



Deriving phylogenetic trees from the similarity analysis of metabolic pathways

Maureen Heymans and Ambuj K. Singh*

Department of Computer Science, University of California, Santa Barbara, CA 93106, USA

Received on January 6, 2003; accepted on February 20, 2003

ABSTRACT

Motivation: Comparative analysis of metabolic pathways in different genomes can give insights into the understanding of evolutionary and organizational relationships among species. This type of analysis allows one to measure the evolution of complete processes (with different functional roles) rather than the individual elements of a conventional analysis. We present a new technique for the phylogenetic analysis of metabolic pathways based on the topology of the underlying graphs. A distance measure between graphs is defined using the similarity between nodes of the graphs and the structural relationship between them. This distance measure is applied to the enzyme-enzyme relational graphs derived from metabolic pathways. Using this approach, pathways and group of pathways of different organisms are compared to each other and the resulting distance matrix is used to obtain a phylogenetic tree.

Results: We apply the method to the Citric Acid Cycle and the Glycolysis pathways of different groups of organisms, as well as to the Carbohydrate metabolic networks. Phylogenetic trees obtained from the experiments were close to existing phylogenies and revealed interesting relationships among organisms.

Availability: Software available upon request from the authors.

Supplementary information: The technical report is available at <http://www.cs.ucsb.edu/~maureen/TR200233.pdf>

Contact: ambuj@cs.ucsb.edu

Keywords: phylogenetic trees, metabolic pathways, graph comparison

INTRODUCTION

Evolutionary and organizational relationships among species have been investigated for several decades. Most of these studies perform a phylogenetic analysis of DNA or protein sequences to study the evolutionary history of organisms from bacteria to humans (Nei, 1996; Olken, 1999; Li *et al.*, 2001). These methods lead to a phyloge-

netic tree in which the nodes represent different species and the edges represent ancestry relationships. Understanding of evolutionary relationships may be further expanded by comparing higher-level functional components among species, such as metabolic pathways. Studies in this direction focusing on individual pathways (Forst and Schulten, 1999, 2001) or on the entire metabolic repertoire (Liao *et al.*, 2002) have been attempted. Such analysis allows us to measure the evolution of complete processes (with different functional roles) rather than the individual elements of conventional phylogenetic analysis.

In this paper, we present a technique for constructing a phylogenetic tree using the structural information inherent in the metabolic pathways of different organisms. To this end, we present a new graph comparison algorithm for computing the evolutionary distance between two pathways. Since evolutionary distance is based on the divergence of the elements constituting the pathways as well as the divergence of the network structure, we combine both these aspects in formulating a measure of the distance between pathways. The former aspect of the distance, i.e. the similarity between two enzymes, can be defined using the sequence similarity of the corresponding genes, or structure similarity of the corresponding proteins, or the similarity between EC (Enzyme Classification) number of the corresponding reactions (NC-IUBMB, 2003). For the latter aspect of the distance (based on network structure), we use an iterative process based on local graph similarity.

Before we apply the graph comparison algorithm, the information in a metabolic pathway or a group of pathways is abstracted into an *enzyme graph*. An enzyme graph removes the information on metabolites and substrates from a pathway and considers only the order of different enzymes present in the pathway. Our graph comparison algorithm is used for a pairwise comparison of the enzyme graphs from different taxa. This yields a distance matrix between the organisms. A phylogenetic tree is constructed from this distance matrix using existing software tools.

We applied our technique to the Glycolysis and Citric Acid Cycle pathways, as well as to the metabolic networks

*To whom correspondence should be addressed.

composed of the Carbohydrate and Lipid metabolic pathways. The clustering of organisms in the resulting phylogenetic trees was consistent with the existing standards. In order to evaluate the quality of our phylogenetic trees, we also used a similarity metric; this metric demonstrated that our approach is superior to competing techniques.

This paper is organized as follows. The first section describes our phylogenetic tree construction algorithm, from extraction of enzyme graphs to graph comparison to tree building. The next section presents the details of the graph comparison algorithm. After that, we describe the experimental results using the Glycolysis pathway, as well as the Carbohydrate and Lipid metabolic networks, on a varying number of organisms. We conclude with a brief discussion in the last section.

OUR ALGORITHM

We divide the process of building phylogenetic trees from metabolic pathways into three steps. In the first step, *enzyme graphs* are constructed for a specific metabolic pathway from a set of organisms under study. In the second step, pairwise comparison of these enzyme-enzyme relational graphs is performed. This yields a distance matrix between organisms. Using this matrix, a phylogenetic tree is computed in the final step with the help of existing software packages. Once a phylogenetic tree has been obtained, we compute its quality by comparing it with existing standards such as trees based on 16SrRNA and NCBI's classification. These steps are detailed next.

Obtaining enzyme graphs from pathways

The collection of reactions and enzymes that an organism uses to achieve a certain metabolic function determines the architecture and topology of the pathway. Metabolic pathways can be abstracted as reaction graphs (networks) with specific graph-topological information, such as connectivity. A metabolic pathway can be represented as a directed reaction graph with substrates as vertices and directed edges denoting reactions (labeled by enzymes) between the vertices.

Given a pathway or a group of pathways, we extract binary relations between enzymes (Goto *et al.*, 1996; Ogata *et al.*, 1996) as follows. Two enzymes are related if they activate reactions which share at least one chemical compound, either as substrate or as product. In the *enzyme graph* $G = (V, E)$ for a given pathway P , the vertex set V consists of the enzymes present in the pathway P and the set of edges E represent the enzyme-enzyme relationships of the pathway. There exists a directed edge from enzyme e_1 to enzyme e_2 in G if e_1 activates some reaction $A \rightarrow B$ (with substrate A and product B) and e_2 activates some reaction $B \rightarrow C$ (with substrate B and product C).

Ogata *et al.* (2000) model metabolic pathways in a similar manner. Each metabolic pathway is treated as

a graph with enzymes (gene products) as vertices and chemical components as edges. Two adjacent vertices representing successive enzymes or reaction steps in the pathway are connected by at least one edge representing a specific chemical compound which is both a substrate of one reaction and a product of the other reaction.

Pairwise comparison of enzyme graphs

Each enzyme graph is specific to a particular organism. A distance matrix between organisms can be computed by performing a pairwise comparison of these graphs. For this, we use a new algorithm that combines similarity between objects represented by the nodes of the graphs and information on the structure of the enzyme graph.

To define a similarity measure between the enzymes of the graph, different notions of relationships between nodes of the graphs (enzymes) can be exploited: sequence similarity of the corresponding genes, structure similarity of the corresponding proteins, or similarity between EC (Enzyme Commission (NC-IUBMB, 2003)) numbers. The EC identifier of an enzyme consists of four digits that categorize the type of the catalyzed chemical reaction. We use a similarity value of 1 if all the four digits of the two reactions are identical, 0.75 if the three first digits are identical, 0.5 if the two first digits are identical, 0.25 if the first digit is identical, and 0 if the first digit is different.

By applying a pairwise comparison to a set of N enzyme graphs, we get an $N \times N$ similarity matrix. The similarity scores ranging from -1 to 1 can be interpreted as distances by using the following formula: $distance = 1 - score$.

Building phylogenetic trees from distance matrices

From the computed distance matrix, we construct a phylogenetic tree with hierarchical clustering algorithms. These methods construct a tree by linking the least distant pair of taxa, followed by successively more distant taxa. There is a wide variety of distance-based clustering algorithms, each based on a different set of assumptions. We use the Phylip (phylogenetic inference) package[†] to construct the phylogenetic trees. This package offers different programs for phylogenetic graph construction. Our trees were constructed using the NJ method (Saitou and Nei, 1987). Trees returned by the this method are unrooted. We reroot these trees using the Midpoint rooting method that chooses the centroid of a tree as the root. The trees are then rendered as graphics using *PhyloDraw*[‡].

Computing the quality of phylogenetic trees

In order to judge the quality of our constructed trees, we need a mechanism to compare the similarity of our trees to existing standards. In this way, we can also compare

[†] <http://evolution.genetics.washington.edu/phylip.html>

[‡] <http://jade.cs.pusan.ac.kr/phylodraw/>

the quality of our trees with those produced by competing techniques. We use a software package called *cousins*[§] to compare the similarity of two trees. This tool performs unordered tree comparison based on cousin distances: a sibling is a cousin of degree 0, a nephew is a cousin of degree 0.5, a first cousin of degree 1, and so on. Two trees are compared based on the set of pairs of each degree.

A NEW ALGORITHM FOR COMPUTING SIMILARITY BETWEEN GRAPHS

The algorithm for computing the similarity between two graphs G_1 and G_2 is divided in four phases. In the first phase, the similarity score between every pair of nodes (a, b) where $a \in G_1$ and $b \in G_2$ is computed by an iterative process. In the second phase, we construct a bipartite graph using the similarity scores, and find a maximal weight matching of this bipartite graph. In the third phase, a similarity measure between every pair of matched nodes is recomputed. Finally, a similarity score between the two graphs is computed by summing the similarity of the matched nodes and by normalizing this sum. We now present the details of each phase.

Obtaining similarity scores between nodes

Let $G_1 = (V_1, E_1, \lambda_1)$, where $|V_1| = n_1$, and $G_2 = (V_2, E_2, \lambda_2)$, where $|V_2| = n_2$, be two directed graphs. G_1 and G_2 are represented by their adjacency matrix A_1 ($n_1 \times n_1$) and A_2 ($n_2 \times n_2$). A $n_1 \times n_2$ similarity matrix S , where the entry $S(a, b)$ expresses the similarity between the node $a \in G_1$ and node $b \in G_2$, is obtained as the limit of a converging iterative process[†]. The similarities between every pair of nodes (a, b) where $a \in G_1$ and $b \in G_2$ are computed simultaneously. We first define a similarity score Sim between every pair of objects represented by the nodes of the two graphs. In the case of the enzyme graphs, the similarity between enzymes can be defined in a number of ways, viz. identity mapping, sequence similarity, or structural similarity. The similarity between every pair of nodes (a, b) of G_1 and G_2 is then defined by combining the notion of similarity between the objects the nodes represent and the similarity of their neighborhood. The basic intuition behind the approach is that two nodes are similar if they reference and are referenced by similar nodes. A similar approach has been recently developed independently by Melnik *et al.* (2002); Jeh and Widom (2002) and Blondel and Van Dooren (2002). However, they do not define a similarity score between graphs, as we do here.

[§] <http://www.cs.nyu.edu/cs/faculty/shasha/papers/cousins.html>

[†] Our proof of convergence in (Heymans and Singh, 2002) requires that the graphs be connected and have additional properties; however, this condition is sufficient but not mandatory, the algorithm converges rapidly even for disconnected graphs.

The similarity scores between nodes, $S(a, b)$, are initialized with $Sim(a, b)$, and then updated simultaneously according to the following mutually recursive rule: two nodes are similar if they link to similar nodes, are referenced by similar nodes, have both missing ingoing (outgoing) edges from (to) similar nodes and have mismatches between edges from (to) dissimilar nodes. The similarity between two nodes (a, b) is computed by summing their similarities and subtracting their dissimilarities. The former consists of four terms, A_1 – A_4 , and the latter consists of four terms, D_1 – D_4 . The first four terms represent the similarity between the presence and absence of edges from and to similar nodes, while the remaining four terms represent the mismatches between these edges. These terms are now discussed in detail. Their mathematical definition can be found in (Heymans and Singh, 2002).

Term $A_1(a, b)$ represents the average similarity between the in-neighbors of a (nodes from which a has incoming edges) and the in-neighbors of b . We first obtain the sum of similarities of the pair of nodes (a_2, b_2) (with $a_2 \in G_1$ and $b_2 \in G_2$) from which a and b have incoming edges. We normalize the sum by dividing it by the total number of in-neighbor pairs, $deg_{in}(a).deg_{in}(b)$ ($deg_{in}(a)$ denotes the number of incoming edges to node a). A slight technicality here is that either a and/or b may not have any in-neighbors. If both a and b have an in-degree of 0, then the term A_1 is defined as the sum of similarities of every pair (a_2, b_2) (with $a_2 \in G_1$ and $b_2 \in G_2$) normalized by the total number of such pairs, $n_1 \times n_2$. If only one of them has an in-degree of 0, then A_1 is set to 0.

Term $A_2(a, b)$ represents the average similarity between the out-neighbors of a (nodes to which a has outgoing edges) and the out-neighbors of b . It is computed over the pair of nodes (a_2, b_2) (with $a_2 \in G_1$ and $b_2 \in G_2$) to which the nodes a and b have outgoing edges. It is defined analogously to A_1 .

The next two terms are motivated by the fact that the absence of edges to similar nodes may be as meaningful as the presence of edges to similar nodes. Term $A_3(a, b)$ is similar to $A_1(a, b)$ except that it works on the complement of the input graphs. It represents the average similarity between the non-in-neighbors of a (nodes from which a has no incoming edges) and the non-in-neighbors of b . We first obtain the sum of similarities of the pair of nodes (a_2, b_2) (with $a_2 \in G_1$ and $b_2 \in G_2$) from which the nodes a and b have no incoming edges. The sum is normalized by dividing by the total number of non-in-neighbor pairs, $(n_1 - deg_{in}(a)).(n_2 - deg_{in}(b))$.

Term $A_4(a, b)$ represents the average similarity between the non-out-neighbors of a (nodes to which a has no outgoing edges) and the non-out-neighbors of b . It is computed over the pair of nodes (a_2, b_2) (with $a_2 \in G_1$ and $b_2 \in G_2$) to which the nodes a and b have no outgoing

edges. It is defined analogously to A_3 .

Term $D_1(a, b)$ represents the dissimilarity between nodes a and b on account of the incoming edges. It is computed over the pair of nodes (a_2, b_2) (with $a_2 \in G_1$ and $b_2 \in G_2$) from which node a has an incoming edges ($a_2 \rightarrow a$) but b does not ($b_2 \not\rightarrow b$).

Term $D_2(a, b)$ is the analogue of $D_1(a, b)$. It considers the similarity of nodes from which a has no incoming edges but b does. Term $D_3(a, b)$ considers the similarity of nodes to which a has an outgoing edge but b does not. Term $D_4(a, b)$ is the analogue of D_3 . It considers the similarity of nodes to which a has no outgoing edges but b does.

The similarity scores $S(a, b)$ are computed by iteration to a fixed point. We initialize the scores $S^0(a, b)$ to $Sim(a, b)$. The scores $S^{(k+1)}(a, b)$ are then recursively computed based on S^k . Since we are only interested in the relative scores, the scores are normalized after each iteration. Here is the outline of the iterative process.

Initialization:

$$S^0(a, b) = Sim(a, b) \quad (1)$$

Iterative step:

$$S^{(k+1)}(a, b) = \left(\frac{A_1^k(a, b) + A_2^k(a, b) + A_3^k(a, b) + A_4^k(a, b)}{4} - \frac{D_1^k(a, b) + D_2^k(a, b) + D_3^k(a, b) + D_4^k(a, b)}{4} \right) \times Sim(a, b) \quad (2)$$

Normalization:

$$S \leftarrow \frac{S}{\|S\|_2} \quad (3)$$

In equation (2), the similarity scores $S(a, b)$ are multiplied by $Sim(a, b)$ in order to combine the neighborhood similarity with the similarity of the objects represented by the nodes. Since each of the four terms A_1 to A_4 , and each of the four terms D_1 to D_4 have a range between -1 and 1 , $S(a, b)$ is also divided by 4 in order to have a range between -1 and 1 . From the equations, we see that the similarity scores are symmetric, i.e. $S(a, b) = S(b, a)$. The convergence of the above iterative process is considered in (Heymans and Singh, 2002).

Bipartite graph matching

At the end of the first phase, we obtain a matrix S that captures the similarity between every pair of nodes of the input graph. The second phase uses these similarities to find the best matching between the graphs. In order to achieve this, we build a bipartite graph and execute a bipartite graph matching algorithm. Given vertex sets V_1 of G_1 and V_2 of G_2 , we construct a bipartite graph $G = (V_1, V_2, S)$ where S is the similarity matrix obtained during the first phase. Once this bipartite graph has been

built, we find the best matching of the graph using an $O((n_1 + n_2)^3)$ Hungarian algorithm (Hopcroft and Karp, 1973; Kuhn, 1955; Baier Saip and Lucchesi, 1993). With the best matching so obtained, we define an $n_1 \times n_2$ boolean matrix M whose entry $M(a, b)$ is set to 1 if nodes a and b have been matched.

Computation of similarity scores between matched nodes

After we find the best correspondence between graphs G_1 and G_2 , we need to obtain the similarity score for this correspondence. As in the first phase, we combine the structural similarity with the node similarity to compute this score. We perform one iteration of a system of equations similar to A_1 – A_4 and D_1 – D_4 . The new set of equations A'_1 – A'_4 and D'_1 – D'_4 is similar to the previous (unprimed) one except that we use $M(a, b)$ instead of $Sim(a, b)$. We also use a new normalization that is square root of the previous one. This is necessary since the maximum size of a matching is the smaller of the input graph sizes; specifically, if a graph is compared to itself then $M(a, b)$ is given by the identity mapping: the similarity terms A'_1 – A'_4 reduce to 1 and the dissimilarity terms D'_1 – D'_4 reduce to 0.

Terms A'_1 – A'_4 and D'_1 – D'_4 incorporate the similarity and the dissimilarity of the best match between graphs G_1 and G_2 . We combine these terms and multiply by the similarity of the nodes to obtain the final value of $S(a, b)$.

Computing graph similarity score

Finally, to obtain the similarity score S_{G_1, G_2} between the graphs G_1 and G_2 , we sum the similarity scores computed in the previous phase over the pair of matched nodes, and normalize the sum by the square root of the product of the number of nodes of G_1 and G_2 , in order to have a similarity score between -1 and 1 . When $G_1 = G_2$, the similarity score will be equal to 1.

$$S_{G_1, G_2} = \frac{\sum_{a \in G_1, b \in G_2, M(a, b)=1} S(a, b)}{\sqrt{n_1 \cdot n_2}} \quad (4)$$

An example

We illustrate our graph matching algorithm with the help of the Citric Acid Cycle pathway for the organisms *Escherichia coli* and *Mus Musculus*. The enzyme graphs of the organisms have been shown in Figure 1. The enzyme graph for *E.Coli* contains 14 enzymes and the one for *Mus Musculus* contains 9 enzymes. After performing the first two phases of our technique (iterative computation of scores and bipartite matching), the resulting matching between enzymes of the two graphs results in Table 1. The values in the second column are the similarity measures obtained in the third phase. Finally, the total similarity

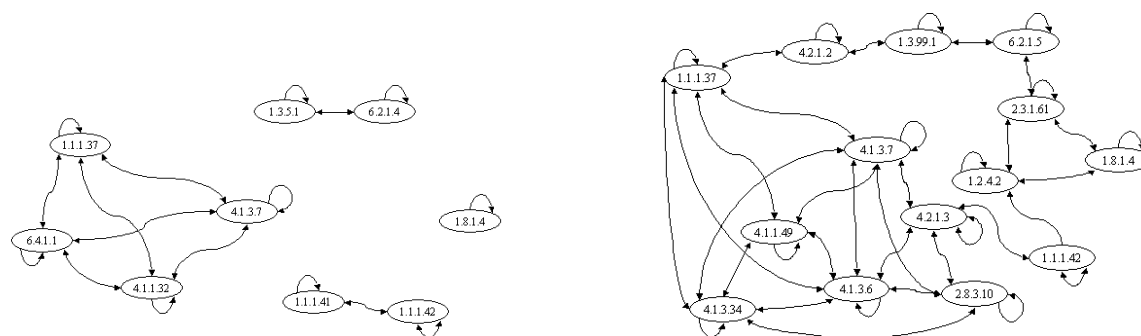


Fig. 1. Enzyme-enzyme relational graph of Citrate cycle (TCA cycle) pathway for *Mus Musculus* (a) and *Escherichia coli* (b).

Table 1. Matching between enzymes of *E.coli* and *Mus Musculus*

Enzyme correspondences <i>E.coli</i> ↔ <i>Mus Musculus</i>	Similarity measures
1.1.1.37 ↔ 1.1.1.37	0.7
1.1.1.42 ↔ 1.1.1.42	0.75
1.2.4.2 ↔ 1.1.1.41	0.14
1.3.9.1 ↔ 1.3.5.1	0.38
1.8.1.4 ↔ 1.8.1.4	0.51
4.1.1.49 ↔ 4.1.1.32	0.53
4.1.3.7 ↔ 4.1.3.7	0.71
6.2.1.5 ↔ 6.2.1.4	0.56

score between these two graphs is computed in the fourth phase to be 0.38.

Time complexity

Let us analyze the time complexity for computing S_{G_1, G_2} . The first phase has a time complexity of $O(Kn_1^2n_2^2)$, where K is the number of iterations. The second phase has a time complexity of $O((n_1 + n_2)^3)$ (graph matching). The third phase is $O(\max(n_1, n_2)n_1n_2)$, and the last step has an $O(1)$ cost. The total complexity is therefore $O(Kn_1^2n_2^2 + (n_1 + n_2)^3)$. Typically, the different equations converge pretty fast ($K \simeq 20$ depending on the size of the graphs), leading to a cubic time complexity in the size of the input graphs.

EXPERIMENTAL RESULTS

Currently Kanehisa (1999); Ogata *et al.* (1999) contains metabolic pathways information for 97 organisms: 10 from eukaryotes, 71 from bacteria and 16 from archaea. We studied 80 of these organisms in our experiments. An overview of these organisms is shown in (Heymans and Singh, 2002). We constructed phylogenetic trees for four different sets of these organisms. A first set of 72 organisms was selected by removing all the organisms

from the 97 Kegg's organisms which have less than three enzymes present in the Glycolysis and Citric Acid Cycle pathways. A second set of 48 organisms was selected by collapsing all organisms with exactly the same network in the Glycolysis and/or Citric Acid Cycle pathways. The third set of 16 organisms is the set of organisms considered by (Liao *et al.*, 2002). The fourth set is composed of eight organisms, two of them are from the eukaryota domain, two other ones are from the archaea domain, and the remaining four are from the bacteria domain. For this set of eight organisms, phylogenetic trees were derived by considering a set of pathways instead of a single pathway.

We evaluated the effectiveness of our technique by comparing the produced phylogenies with the NCBI taxonomy[†] (or the 16S rRNA based tree), and obtaining a single *similarity measure*. Comparative evaluation of our methods was carried out by examining a few other existing techniques, comparing their trees again with the NCBI taxonomy to obtain their similarity measures, and comparing the measures with those produced by our technique. Among similar approaches, Liao *et al.* (2002) have developed a computational method to compare organisms based on whole metabolic pathway analysis. The presence and absence of metabolic pathways in organisms is profiled as a boolean vector. Using this methodology and some specific distance measures on these profiles, pairwise comparisons of a set of completed genomes are performed, and phylogenetic trees are constructed using hierarchical clustering. The results provide a perspective on the relationship among organisms that is different from conventional phylogenetic trees based on 16s rRNA. Forst and Schulten (2001, 1999) also extend conventional phylogenetic analysis of individual elements in different organisms to the organisms' metabolic networks. They outline a method that combines sequence information of enzymes with information of the underlying networks. A global distance between pathways is defined using

[†] <http://www.ncbi.nlm.nih.gov/Taxonomy/>

distances between substrates and distances between corresponding enzymes. The analysis is applied to a variety of networks yielding a comprehensive understanding of similarities and differences between organisms. Ettinger (2002) relates the problem of comparing stoichiometric structure of two reaction systems to the graph isomorphism problem. Results for the Glycolysis pathway and the Carbohydrate metabolism are shown in the following subsections. Results for the Citric Acid Cycle pathway and for the Carbohydrate and Lipid metabolisms can be found in (Heymans and Singh, 2002).

Phylogenetic trees based on the Glycolysis pathway

Glycolysis was one of the first metabolic pathways studied and is one of the best understood, in terms of the enzymes involved, their mechanisms of action, and the regulation of the pathway to meet the needs of the organism and the cell. The Glycolysis pathway is extremely ancient in evolution, and is common to essentially all living organisms. It is found in essentially all free living forms of organisms, conserved well in the genetic code, and the only set of processes to occur in the Cytosol.

Phylogenetic trees for the datasets of 72 and 48 organisms

Figure 2 depicts the phylogenetic tree computed for the sets of 72 organisms. The tree for the set of 48 organisms can be found in (Heymans and Singh, 2002). With few exceptions, going from 72 to 48 organisms did not affect the relative position of the different organisms on the distance trees generated by our approach. This indicates the robustness of our technique. In both trees, organisms within a same genus are closely clustered together. They typically have similar or even exactly the same pathway, and get high similarity values. *Chlamydia* CPN, CPJ, CPA, CTR and CMU are grouped together, proteobacteria beta subdivision NME and NMA are grouped together, and so are *E. Coli* ECS, ECO, ECJ and ECE.

In both the trees (for 72 and 48 organisms), we find separate clusters corresponding to the three domains of life. In Figure 2, we find two clusters of archaea organisms: one cluster with AFU, STO, PAB, PFU, MJA, and MTH (with the methanobacterium MTH and the methanococcus MJA which forms a subcluster), and another cluster with TAC, TVO, SSO, PAI, APE, HAL. The eukaryota are grouped in two clusters: one is composed of the mammals RNO, HSA, and MMU, and another one of the remaining eukaryota DME, SCE, CEL, SPO, and ATH. For the bacteria, a few clusters represent the different subdivision of the proteobacteria. One cluster appears with the proteobacteria XCC, BME, CCR, SME, ATU, and ATC. All the bacteria from the alpha subdivision are present in this cluster, except MLO which is in a upper cluster. Another cluster appears with the proteobacteria gamma subdivision VCH, STY, STM, ECS, ECO, ECJ, ECE, and YPE. And another

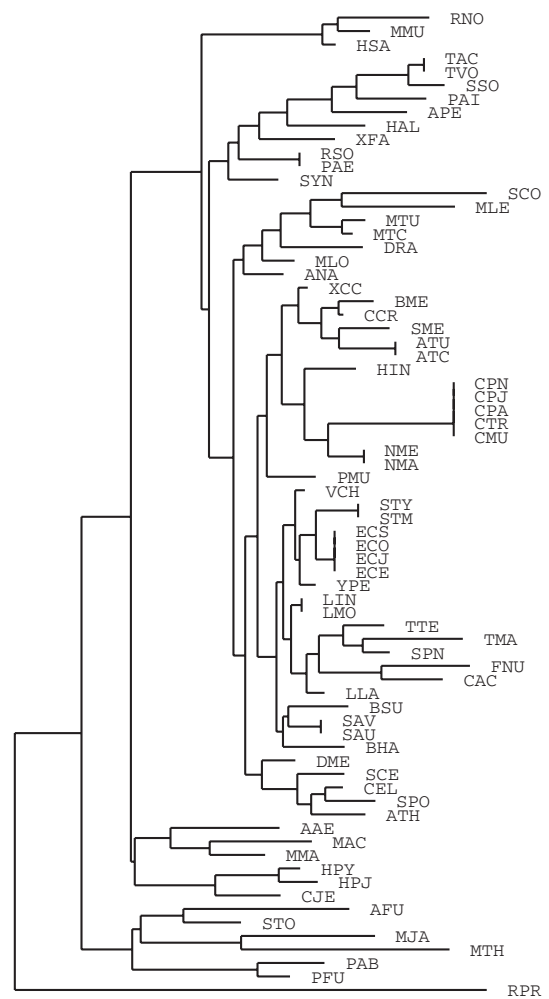


Fig. 2. Phylogenetic trees built using the Glycolysis pathway for 72 organisms.

cluster is composed of the proteobacteria delta subdivision HPY, HPJ, and CJE. The firmicutes are divided in the two groups bacillus and actinobacteria. The firmicutes bacillus LIN, LMO, TTE, SPN, CAC, and LLA form one of these clusters, and the firmicutes actinobacteria SCO, MLE, MTU, and MTC form the other one. We computed similarity measures based on the NCBI taxonomy using the *cousins* tool (Zhang *et al.*, 1996). We also obtained phylogenetic trees for the NCE method (Number of Common Enzymes) in which the phylogenetic analysis is based on the number of common enzymes between two organisms. We do not compare our phylogenies with the 16s rRNA based trees for the set of 48 and 72 organisms since the 16s rRNA sequences are still unpublished for some of the organisms. Table 2 shows the similarity measures of our technique and the NCE technique obtained by using the NCBI taxonomy as the standard. Our method outperforms the NCE technique.

Table 2. Similarity measures based on the NCBI taxonomy for the glycolysis pathway

Technique	72 organisms	48 organisms
Our technique	0.19	0.18
NCE technique	0.14	0.16

Table 3. Similarity measures based on the NCBI taxonomy for the glycolysis pathway for the set of 16 organisms

Technique	Similarity
Our technique	0.26
NCE technique	0.19
16S rRNA	0.22
Liao <i>et al.</i> 's technique	0.16

Phylogenetic trees for the dataset of 16 organisms Figure 3 depicts the phylogenetic tree computed for the set of 16 organisms. The two mycoplasma MGE and MGN have a really low distance of 0.05 and are clustered together. They are the two closest organisms. The two archaea AFU and MJA are also grouped together. The similarity measures using the NCBI taxonomy as the standard are shown in Table 3 for our technique and three others: NCE, 16S rRNA, and Liao *et al.*. Our method outperforms the other techniques. Table 4 shows the similarity measures when the 16S rRNA tree is chosen as the standard. Our method against obtains the best alignment.

Phylogenetic trees based on carbohydrate metabolism

In our last set of experiments, we considered two larger groups of pathways: the first was the group of carbohydrate metabolic pathways, and the second was the group of carbohydrate and lipid metabolic pathways. Carbohydrate metabolism is composed of the glycolysis, citrate cycle (TCA cycle), pentose phosphate, pentose, and glucuronate interconversions pathways, and fructose and mannose, galactose, ascorbate and aldarate, pyruvate, glyoxylate and dicarboxylate, propanoate, butanoate,

Table 4. Similarity measures based on the 16S rRNA tree for the glycolysis pathway for the set of 16 organisms

Technique	Similarity
Our technique	0.27
NCE technique	0.18
Liao <i>et al.</i> 's technique	0.12

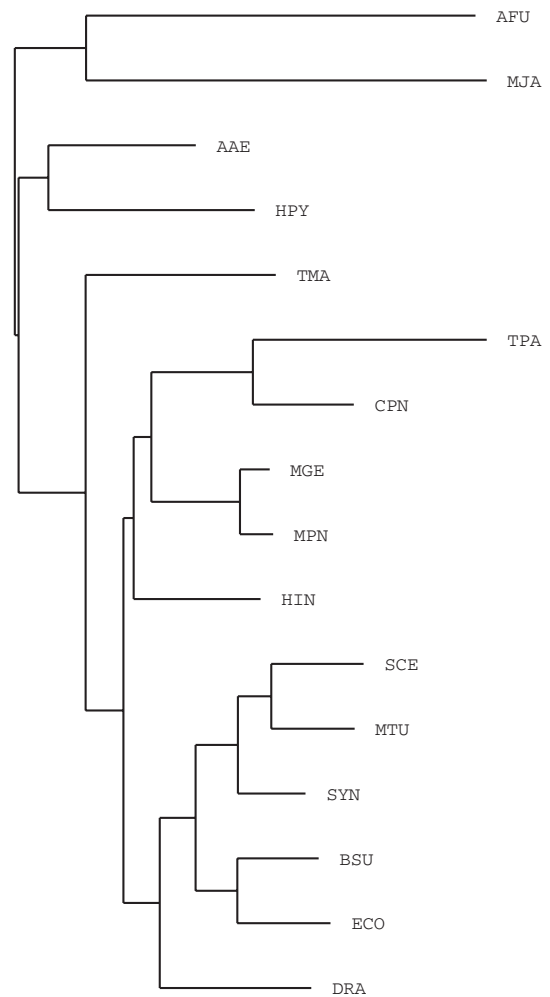


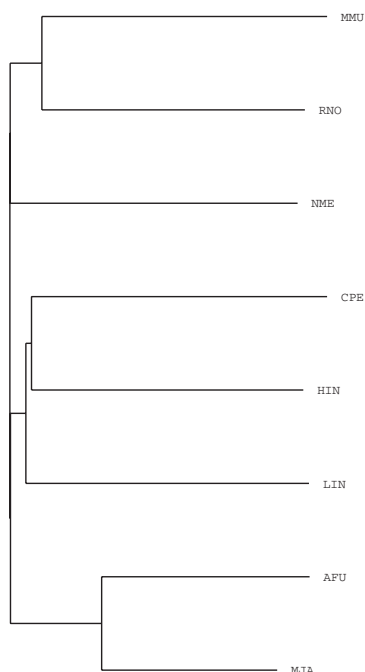
Fig. 3. Phylogenetic tree for 16 organisms built from comparison of Glycolysis pathway.

C5-Branched dibasic acid metabolisms. Carbohydrates serve as the primary source of energy in the cell, and carbohydrate metabolism is central to all metabolic processes.

The 8 organisms that we considered for this experiment are shown in Table 5. We also indicate the number of enzymes present for each organism for the group of pathways. Figure 4 depicts the phylogenetic trees computed for these organisms using the group of carbohydrate metabolisms. The results for the carbohydrate and lipid metabolisms are found in (Heymans and Singh, 2002). The two archaea AFU and MJA are the two closest organisms with a distance of 0.55. They form a separate cluster in the phylogenetic trees. The two eukaryota RNO and MMU are also grouped together with a distance of 0.78. RNO is the closest organism to MMU. In the tree, the bacteria CPE, HIN and LIN are clustered together.

Table 5. Set of eight organisms (N is the number of enzymes in carbohydrate metabolism)

Code	Organism	Domain	N
RNO	<i>Rattus norvegicus</i>	Eukaryota	62
MMU	<i>Mus musculus</i>	Eukaryota	65
AFU	<i>Archaeoglobus fulgidus</i>	Archaea	45
MJA	<i>Methanococcus jannaschii</i>	Archaea	32
NME	<i>Neisseria meningitidis</i>	Proteobacteria	60
HIN	<i>Haemophilus influenzae</i>	Proteobacteria	75
LIN	<i>Listeria innocua</i>	Bacteria Firmicute	77
BSU	<i>Bacillus subtilis</i>	Bacteria Firmicute	113

**Fig. 4.** Phylogenetic trees for the dataset of 8 organisms built from comparison of carbohydrate metabolism.

The proteobacteria NME has a lower distance to the archaea AFU and MJA. The three of them belong to the Prokaryote classification. We constructed phylogenetic trees for these organisms using our technique.

Finally, we evaluated our technique and the NCE technique for the group of pathways. We measured the correspondences of the generated phylogenies to the NCBI taxonomy. The similarity measures are shown in Table 6. As a point of comparison, we also show the similarity measures for trees constructed using our method for the glycolysis and the citric acid cycle pathway. It is evident that studying a group of metabolic pathways instead of a single pathway helps in the construction of better phylogenies.

Table 6. Similarity measures based on the NCBI taxonomy for the dataset of 8 organisms

Technique	Similarity
Our technique for Carbohydrate metabolism	0.93
NCE technique for Carbohydrate metabolism	0.71
Our technique for Glycolysis pathway	0.75
Our technique for Citric Acid Cycle	0.57

CONCLUSION

We proposed a technique for constructing phylogenetic trees using the structural information inherent in the metabolic pathways of different organisms. To this end, we presented a new graph comparison algorithm for computing the evolutionary distance between two pathways. Since evolutionary distance is based on the divergence of the elements constituting the pathways as well as the divergence of the network structure, we combine both these aspects in formulating a measure of the distance between pathways. The originality of our graph theoretical approach relies in the combination of topological similarities with the similarities of the enzymes present in the networks. This approach is useful for understanding higher order functions encoded in the network of interacting enzymes. The effectiveness of our method was demonstrated by applying it to a number of pathways: Glycolysis, Citric Acid Cycle, Carbohydrate and Lipid metabolisms. The clustering of organisms in the resulting phylogenetic trees was consistent with the NCBI taxonomy at numerous levels. In order to compare the quality of our phylogenetic trees, we used a similarity metric that compared our trees with existing standards.

A large number of metabolic pathway databases are currently available, viz. Kegg[†], EcoCyc[‡], and WIT[§]. The main difference between these databases is in the way a pathway is built for different organisms. In Kegg, pathways are consensus views not specific to a particular organism. For each consensus pathway view, enzymes thought to exist in a particular organism can be highlighted. In WIT, consensus views exist, but pathway collections are organized by species. In EcoCyc, each database is specific to a particular organism. It has the advantage of being experimentally verified. It would be very interesting to perform experiments on different databases and compare the computed phylogenies. We chose Kegg in our experiments because it had a large number of organisms.

Our experiments so far have considered only the reaction types of the enzymes in labeling the nodes and in defining node similarity. Further refinements of this

[†] <http://www.genome.ad.jp/kegg/>

[‡] <http://www.ecocyc.org/>

[§] <http://wit.mcs.anl.gov/WIT2/>

general approach may lead to more accurate representation of pathways and distance computation. For example, one can label the nodes with enzymes, and consider the sequence/structure distance between the corresponding proteins in defining distance measures. However, multi-functional enzymes may pose a problem and their different functionalities will need to be considered separately. Another approach towards refining the graph representation may be to include substrate information along with the reaction types to distinguish between enzymes that have the same EC number. Distance between substrates could be defined using their chemical formulae. Extending our approach to other kinds of pathways where labels such as EC numbers do not provide information on the kind of interaction will be challenging.

While constructing an enzyme graph, two enzymes are linked only if the substrate of one is the product of the other. This approach fails to handle cases when the substrate or the product are at entry or exit points of a pathway. In such cases the enzymes will have no link between them. The construction of the graph should be modified in order to include these relationships.

We have used the cousin distance as the criteria for the validation of our phylogenetic trees. There are some other metrics (Stockham *et al.*, 2002) that could also be used. This is planned for the future.

We considered some small groups of pathways in deriving the phylogenetic trees. It should be possible to extend this analysis by considering the whole metabolic network. This type of analysis will use more information on the functional roles and relationships of the enzymes present in the metabolic network.

REFERENCES

- Blondel, V. and Van Dooren, P. (2002) A measure of similarity between graph vertices. With applications to synonym extraction and web searching. *Technical Report UCL 02-50*. Universite Catholique de Louvain, Belgium.
- Ettinger, M. (2002) The complexity of comparing reaction systems. *Bioinformatics*, **18**(3), 465–469.
- Forst, C.V. and Schulten, K. (1999) Evolution of metabolisms: a new method for the comparison of metabolic pathways. in *Proceedings of the Third Annual International Conference on Computational Molecular Biology (RECOMB 1999)*. ACM Press, pp. 174–181.
- Forst, C.V. and Schulten, K. (2001) Phylogenetic analysis of metabolic pathways. *J. Mol. Evol.*, **52**, 471–489.
- Goto, S., Bono, H., Ogata, H., Fujibuchi, W., Nishioka, T., Sato, K. and Kanehisa, M. (1996) Organizing and computing metabolic pathway data in terms of binary relations. *Pac. Symp. Biocomput.*, 175–186.
- Heymans, M. and Singh, A. (2002) Building phylogenetic trees from similarity analysis of metabolic pathways. *Technical Report 2002-33*. Department of Computer Science, University of California, Santa Barbara, December 2002, <http://www.cs.ucsb.edu/~maureen/TR200233.pdf>
- Hopcroft, J.E. and Karp, R.M. (December 1973) An algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, **2**(4), 225–231.
- Jeh, G. and Widom, J. (2002) SimRank: a measure of structural-context similarity. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Edmonton, Alberta, Canada.
- Kanehisa, M. (1999) KEGG: from genes to biochemical pathways. In Letovsky, S. (ed.), *Bioinformatics: Databases and Systems*. Kluwer Academic Publishers, pp. 63–76.
- Kuhn, H.W. (1955) The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, **2**(1), 83–97.
- Li, M., Badger, J.H., Chen, X., Kwong, S., Kearney, P. and Zhang, H. (2001) An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, **17**, 149–154.
- Liao, L., Kim, S. and Tomb, J.F. (2002) Genome comparisons based on profiles of metabolic pathways. In *Sixth International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES 2002)*. Crema, Italy.
- Melnik, S., Garcia-Molina, H. and Rahm, E. (February 2002) Similarity flooding: a versatile graph matching algorithm. In *Proceedings of Eighteenth International Conference on Data Engineering*. San Jose, California.
- Nei, M. (1996) Phylogenetic analysis in molecular evolutionary genetics. *Annu. Rev. Genet.*, **30**, 371–403.
- Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB) (2003) Enzyme Nomenclature Recommendations. <http://www.chem.qmul.ac.uk/iubmb/enzyme/>.
- Ogata, H., Bono, H., Fujibuchi, W., Goto, S. and Kanehisa, M. (1996) Analysis of binary relations and hierarchies of enzymes in the metabolic pathways. *Genome Inform.*, **7**, 128–136.
- Ogata, H., Fujibuchi, W., Goto, S. and Kanehisa, M. (2000) A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. *Nucleic Acids Res.*, **28**, 4021–4028.
- Ogata, H., Sato, K., Goto, S., Fujibuchi, W., Bono, H. and Kanehisa, M. (1999) KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, **27**, 29–34.
- Olken, F. (1999) Phylogenetic Tree Computation Tutorial. <http://pga.lbl.gov/Workshop/April2002/lectures/Olken.pdf>.
- Baier Saip, H.A. and Lucchesi, C.L. (1993) Matching algorithms for bipartite graph. *Technical Report DCC-03/93*. Departamento de Cincia da Computao, Universidade Estadual de Campinas.
- Saitou, N. and Nei, M. (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, **4**(4), 406–425.
- Stockham, C., Wang, L. and Warnow, T. (2002) Statistically based postprocessing of phylogenetic analysis by clustering. *Bioinformatics*, **18**(3), 465–469.
- Zhang, K., Wang, J.T.L. and Shasha, D. (1996) On the editing distance between undirected acyclic graphs. *Int. J. Foundations Comput. Sci.*, **7**(1), 43–57.