by the following:

```
3. Algorithm: Least Square Method for Optimising the Weights of Edges :
    Input: $D^{nxn}$ dissimilarity matrix, $C^{(2n-1)x(2n-1)}$ incidence matrix
    Output:, W, the optimised weights of edges
        • Find the path-edge incidence matrix P using the incidence matrix C
        • Find the $y^*$ optima of Eq.4
        • Generate the W matrix using $y^*$ and return
```

Based on the above extensions, the LS-UPGMA method is suitable for handling a non-symmetric dissimilarity measure.

Experiments

In the experiments 73 different [NiFe] hydrogenase sequences were used for testing the proposed algorithms, i.e. we built phylogenetic trees using the LS-UPGMA method with a specific non-symmetric dissimilarity measure. But first we will describe the dissimilarity measure used for testing, and then we will apply it to the evaluation domain- After we will discuss the results.

The dissimilarity measure used for testing

In Section 2 we defined the non-symmetric dissimilarity measure concept. It was mentioned that we need to select a compressor algorithm to fully determine an NSDM. In the literature two types of compressor method are known: the statistical (Huffmann algorithm) one and the factorisation one (PPMZ, Lempel-Ziv algorithm [4], [5]). Roughly speaking, a statistical compressor first measures the frequency of each symbol occurring in the input, and in the output it defines shorter codes for higher frequency symbols. Instead of statistical information factorisation, compressors work with connected substrings of the string to be compressed. These methods try to find the substring factorisation in a quasi-optimal way.

Our main aim here is to furnish a measure which can estimate the relative information content among strings. This seems to be obvious if we want to express a string in terms of the substring of another string. This approach is very similar to ours. That is why we chose the Lempel-Ziv algorithm [4], [5] - a factorisation compressor method - to define a non-symmetric dissimilarity measure.

Evaluation domain and tests

The calculated tree topologies are shown in *Fig. 1* and *Fig. 2* for the large subunit of 73 [NiFe] hydrogenases. Since the phylogenetic trees are derived from a symmetric dissimilarity matrix the topology of the left and right handed trees are the same. Only the weights of the paths between the root and the leaves are different. Other differences between the right and left oriented trees will not be discussed here.

The topology of phylogenetic tree is different from the phylogenetic trees obtained by different methods [12], [6] for hydrogenases but, the main characteristics of the [NiFe] hydrogenase family remain the same. We can also distinguish four different hydrogenase groups. The second group (Group-B) contains almost exclusively membrane bound hydrogen uptake hydrogenases (Hup). According to the classification made by Vignais, this corresponds to Group-1 [6]. The main difference is that membrane-bound Hyn hydrogenases do not belong to this group. In our algorithm these hydrogenases form a separate group (Group-A), together with some membrane bound H_2 evolving hydrogenases which were previously assigned to a different group (Group-4). Since the physiological role of these Hyn hydrogenases are not completely elucidated we assign this Group-A as the group of H_2 evolving hydrogenases. Group-C is a mixed group of Group-2 and Group-3 [12]. It contains almost all HoxH hydrogenases as a separate subgroup, some ferredoxin reducing hydrogenases, all Hup hydrogenases from cyanobacteria and also the sensory hydrogenases. The subgrouping of these types of hydrogenases is not well established. The D Group can, however, be divided unequivocally into 3 separate subgroups. All bifunctional thermophylic hydrogenases, most of the ferredoxin reducing hydrogenases and some of the hydrogen evolving hydrogenases form a separate subgroup.

It is also striking that these groups can not just be recognised by the second division from the root as they were classified previously [6], but the distance from the root is uniform within the groups. This distance is different for right and left handed trees, however. Moreover this tendency can be justified within the subgroups as well (see Group-D). These two criteria make our trees more reliable than any other previously described phylogenetic tree.

Conclusions and future work

In this paper we presented a new tree-building methodology that is based on NSDM. Then we defined a compression-based NSDM on which we could build a variant of the modified UPGMA method. The optimal weightings for edges of the tree were obtained using the method of least squares. It enabled us to build a weighted directed phylogenetic tree. We applied this to a biological problem, namely that of protein sequences. We performed tests on data for 73 different [NiFe] hydrogenases. It turned out that our dissimilarity measure could provide relevant information for biologists, which could be of advantage in phylogenetic studies. The notion of a non-symmetric dissimilarity measure will also be examined further to learn more about its advantages and limitations in real applications.

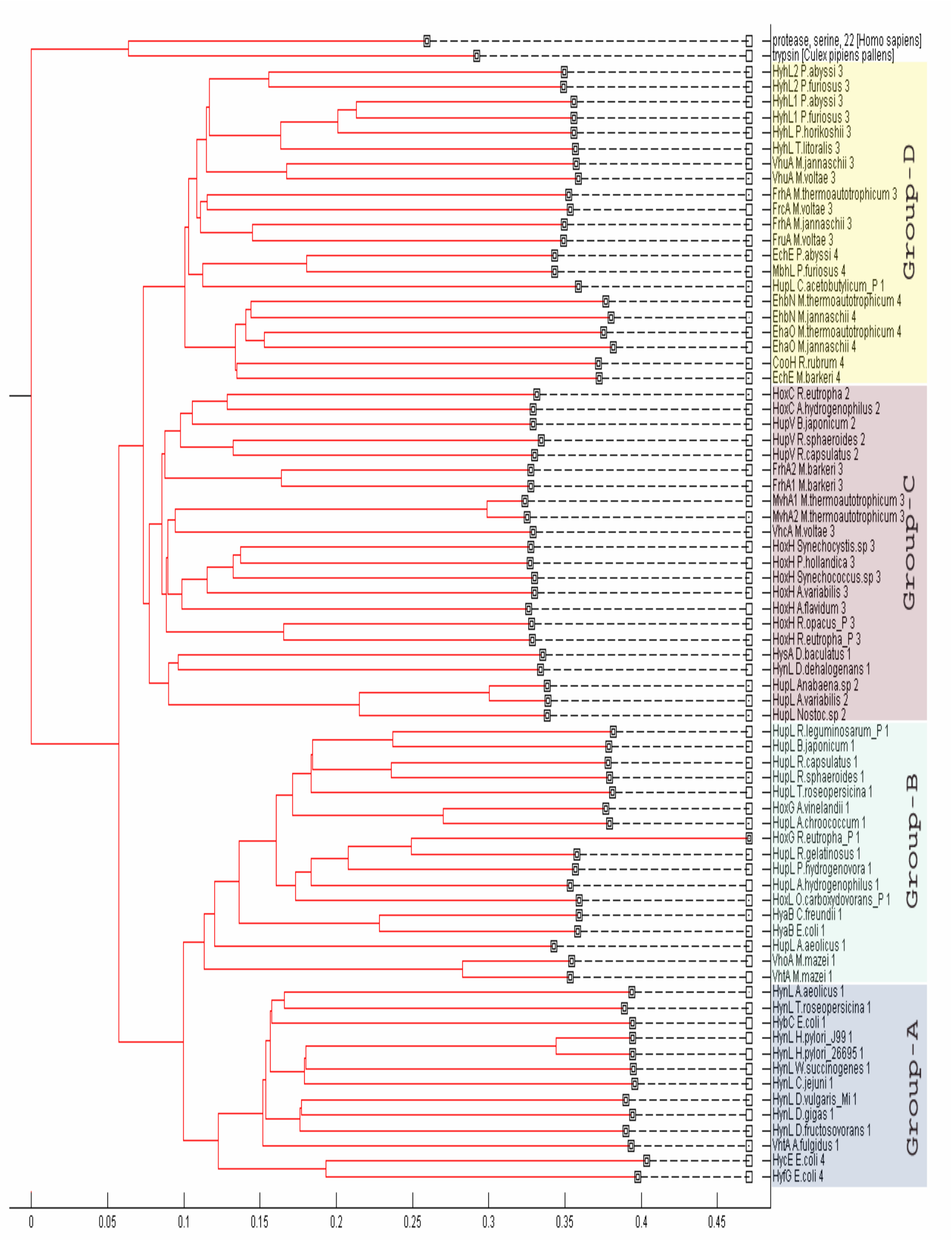

Figure 1. *The left-handed phylogenetic tree of [NiFe]-H2-ases with an outgroup. On the right of the picture there are the names of the hydrogenases with the group number they were previously classified in [9]. The shading indicates our group classification.*

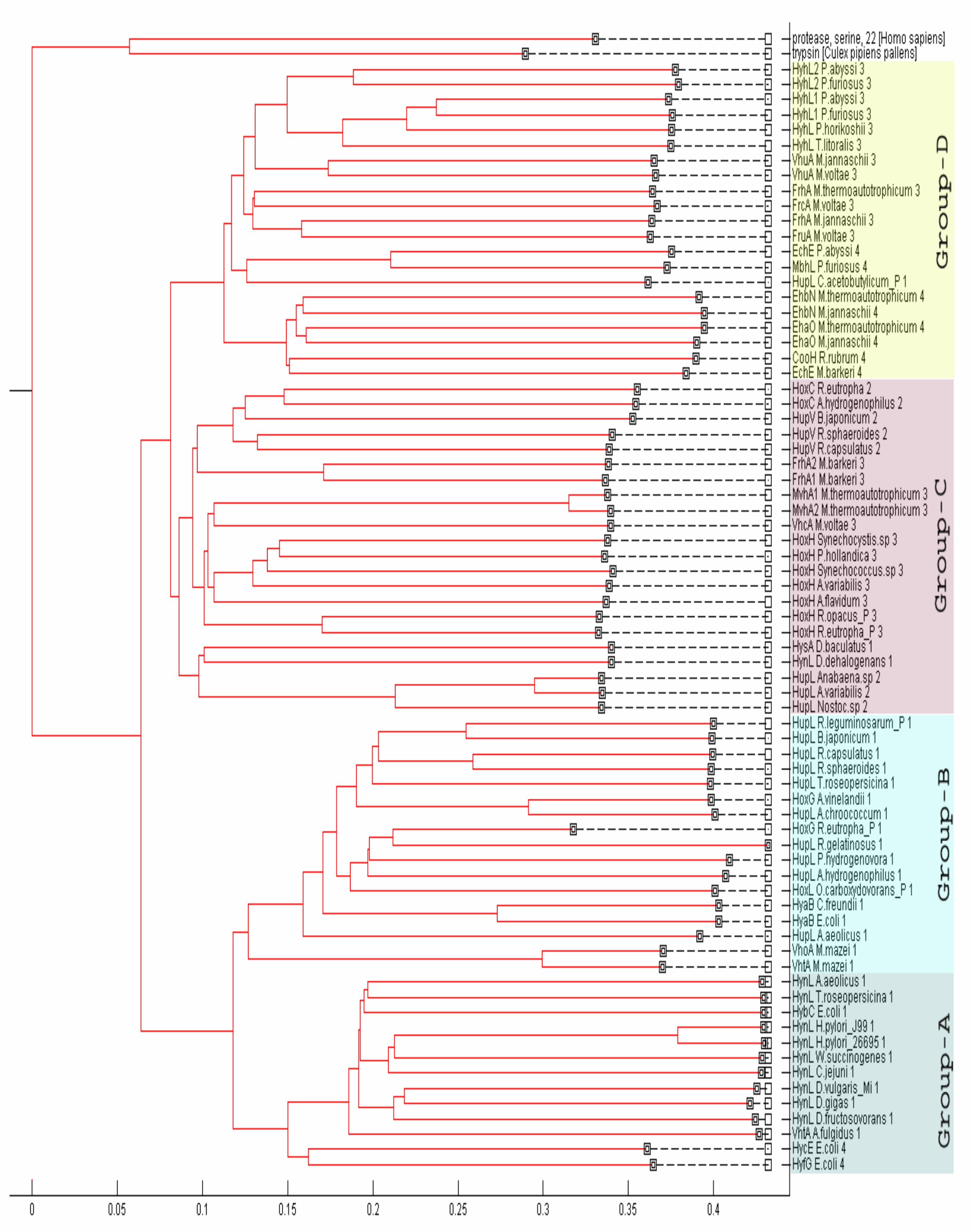

Figure 2. *The right-handed phylogenetic tree of [NiFe]-H2-ases with an outgroup. On the right of the picture there are the names of the hydrogenases with the group number they were previously classified in [9]. The shading indicates our group classification.*

We intend to further examine the algorithm which was presented here to find out whether additional extensions of UPGMA are possible, and also examine other algorithms that use a non-symmetric dissimilarity measure. Another aggregation operator in the first step of MUPGMA could be used as well instead of the current one.

We would also like to build a number of tree topologies at the same time in order to cover a greater part of the solution space.

REFERENCES

[1] Papadimitriou, C.H. (1994): Computational complexity. - Addison-Wesley Publishing Company, Inc.

[2] Li, M., Vitányi, P. (1993): An Introduction to Kolmogorov Complexity and its Applications. – Springer-Verlag, New York.

[3] Cilibrasi, R., Vitanyi, P. (2004): Clustering by compression. - IEEE Transactions on Infomation Theory, See http://arxiv.org/abs/cs.CV/0312044.

[4] Ziv, J., Lempel, A. (1977): A universal algorithm for sequential data compression. - IEEE Trans. on Inf. Th. IT-23 337-343.

[5] Ziv J., Lempel A. (1978): Compression of individual sequences via variable-rate coding. - IEEE Trans. on Inf. Th. IT-24 530-536.

[6] Vignais, P.M., Billoud, B., Meyer, J. (2001): Classification and phylogeny of hydrogenases. - FEMS Microbiology Reviews 25: 455-501

[7] Kimura, M. (1980): A simple method for estimating evolutionary rate of base substitution through comparative studies of nucleotide sequences. - J Mol Evol 16: 111-120.

[8] Jukes, T.H., Cantor, C. (1969): Mammalian Protein Metabolism, chapter Evolution of protein molecules, pages 21-132. - Academic Press, New York.

[9] Michenerand, C.D., Sokal, R.R. (1957): A quantitative approach to a problem in classification. - Evolution, 11:130-162

[10] Cavalli-Sforza, L.L., Edwards, A.W.F. (1967): Phylogenetic analysis: models and estimation procedures. - Evolution 21: 550-570.

[11] Foulds, L.R., Graham, R.L. (1982): The Steiner problem in phylogeny is NP-complete. - Adv. Appl. Math., 3: 43-49.

[12] Wu, L.F., Mandrand, M. A. (1993): Microbial hydrogenases: Primary structure, classification, signatures and phylogeny. - FEMS Microbiol. Rev. 104: 243-270.

[13] Smith, T.F., Waterman, M. S. (1981): Identification of Common Molecular Subsequences. -, J. Mol. Biol. 147: 195-197.

[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] http://wwwabi.snv.jussieu.fr/research/hydrogenases/index.html