Deriving phylogenetic trees from the similarity analysis of metabolic pathways

Maureen Heymans Ambuj K. Singh
Department of Computer Science
University of California, Santa Barbara, CA 93106
{maureen,ambuj}@cs.ucsb.edu

December 28, 2002

Abstract

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

1 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 [33, 39]. These methods lead to a phylogenetic 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. In such pathways, enzymes, substrates, and reactions are grouped conceptually into networks as part of a dynamic information processing system. A metabolic pathways is a series of individual chemical reactions in a living system that combine to perform one or more important functions (viz., glycolysis and Kreb's cycle). Comparative analysis of metabolic pathways in different genomes yields important information on their evolution. Studies in this direction focusing on individual pathways [15, 16, 17] or on the entire metabolic repertoire [30] 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 [34]. For the latter aspect of the distance (based on network structure), we use a heuristic based on random walks [19, 28].

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 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. Section 2 summarizes the traditional phylogenetic tree construction algorithms and existing techniques for comparative analysis of metabolic pathways. Section 3 describes our phylogenetic tree construction algorithm, from extraction of enzyme graphs to graph comparison to tree building. Section 4 presents the details of the graph comparison algorithm. Section 5 describes the experimental results using the Glycolysis and Krebs Citric Acid Cycle pathways, as well as the Carbohydrate and Lipid metabolic networks, on a varying number of organisms. We conclude with a brief discussion in Section 6. A proof of convergence of our graph comparison technique appears in an Appendix.

2 Related Work

To study the evolutionary relationships between organisms, various methods can be employed to estimate when the species may have diverged from a common ancestor. Having this information allows one to construct a phylogenetic tree in which species are arranged on branches that link them according to their evolutionary descent. The most popular and frequently used methods of tree building can be classified into two major categories [33, 39]: phenetic methods based on distances and cladistic methods based on characters. The former measures the pairwise distance/dissimilarity between two organisms and constructs the tree totally from the resultant distance matrix. In the latter, trees are calculated by considering the various possible evolutions and are based on parsimony or likelihood methods. The resulting tree is the one that optimizes the evolution.

2.1 Phylogenetic trees based on DNA and RNA sequences

Construction of phylogenetic trees for a group of taxa requires data for each of the taxa that contain information about their evolutionary history. Historically, morphological data was used for inferring phylogenies. However, the abundance of DNA/RNA sequence data currently available for a variety of organisms has led to phylogenetic inference based on these data. Most of the phylogeny algorithms rely on multi-sequence alignments [13] of cautiously selected characteristic sequences: sequences of a single protein or single gene from each organism. Numerous studies have used the Ribosomal RNA 16S sequences, because these sequences exist in all organisms and are highly conserved [12, 31].

However, in spite of the success of rRNA microbial taxonomy, the evolutionary relationships between major groups of organisms are still unclear because phylogenetic analysis of single gene sequences lacks the

information to resolve deep branches in the tree. Further, misalignment and differing evolutionary rates can result in phylogenetic trees with the wrong topology. The recently completed sequences of several organism genomes provide an enormous amount of data with which to address some of these problems. Phylogenetic analysis can also be performed using the whole genome [14, 29], leading to more precise studies. However, only a limited number of genome sequences are available to date, and hence the results of such techniques may not be representative of the whole picture.

2.2 Phylogenetic trees based on metabolic pathways

Metabolism is defined as the set of complex physical and chemical processes involved in the maintenance of life. It is comprised of a vast repertoire of enzymatic reactions and transport processes used to convert thousands of organic compounds into the various molecules necessary to support cellular life. The metabolism of each organism is subdivided in different metabolic pathways. Each metabolic pathway is a series of individual chemical reactions in a living system that combine to perform one or more important functions. The product of one reaction in a pathway serves as the substrate for the following reaction. Recently a few methods that use organisms' metabolic pathways to compute a phylogenetic tree have been proposed [15, 16, 17, 30, 47]. Comparative analysis of the pathways can shed important information on evolution, and allows the analysis of complete processes rather than the individual elements typical of a conventional phylogenetic analysis.

Liao et al. [30] 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. Based on this methodology and using 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 [15, 16, 17] 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 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. Our work is different in that we use the structure of the network to compute distances, not merely the presence/absence of enzymes.

Tohsato et al. [47] present a method for the comparative analysis of genomes and metabolic pathways based on similarity between gene orders and enzymatic reactions. To measure the reaction similarity, they formalize a scoring system using the functional hierarchy of the EC numbers of enzymes. They use a dynamic programming based technique to align two or more pathways. The similarity score between pathways is expressed as the information content of their alignment. They apply their algorithm to the metabolic pathways in Escherichia coli. Unfortunately, their algorithm does not consider branching pathways that occur in some metabolic pathways, it can only be applied to a line graph.

2.3 Pathway databases

A large number of metabolic pathway databases are currently available, viz., Kegg, EcoCyc, and WIT. The Kyoto Encyclopedia of Genes and Genomes (KEGG) [1, 24, 37, 38] server is a repository of metabolic pathways for organisms with completely sequenced genomes. KEGG provides information on molecular and cellular biology in terms of the information pathways that consist of interacting molecules or genes. It also provides links from the gene catalogs produced by genome sequencing projects. The KEGG database provides metabolic pathway maps and regulatory pathways maps, which can be viewed in terms of a specific

organism. It provides enzyme data with links to pathways, genes, diseases (OMIM database), motif and PDB (Protein Data Bank) structures. We use the Kegg database in our study to obtain metabolic pathways of different organisms as well as information about the enzymes present in the pathways and their classification.

EcoCyc [2, 25, 26] was originally a database for metabolic pathways in Escherichia Coli. It has been extended to other microbial organisms to produce the database MetaCyc. The database is based on published experimental data and, unlike KEGG, also includes information about genes that have not yet been sequenced but whose function has been characterized by genetic and biochemical approaches.

WIT (What is There) [3] is another database which provides information on gene and operon organization, as well as information about metabolic networks for completely or partially sequenced genomes. Currently 53 genomes of microbial origin and one of a multicellular organism are accessible via the WIT system. Of these genomes, 42 have been completely sequenced and the remaining are subjects of ongoing sequencing projects.

The main difference between the above 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.

2.4 Graph comparison techniques

The problem of computing similarity between graphs has been studied in various domains: structural pattern recognition, computer vision, schema comparison in databases, etc. Sanfeliu and Fu [44] group distance measures on graphs into two categories:

- Feature-based distances: A set of features is extracted from each graph. These features are combined into a vector on which existing distance measures (such as Euclidean) can be defined [10].
- Cost-based distances: The distance between two graphs measures the number of modifications [8] required in order to transform the first graph to the second graph. A set of edit operations, such as deletion, insertion, and substitution of nodes and edges are defined, and the similarity of two graphs is computed as the least cost sequence of edit operations that transforms one graph into the other.

As an example of the first category of distances, Papadopoulos and Manolopoulos [40] construct a feature vector for a graph by calculating the degree of each vertex in the graph. Due to errors and distortions in the input data and the graph models, approximate or error-tolerant graph matching methods are needed in many applications. The second category of distance measures deals with this problem well. Shasha et al. [49] define an edit distance between Connected Undirected Acyclic graphs with Labelled nodes being (CUAL). Since computing this distance is NP-complete, they propose a constrained distance metric, called the degree-2 distance, by requiring that any node to be inserted (deleted) have no more than 2 neighbors. With this metric, they present an algorithm to solve the problem.

In addition to these two categories, graph matching [11] and subgraph isomorphism [48, 41] are frequently used to compare graphs. Bunke and Shearer [9] defined a graph distance metric based on the maximal common subgraph. The main problem with subgraph isomorphism is that it is an NP-complete problem[1].

Several approaches have been developed recently for computing structural similarities. Melnik et al. [32] find the best matching between two graphs using an iterative fixpoint computation. Their technique considers

[1]Matching labeled graphs is not a hard problem when the matching between labels is one to one. But when that is not the case and a distance measure is defined on the labels, the problem again becomes difficult.

the nodes of two graphs to be similar when their adjacent nodes are similar. Unlike our approach, they only consider similarities of the "out-neighbors" and "in-neighbors"; our technique incorporates the similarities of missing ingoing and outgoing edges too in the computation of the nodes similarities. They also do not define a distance measure between the two graphs, concentrating instead on finding the optimal mapping between nodes. In a similar vein, Jeh and Widom [23] define a similarity measure between the nodes of a given graph. The intuition behind their algorithm is that "two objects are similar if they are related to similar objects". Similarity measures between two nodes are defined by the similarities of the nodes which refer to these nodes. This approach defines similarity between nodes of a graph and not between graphs, as we propose in this paper. Blondel and Van Dooren [7] define a concept of similarity between vertices of directed graphs. The similarity scores between two nodes are defined by the similarities of the ancestors and descendants of the nodes. They show that the similarity scores are given by the components of a dominant vector of a non-negative matrix and propose an iterative method to compute them.

3 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.

3.1 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 (labelled by enzymes) between the vertices.

Given a pathway or a group of pathways, we extract binary relations between enzymes [35, 18] 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. [36] 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.

3.2 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. The algorithm is detailed in Section 4.

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 [34]) numbers. The EC identifier of an enzyme consists of four digits that categorize the type of the catalyzed chemical reaction. In the experimental results presented later, we use this information. 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: $distance = 1 - score$. Two identical graphs will have a distance of 0. An $N \times N$ distance matrix is obtained in this manner.

3.3 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. The simplest of these is UPGMA (Unweighted Pair Group Method using Arithmetic averages) [45]. In this technique, the two closest pair of organisms are merged into a new node and the distances to the merged node are recomputed via the mean of the pairwise distances to the leaves. By repeating this process, we obtain the needed dendrogram. Another very popular distance method is the NJ (Neighbor Joining) Method [43]. This method attempts to correct the UPGMA method for its (frequently invalid) assumption that the same rate of evolution applies to each branch. NJ starts by correcting the distance matrix for overall divergence from other taxa. The least distant pairs of nodes (most closely related taxa) are connected, their common ancestral node is added to the tree, and the terminal nodes are pruned from the tree. This continues until only two nodes remain and all taxa are connected. This method yields an unrooted tree.

We use the Phylip (phylogenetic inference) package [4] to construct the phylogenetic trees. This package offers different programs for phylogenetic graph construction. Our trees were constructed using the NJ method. The 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* [5].

3.4 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 the quality of our trees with those produced by competing techniques. We use a software package called *cousins* [6] 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.

4 A new algorithm for computing similarity between graphs

We begin with some basic definitions.

Definition 4.1 *A directed graph G is a 3-tuple $G = (V, E, \lambda)$, where V is the set of vertices (nodes), $E \subseteq V \times V$ is a set of edges, and $\lambda : V \to L_V$ is a function assigning labels to the vertices. If $V = \emptyset$, then G is the empty graph. If for every edge $(a, b) \in E$, there exists the edge $(b, a) \in E$, then the graph is said to be undirected. A graph is weighted if a cost function $c : E \to \Re$ has been defined.*

Definition 4.2 *A graph can be represented by its adjacency matrix. If the graph G has n vertices, then the adjacency matrix is an $n \times n$ matrix A defined by*

$$A(i,j) = \begin{cases} 1 & \text{if } i \to j \text{ in } G \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

If a graph G is undirected, then its adjacency matrix is symmetric.

Definition 4.3 *A matching between two graphs $G_1 = (V_1, E_1, \lambda_1)$ with $|V_1| = n_1$ and $G_2 = (V_2, E_2, \lambda_2)$ with $|V_2| = n_2$ consists of a one-to-one mapping M which associates vertices of G_1 to vertices of G_2. Generally, the mapping is expressed as the set of ordered pairs (a, b) (with $a \in G_1$ and $b \in G_2$). The mapping can be expressed by an $n_1 \times n_2$ mapping matrix M where the entry $M(a, b)$ with $a \in G_1$ and $b \in G_2$ is defined as:*

$$M(a,b) = \begin{cases} 1 & \text{if } a \text{ and } b \text{ are matched} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The matching problem between two graphs can be solved using the well-known matching problem in bipartite graphs [20, 27, 42]. Such a bipartite graph is formed by pairing up the nodes from one graph with those of the other graph. The weight on the edges of the bipartite graph is based on the similarity of the two pairs of nodes. First, we define a bipartite graph and the notion of matching on bipartite graphs.

Definition 4.4 *A bipartite graph $G = (V_1, V_2, E)$ ($E \subseteq V_1 \times V_2$) is a graph whose vertex set V can be partitioned into two subsets V_1 and V_2 in such a way that every edge of G joins a vertex in V_1 to a vertex in V_2. No two vertices within the same set are adjacent.*

Definition 4.5 *A matching in a bipartite graph $G = (V_1, V_2, E)$ is a set of edges $E' \subseteq E$ in which with no two edges share a common vertex. The weight of the matching is the sum of the weights of the edges. A maximum weight matching is one with the maximum weight over all matchings.*

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 scores between every pair of nodes (a, b) where $a \in G_1$ and $b \in G_2$ are 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.

4.1 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[2]. 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.

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.

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. Here is the mathematical definition:

$$A_1^{(k)}(a,b) = \begin{cases} \sum_{a_2 \to a, b_2 \to b} \frac{S^k(a_2,b_2)}{deg_{in}(a) deg_{in}(b)} & \text{if } deg_{in}(a) \neq 0 \text{ and } deg_{in}(b) \neq 0 \\ \sum_{a_2 \in G_1, b_2 \in G_2} \frac{S^k(a_2,b_2)}{n_1 \times n_2} & \text{if } deg_{in}(a) = deg_{in}(b) = 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $a_2 \in G_1, b_2 \in G_2$. Notation $a_2 \to a$ means there exists an edge from a_2 to a.

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:

$$A_2^{(k)}(a,b) = \begin{cases} \sum_{a \to a_2, b \to b_2} \frac{S^k(a_2,b_2)}{deg_{out}(a) deg_{out}(b)} & \text{if } deg_{out}(a) \neq 0 \text{ and } deg_{out}(b) \neq 0 \\ \sum_{a_2 \in G_1, b_2 \in G_2} \frac{S^k(a_2,b_2)}{n_1 \times n_2} & \text{if } deg_{out}(a) = deg_{out}(b) = 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $deg_{out}(a)$ is the number of outgoing edges for a.

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, $(n_1 - deg_{in}(a)).(n_2 - deg_{in}(b))$. The mathematical definition is as follows:

[2]Our proof of convergence in the Appendix requires that the graphs be connected; however, the algorithm converges rapidly even for disconnected graphs.

$$A_3^{(k)}(a,b) = \begin{cases} \sum_{a_2 \not\to a, b_2 \not\to b} \frac{S^k(a_2,b_2)}{(n_1 - deg_{in}(a)).(n_2 - deg_{in}(b))} & \text{if } deg_{in}(a) \neq n_1 \text{ and } deg_{in}(b) \neq n_2 \\ \sum_{a_2 \in G_1, b_2 \in G_2} \frac{S^k(a_2,b_2)}{n_1 \times n_2} & \text{if } (n_1 - deg_{in}(a)) = (n_2 - deg_{in}(b)) = 0 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $a_2 \not\to a$ means there is no edge from a_2 to a.

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:

$$A_4^{(k)}(a,b) = \begin{cases} \sum_{a \not\to a_2, b \not\to b_2} \frac{S^k(a_2,b_2)}{(n_1 - deg_{out}(a)).(n_2 - deg_{out}(b))} & \text{if } deg_{out}(a) \neq n_1 \text{ and } deg_{out}(b) \neq n_2 \\ \sum_{a_2 \in G_1, b_2 \in G_2} \frac{S^k(a_2,b_2)}{n_1 \times n_2} & \text{if } (n_1 - deg_{out}(a)) = (n_2 - deg_{out}(b)) = 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

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 ($a_2 \to a$) but b does not ($b_2 \not\to b$):

$$D_1^{(k)}(a,b) = \begin{cases} \sum_{a_2 \to a, b_2 \not\to b} \frac{S^k(a_2,b_2)}{(deg_{in}(a)).(n_2 - deg_{in}(b))} & \text{if } deg_{in}(a) \neq 0 \text{ and } deg_{in}(b) \neq n_2 \\ \sum_{a_2 \in G_1, b_2 \in G_2} \frac{S^k(a_2,b_2)}{n_1 \times n_2} & \text{if } (deg_{in}(a)) = (n_2 - deg_{in}(b)) = 0 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

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:

$$D_2^{(k)}(a,b) = \begin{cases} \sum_{a_2 \not\to a, b_2 \to b} \frac{S^k(a_2,b_2)}{(n_1 - deg_{in}(a)).(deg_{in}(b))} & \text{if } deg_{in}(a) \neq n_1 \text{ and } deg_{in}(b) \neq 0 \\ \sum_{a_2 \in G_1, b_2 \in G_2} \frac{S^k(a_2,b_2)}{n_1 \times n_2} & \text{if } (n_1 - deg_{in}(a)) = (deg_{in}(b)) = 0 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Term $D_3(a,b)$ considers the similarity of nodes to which a has an outgoing edge but b does not:

$$D_3^{(k)}(a,b) = \begin{cases} \sum_{a \to a_2, b \not\to b_2} \frac{S^k(a_2,b_2)}{(deg_{out}(a)).(n_2 - deg_{out}(b))} & \text{if } deg_{out}(a) \neq 0 \text{ and } deg_{out}(b) \neq n_2 \\ \sum_{a_2 \in G_1, b_2 \in G_2} \frac{S^k(a_2,b_2)}{n_1 \times n_2} & \text{if } (deg_{out}(a)) = (n_2 - deg_{out}(b)) = 0 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

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:

$$D_4^{(k)}(a,b) = \begin{cases} \sum_{a \not\to a_2, b \to b_2} \frac{S^k(a_2,b_2)}{(n_1 - deg_{out}(a)).(deg_{out}(b))} & \text{if } deg_{out}(a) \neq n_1 \text{ and } deg_{out}(b) \neq 0 \\ \sum_{a_2 \in G_1, b_2 \in G_2} \frac{S^k(a_2,b_2)}{n_1 \times n_2} & \text{if } (n_1 - deg_{out}(a)) = (deg_{out}(b)) = 0 \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

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:

$$S^0(a,b) = Sim(a,b) \tag{11}$$

Iterative step:

$$S^{(k+1)}(a,b) = \frac{A_1^k(a,b) + A_2^k(a,b) + A_3^k(a,b) + A_4^k(a,b) - D_1^k(a,b) - D_2^k(a,b) - D_3^k(a,b) - D_4^k(a,b)}{4} \times Sim(a,b) \tag{12}$$

Normalization:

$$S \leftarrow \frac{S}{\parallel S \parallel_2} \tag{13}$$

In equation (12), 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 the Appendix.

4.2 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 [20, 27, 42]. 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.

4.3 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 mappings, the similarity terms A_1'–A_4' reduce to 1 and the dissimilarity terms D_1'–D_4' reduce to 0.

$$A_1'(a,b) = \begin{cases} \sum_{a_2 \to a, b_2 \to b} \frac{M(a_2,b_2)}{\sqrt{deg_{in}(a) deg_{in}(b)}} & \text{if } deg_{in}(a) \neq 0 \text{ and } deg_{in}(b) \neq 0 \\ \sum_{a_2 \in G_1, b_2 \in G_2} \frac{M(a_2,b_2)}{\sqrt{n_1 \times n_2}} & \text{if } deg_{in}(a) = deg_{in}(b) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

$$A_2'(a,b) = \begin{cases} \sum_{a \to a_2, b \to b_2} \frac{M(a_2,b_2)}{\sqrt{deg_{out}(a) deg_{out}(b)}} & \text{if } deg_{out}(a) \neq 0 \text{ and } deg_{out}(b) \neq 0 \\ \sum_{a_2 \in G_1, b_2 \in G_2} \frac{M(a_2,b_2)}{\sqrt{n_1 \times n_2}} & \text{if } deg_{out}(a) = deg_{out}(b) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

$$A'_3(a,b) = \begin{cases} \sum_{a_2 \not\to a, b_2 \not\to b} \frac{M(a_2,b_2)}{\sqrt{(n_1 - deg_{in}(a)).(n_2 - deg_{in}(b))}} & \text{if } deg_{in}(a) \neq n_1 \text{ and } deg_{in}(b) \neq n_2 \\ \sum_{a_2 \in G_1, b_2 \in G_2} \frac{M(a_2,b_2)}{\sqrt{n_1 \times n_2}} & \text{if } (n_1 - deg_{in}(a)) = (n_2 - deg_{in}(b)) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

$$A'_4(a,b) = \begin{cases} \sum_{a \not\to a_2, b \not\to b_2} \frac{M(a_2,b_2)}{\sqrt{(n_1 - deg_{out}(a)).(n_2 - deg_{out}(b))}} & \text{if } deg_{out}(a) \neq n_1 \text{ and } deg_{out}(b) \neq n_2 \\ \sum_{a_2 \in G_1, b_2 \in G_2} \frac{M(a_2,b_2)}{\sqrt{n_1 \times n_2}} & \text{if } (n_1 - deg_{out}(a)) = (n_2 - deg_{out}(b)) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

$$D'_1(a,b) = \begin{cases} \sum_{a_2 \to a, b_2 \not\to b} \frac{M(a_2,b_2)}{\sqrt{(deg_{in}(a)).(n_2 - deg_{in}(b))}} & \text{if } deg_{in}(a) \neq 0 \text{ and } deg_{in}(b) \neq n_2 \\ \sum_{a_2 \in G_1, b_2 \in G_2} \frac{M(a_2,b_2)}{\sqrt{n_1 \times n_2}} & \text{if } (deg_{in}(a)) = (n_2 - deg_{in}(b)) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

$$D'_2(a,b) = \begin{cases} \sum_{a_2 \not\to a, b_2 \to b} \frac{M(a_2,b_2)}{\sqrt{(n_1 - deg_{in}(a)).(deg_{in}(b))}} & \text{if } deg_{in}(a) \neq n_1 \text{ and } deg_{in}(b) \neq 0 \\ \sum_{a_2 \in G_1, b_2 \in G_2} \frac{M(a_2,b_2)}{\sqrt{n_1 \times n_2}} & \text{if } (n_1 - deg_{in}(a)) = (deg_{in}(b)) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

$$D'_3(a,b) = \begin{cases} \sum_{a \to a_2, b \not\to b_2} \frac{M(a_2,b_2)}{\sqrt{(deg_{out}(a)).(n_2 - deg_{out}(b))}} & \text{if } deg_{out}(a) \neq 0 \text{ and } deg_{out}(b) \neq n_2 \\ \sum_{a_2 \in G_1, b_2 \in G_2} \frac{M(a_2,b_2)}{\sqrt{n_1 \times n_2}} & \text{if } (deg_{out}(a)) = (n_2 - deg_{out}(b)) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

$$D'_4(a,b) = \begin{cases} \sum_{a \not\to a_2, b \to b_2} \frac{M(a_2,b_2)}{\sqrt{(n_1 - deg_{out}(a)).(deg_{out}(b))}} & \text{if } deg_{out}(a) \neq n_1 \text{ and } deg_{out}(b) \neq 0 \\ \sum_{a_2 \in G_1, b_2 \in G_2} \frac{M(a_2,b_2)}{\sqrt{n_1 \times n_2}} & \text{if } (n_1 - deg_{out}(a)) = (deg_{out}(b)) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

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 $S(a,b)$:

$$S(a,b) = \frac{A'_1(a,b) + A'_2(a,b) + A'_3(a,b) + A'_4(a,b) - D'_1(a,b) - D'_2(a,b) - D'_3(a,b) - D'_4(a,b)}{4} \times Sim(a,b) \tag{22}$$

4.4 Computing graph similarity score

Finally, to get 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.

$$S_{G_1,G_2} = \frac{\sum_{a \in G_1, b \in G_2, M(a,b)=1} S(a,b)}{\sqrt{n_1.n_2}} \tag{23}$$

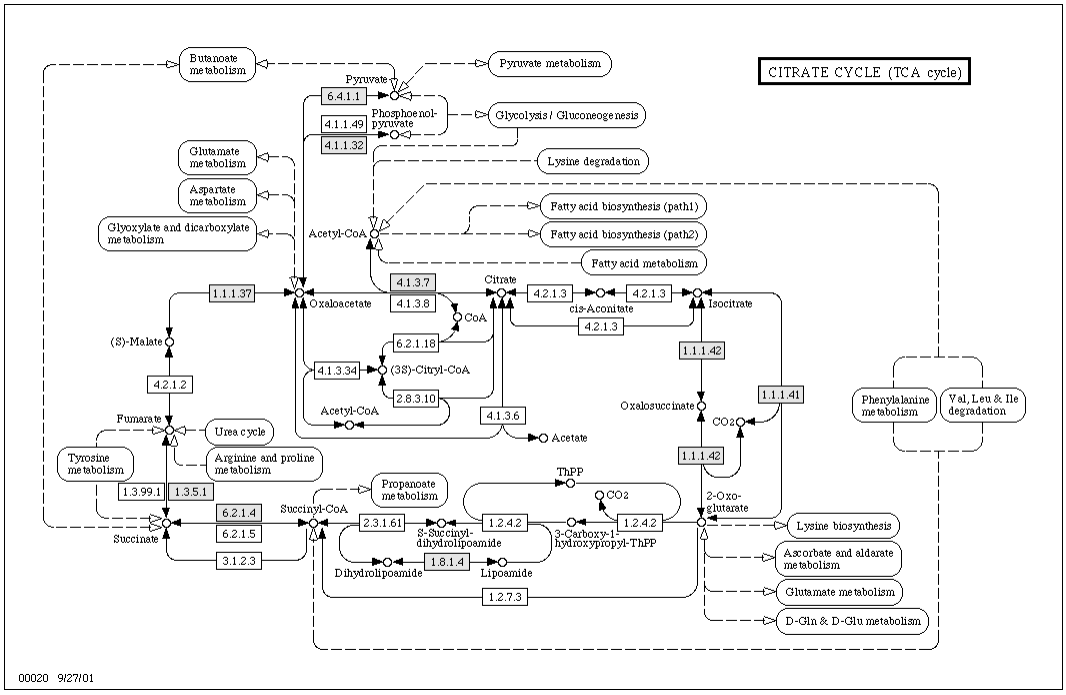


Figure 1: Citrate cycle (TCA cycle) pathway. The enzymes are shown in boxes with the EC numbers inside. The shaded represent those enzymes whose genes are identified in *Mus Musculus.*

4.5 An example

We illustrate our graph matching algorithm with the help of the Citric Acid Cycle pathways for the organisms Escherichia coli and Mus Musculus. The metabolic pathways for these organisms are shown in Figures 1 and 2. The corresponding enzyme graphs of the organisms have been shown in Figure 3. 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 real numbers between parentheses are the similarity measures obtained in the third phase. Finally, the total similarity score between these two graphs is computed in the fourth phase to be 0.38.

E. coli $\leftrightarrow$ Mus Musculus	
1.1.1.37 $\leftrightarrow$ 1.1.1.37	(0.7)
1.1.1.42 $\leftrightarrow$ 1.1.1.42	(0.75)
1.2.4.2 $\leftrightarrow$ 1.1.1.41	(0.14)
1.3.99.1 $\leftrightarrow$ 1.3.5.1	(0.38)
1.8.1.4 $\leftrightarrow$ 1.8.1.4	(0.51)
4.1.1.49 $\leftrightarrow$ 4.1.1.32	(0.53)
4.1.3.7 $\leftrightarrow$ 4.1.3.7	(0.71)
6.2.1.5 $\leftrightarrow$ 6.2.1.4	(0.56)

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

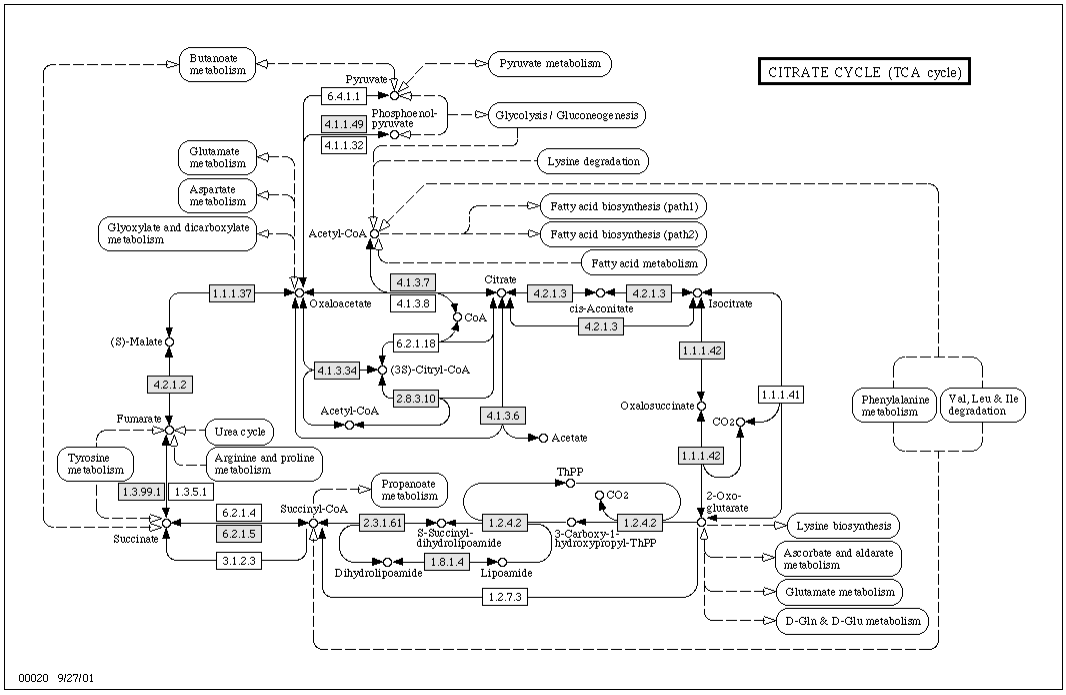
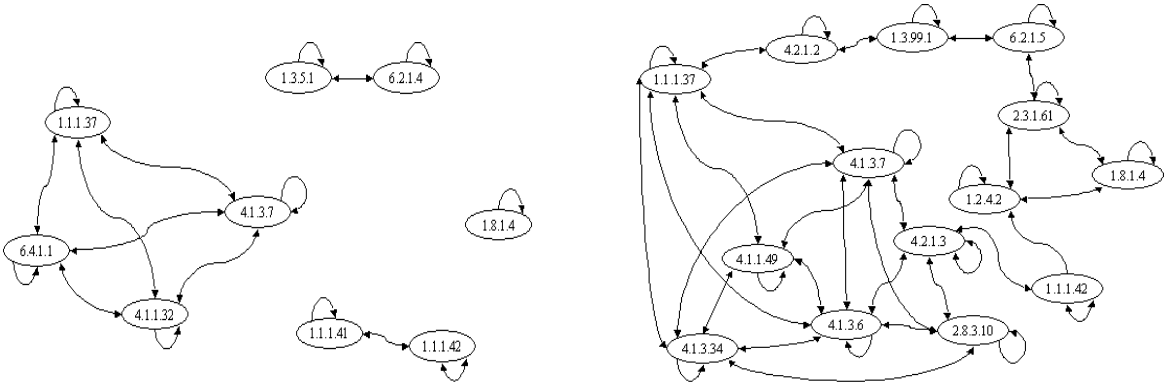


Figure 2: Citrate cycle (TCA cycle) pathway. The enzymes are shown in boxes with the EC numbers inside. The shaded represent those enzymes whose genes are identified in *Escherichia coli*.



(a) *Mus Musculus* (b) *Escherichia coli*

Figure 3: Enzyme-enzyme relational graph of Citrate cycle (TCA cycle) pathway for *Mus Musculus* (a) and *Escherichia coli* (b).

4.6 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 $O(Kn_1^2.n_2^2+(n_1+n_2)^3)$. 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.

5 Experimental results

Currently Kegg [1, 24, 37, 38] contains metabolic pathways information for 97 organisms, 10 from eukaryotes, 71 from bacteria and 16 from archaea. We studies 80 of these organisms in our experiments. An overview of these organisms is shown in table 2. 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 [30]. 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.

5.1 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 glycolytic 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.

5.1.1 Phylogenetic trees for the datasets of 72 and 48 organisms

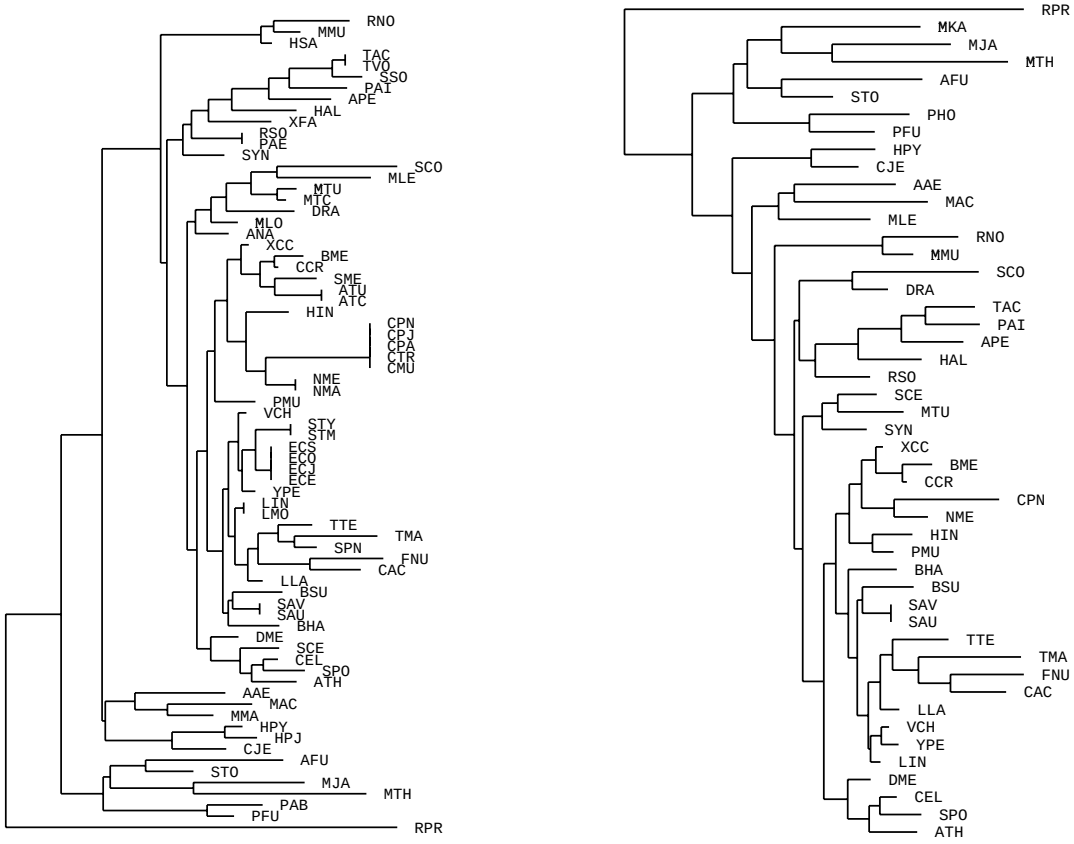
Figure 4 depicts the phylogenetic tree computed for the sets of 72 organisms and 48 organisms. With few exceptions, going from 72 to 48 organisms did not affect the relative position of the different organisms on the distance trees generated by our approach. This indicates the robustness of our technique. 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. Chlamedya 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 4(a), 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),

Code	Organism	D[a]	N_1^b	N_2^c	Code	Organism	D[a]	N_1^b	N_2^c
HSA	Homo sapiens	E	14	25	MMU	Mus musculus	E	9	24
RNO	Rattus norvegicus	E	8	20	DME	Drosophila melanogaster	E	14	21
CEL	Caenorhabditis elegans	E	15	21	ATH	Arabidopsis thaliana	E	13	20
SCE	Saccharomyces cerevisiae	E	13	22	SPO	Schizosaccharomyces pombe	E	12	21
ECO	Escherichia coli K-12 MG1655	B	14	23	ECJ	Escherichia coli K-12 W3110	B	14	23
ECE	Escherichia coli O157:H7 EDL933	B	14	23	ECS	Escherichia coli O157:H7 Sakai	B	14	23
STY	Salmonella typhi CT18	B	14	23	STM	Salmonella typhimurium LT2	B	14	23
YPE	Yersinia pestis CO92	B	11	21	HIN	Haemophilus influenzae Rd	B	11	17
PMU	Pasteurella multocida PM70	B	11	19	XFA	Xylella fastidiosa 9a5c	B	11	16
XCC	Xanthomonas campestris	B	11	20	PAE	Pseudomonas aeruginosa PA01	B	13	19
NME	Neisseria meningitidis	B	9	17	NMA	Neisseria meningitidis Z2491	B	9	17
RSO	Ralstonia solanacearum GMI1000	B	12	19	HPY	Helicobacter pylori 26695	B	6	11
HPJ	Helicobacter pylori J99	B	6	10	CJE	Campylobacter jejuni NCTC11168	B	10	12
MLO	Mesorhizobium loti MAFF303099	B	13	21	SME	Sinorhizobium meliloti 1021	B	14	22
ATU	Agrobacterium tumefaciens C58	B	12	20	ATC	Agrobacterium tumefaciens C58 (Cereon)	B	14	20
BME	Brucella melitensis 16M	B	14	20	CCR	Caulobacter crescentus	B	12	19
BSU	Bacillus subtilis 168	B	12	23	BHA	Bacillus halodurans C-125	B	13	21
SAU	Staphylococcus aureus N315	B	12	24	SAV	Staphylococcus aureus Mu50	B	11	24
LMO	Listeria monocytogenes EGD-e	B	7	22	LIN	Listeria innocua CLIP 11262	B	7	22
LLA	Lactococcus lactis IL1403	B	8	21	CAC	Clostridium acetobutylicum ATCC824	B	7	19
TTE	Thermoanaerobacter tengcongensis	B	8	18	MGE	Mycoplasma genitalium G-37	B	1	14
MPN	Mycoplasma pneumoniae M129	B	1	15	MTU	Mycobacterium tuberculosis H37Rv	B	14	20
MTC	Mycobacterium tuberculosis CDC1551	B	14	20	MLE	Mycobacterium leprae TN	B	11	16
SCO	Streptomyces coelicolor A3(2)	B	10	16	FNU	Fusobacterium nucleatum	B	5	15
CTR	Chlamydia trachomatis (serovar D)	B	8	13	CMU	Chlamydia muridarum	B	8	13
CPN	Chlamydophila pneumoniae CWL029	B	8	13	CPA	Chlamydophila pneumoniae AR39	B	8	13
CPJ	Chlamydophila pneumoniae J138	B	8	13	TPA	Treponema pallidum Nichols	B	1	10
SYN	Cyanobacteria Synechocystis	B	8	19	ANA	Anabaena sp. PCC7120	B	8	20
DRA	Deinococcus radiodurans R1	B	13	19	AAE	Aquifex aeolicus VF5	B	9	12
TMA	Thermotoga maritima MSB8	B	6	16	MJA	Methanococcus jannaschii DSM2661	A	8	6
MAC	Methanosarcina acetivorans C2A	A	8	14	MMA	Methanosarcina mazei Goe1	A	8	13
MTH	thermoautotrophicum deltaH	A	9	5	AFU	Archaeoglobus fulgidus DSM4304	A	8	7
HAL	Halobacterium sp. NRC-1	A	11	4	TAC	Thermoplasma acidophilum	A	10	13
TVO	Thermoplasma volcanium GSS1	A	10	13	PHO	Pyrococcus horikoshii OT3	A	4	6
PAB	Pyrococcus abyssi pab	A	4	7	PFU	Pyrococcus furiosus DSM3638	A	6	8
APE	Crenarchaeota Aeropyrum pernix K1	A	11	13	SSO	Sulfolobus solfataricus P2	A	12	12
STO	Sulfolobus tokodaii strain7	A	12	10	PAI	Pyrobaculum aerophilum IM2	A	9	13

Table 2: Organisms studied

[a] Domain:A..archaea, B..bacteria, E..eukarya; [b] N_1 number of enzymes in Citric Acid Cycle pathway; [c] N_2 number of enzymes in Glycolysis pathway



(a) 72 organisms

(b) 48 organisms

Figure 4: Phylogenetic trees built using the glycolysis pathway for (a) 72 organisms, and (b) 48 organisms.

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 taxomomy for the glycolysis pathway.

Technique	
Our technique	0.26
NCE technique	0.19
16S rRNA	0.22
Liao at al.'s technique	0.16

Table 4: Similarity measures based on the NCBI taxomomy for the glycolysis pathway.

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 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. (Note that the dataset of 48 organisms does not include exactly similar pathways, and therefore some of the above clusters do not exist there.)

We computed similarity measures based on the NCBI taxonomy using the *cousins* tool [49]. We also obtained phylogenetic trees for the NCE (Number of Common Enzymes) [15, 16, 17] in which the phylogenetic analysis is based on the number of common enzymes between two organisms. We are not comparing 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 3 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.

5.1.2 Phylogenetic trees for the dataset of 16 organisms

Figure 5 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 similarity measures using the NCBI taxonomy as the standard are shown in table 4 for our technique and three others: NCE, 16S rRNA, and Liao et al. Our method outperforms the other techniques. Table 5 shows the similarity measures when the 16S rRNA tree is chosen as the standard. Our method against obtains the best alignment.

5.2 Phylogenetic trees based on Kreb's Citric Acid Cycle (TCA cycle)

The evolutionary origin of the Krebs citric acid cycle (Krebs cycle) has long been a model in the understanding of the origin and evolution of metabolic pathways. Although the chemical steps of the cycle are preserved intact throughout nature, diverse organisms make diverse use of its chemistry. In some cases

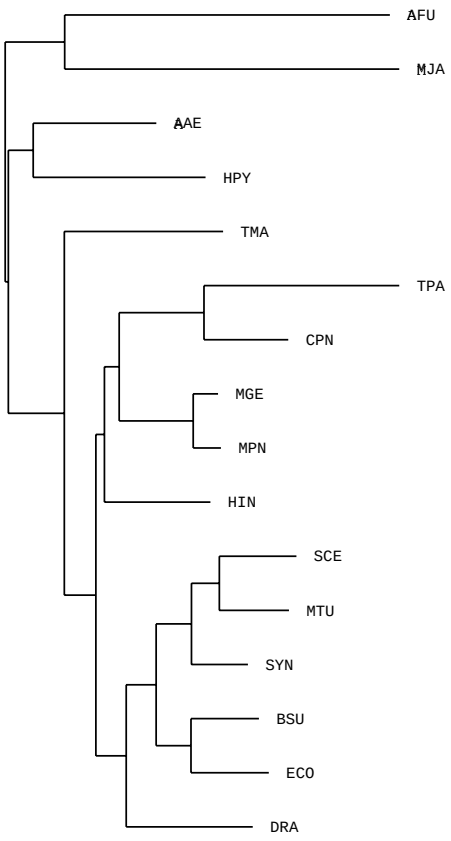
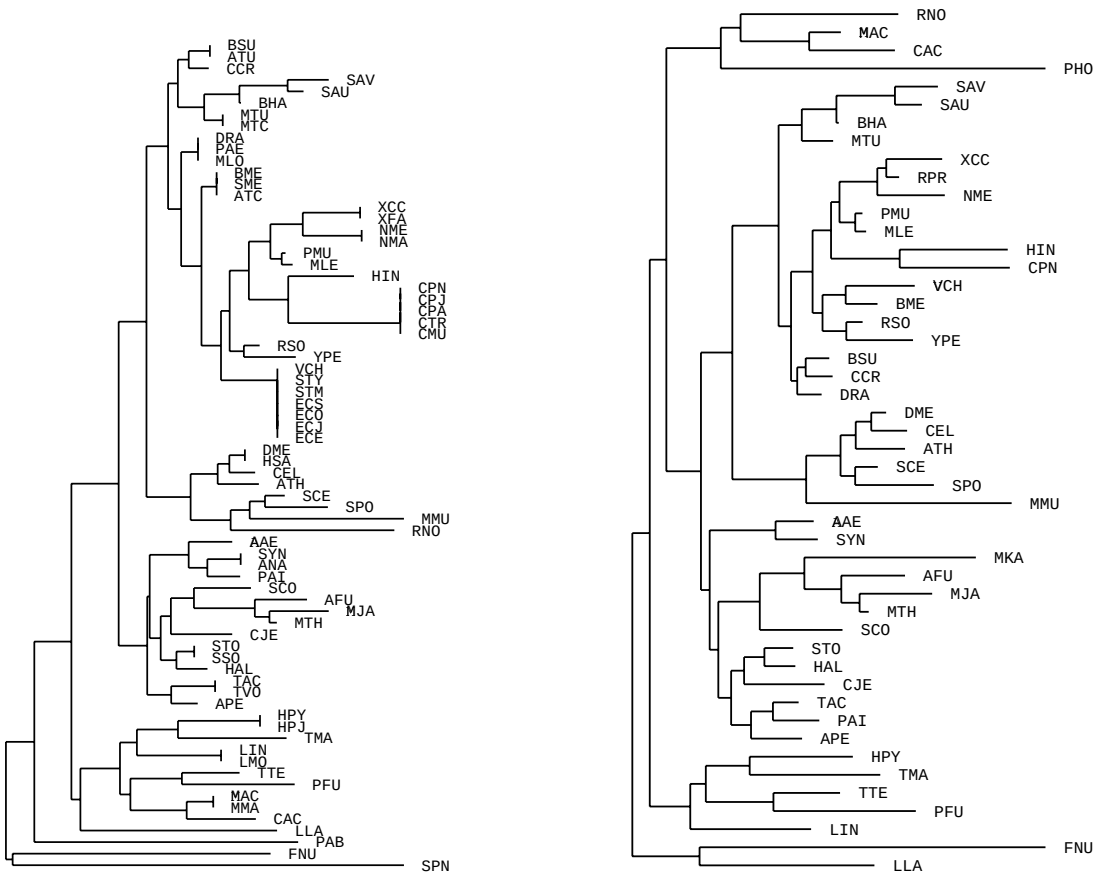


Figure 5: Phylogenetic tree for 16 organisms built from comparison of Glycolysis pathway.

Technique	
Our technique	0.27
NCE technique	0.18
Liao et al.'s technique	0.12

Table 5: Similarity measures based on the 16S rRNA tree for the glycolysis pathway.



(a) 72 organisms

(b) 48 organisms

Figure 6: Phylogenetic trees built using the Citric Acid Cycle pathways for (a) 72 organisms and (b) 48 organisms.

organisms use only selected portions of the cycle. Experiments were performed for three different sets of organisms (datasets of 72, 48, and 16 organisms). The results are presented next.

5.2.1 Phylogenetic trees for the datasets of 72 and 48 organisms

Figure 6 depicts the phylogenetic trees computed for the sets of 72 and 48 organisms. As in the case of the Glycolysis pathway, scaling up from 72 to 48 organisms did not affect the relative position of the different organisms on the distance trees. We can again notice that organisms within the same genus are clustered together: in figure 6(a), the E.coli and the salmonella organisms ECO, ECJ, ECS, ECE, STY, and STM have the same Citric Acid Cycle pathway and are clustered together. That is also the case for XCC and XFA, for the chlamydia CPN,CPJ, CPA, CTR, and CMU, for the mycobacteria MTU and MTC, for the helicobacter pylori HPY and HPJ, for the neisseria meningitidis NME and NMA, and for the methanosarcina MMA and MAC. We again find separate clusters corresponding to the three domains of life. In figure 6(a), all the archaea, PAI, AFU, MJA, MTH, STO, SSO, HAL, TAC, TVO, and APE are grouped together. All the

Technique	72 organisms	48 organisms
Our technique	0.25	0.27
NCE technique	0.24	0.28

Table 6: Similarity measures based on the NCBI taxomomy tree for the Citric Acid Cycle pathway.

	SCE	AFU	MJA	AAE	TMA	MTU	TPA	CPN	SYN	DRA	BSU	HPY	MGE	MPN	HIN	ECO
SCE	0.0	0.58	0.62	0.44	0.72	0.29	1.0	0.5	0.51	0.25	0.19	0. 67	1.0	1.0	0.5	0.41
AFU	0.58	0.0	0.21	0.25	0.46	0.41	1.0	0.48	0.37	0.45	0.41	0 .49	1.0	1.0	0.59	0.6
MJA	0.62	0.21	0.0	0.3	0.45	0.42	1.0	0.53	0.41	0.49	0.45	0. 64	1.0	1.0	0.64	0.63
AAE	0.44	0.25	0.3	0.0	0.53	0.35	1.0	0.46	0.1	0.3	0.25	0.44	1.0	1.0	0.55	0.47
TMA	0.72	0.46	0.45	0.53	0.0	0.58	1.0	0.65	0.48	0.64	0.61	0 .29	1.0	1.0	0.73	0.66
MTU	0.29	0.41	0.42	0.35	0.58	0.0	1.0	0.39	0.41	0.08	0.15	0 .56	1.0	1.0	0.35	0.26
TPA	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
CPN	0.5	0.48	0.53	0.46	0.65	0.39	1.0	0.0	0.4	0.37	0.32	0.69	1.0	1.0	0.27	0.41
SYN	0.51	0.37	0.41	0.1	0.48	0.41	1.0	0.4	0.0	0.37	0.32	0.3 8	1.0	1.0	0.5	0.41
DRA	0.25	0.45	0.49	0.3	0.64	0.08	1.0	0.37	0.37	0.0	0.07	0. 59	1.0	1.0	0.29	0.19
BSU	0.19	0.41	0.45	0.25	0.61	0.15	1.0	0.32	0.32	0.07	0.0	0 .55	1.0	1.0	0.33	0.25
HPY	0.67	0.49	0.64	0.44	0.29	0.56	1.0	0.69	0.38	0.59	0.55	0.0	1.0	1.0	0.7	0.61
MGE	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0
MPN	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0
HIN	0.5	0.59	0.64	0.55	0.73	0.35	1.0	0.27	0.5	0.29	0.33	0.7	1.0	1.0	0.0	0.2
ECO	0.41	0.6	0.63	0.47	0.66	0.26	1.0	0.41	0.41	0.19	0.25	0.61	1.0	1.0	0.2	0.0

Table 7: Distance matrix for the dataset of 16 organisms using the Citric Acid Cycle pathway.

eukaryota DME, HSA, CEL, ATH, SCE, SPO, MMU, and RNO are also grouped in a separate cluster. A few interesting clusterings are found for the bacteria. One cluster is composed of firmicutes BSU , SAU, SAV, BHA, MTU, and MTC. A few clusters are composed of proteobacteria, such as the cluster XCC, XFA, NME, NMA, and PMU and the cluster with proteobacteria alpha subdivision organisms BME, SME, and ATC. In figure 6(b), we find a cluster with organisms from the archaea domain: MKA, AFU, MJA,MTH, STO, HAL, TAC, PAI, and APE and another cluster with organisms from the eukaryota domain: DME, CEL,ATH, SCE, SPO, and MMU. For the bacteria, the firmicutes SAV, SAU, BHA, MTU are clustered together, and we also find a cluster of proteobacteria with XCC, RPR, NME, PMU, HIN, VCH, BME, RSO, and YPE.

Table 6 shows the similarity measures of our technique and the NCE technique obtained by using the NCBI taxonomy as the standard. As we see in table 6, the similarity measures for both techniques are similar.

5.2.2 Phylogenetic trees for the dataset of 16 organisms

Table 7 depicts the distance matrix obtained from our algorithm for the set of 16 organisms considered by Liao et al. [30]. Figure 7 depicts the phylogenetic tree obtained from this matrix. As can be seen, TPA, MGE, and MGN have no revealed Citric Acid Cycle and have therefore a distance measure of 1 with all the other organisms. In figure 7, the two archaea AFU and MJA are clustered together, they have a high similarity of 0.79. This similarity is the highest for both. The proteobacteria gamma sudivision HIN and ECO are also clustered together and have a similarity of 0.8. The two closest organisms are BSU and DRA

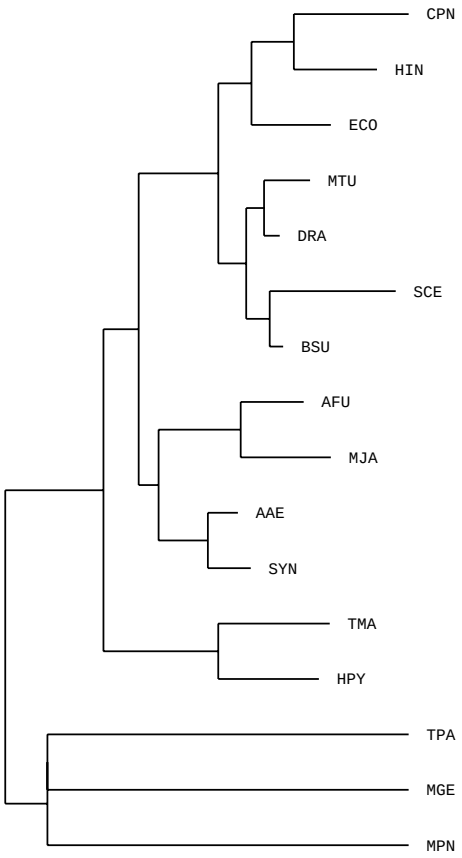


Figure 7: Phylogenetic trees for 16 organisms built from comparison of Citric Acid Cycle pathway.

Technique	
Our technique	0.42
NCE technique	0.2
16S rRNA	0.22
Liao's technique	0.16

Table 8: Similarity measures based on the NCBI taxonomy for the Citric Acid Cycle pathway.

Technique	
Our technique	0.23
NCE technique	0.19
Liao's technique	0.12

Table 9: Similarity measures based on the 16S rRNA tree for the Citric Acid Cycle pathway.

(with a distance of 0.08) and they actually belong to the same superfamily of Bacillus lipases. In the 16S rRNA based tree, BSU and DRA are grouped together.

Table 8 shows the similarity measures (using the NCBI taxonomy as the standard) for our technique and three others: NCE, 16S rRNA, and Liao et al. Our method outperforms the other techniques. Table 9 shows the similarity measures when the 16S rRNA tree is chosen as the standard. Our method again obtains the best alignment.

5.3 Phylogenetic trees based on carbohydrate & lipid 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, 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. Lipid metabolism is composed of the fatty acid biosynthesis (path 1 and path 2), fatty acid metabolism, synthesis and degradation of ketone bodies, sterol biosynthesis, bile acid biosynthesis, C21-Steroid hormone metabolism, and androgen and estrogen metabolism pathways. Lipids comprise one of the most important classes of complex molecules present in animal cells and tissues. Lipid diversity and level in the cells, tissues, and organs are determined by the processes of lipid metabolism (LM), which include lipid transport, consumption and intracellular utilization, de novo synthesis, degradation, and excretion. The processes of lipid metabolism require the involvement of numerous proteins with different functions. These proteins together with their genes are the components of the LM system. Interest in the LM system is due to its important role in the vital activity of the organism and to the fact that the distortions in its functioning are among the causes of different human diseases. The amount of experimental data on different peculiarities of functioning of this system has grown enormously.

The 8 organisms that we considered for this experiment are shown in table 10. We also indicate the number of enzymes present for each organism for the two groups of pathways. Tables 11 and 12 depict the distance matrix obtained from our algorithm for the two kinds of metabolisms. As we can see, the two archaea AFU and MJA are the two closest organisms with a distance of 0.55 in (a) (and of 0.54 in (b)). 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 in (a) (and of 0.8 in (b)). RNO is the closest organism to MMU.

Code	Organism	D[a]	N_3^b	N_4^c
RNO	Rattus norvegicus	Eukaryota	62	105
MMU	Mus musculus	Eukaryota	65	96
AFU	Archaeoglobus fulgidus	Archaea	45	54
MJA	Methanococcus jannaschii	Archaea	32	39
NME	Neisseria meningitidis	ProteoBacteria	60	68
HIN	Haemophilus influenzae	Proteobacteria	75	83
LIN	Listeria innocua	Bacteria Firmicute	77	89
BSU	Bacillus subtilis	Bacteria Firmicute	113	125

Table 10: Set of eight organisms
[a] Domain; [b] N_3 number of enzymes in carbohydrate metabolism; [c] N_4 number of enzymes in carbohydrate and lipid metabolisms

	CPE	MMU	RNO	AFU	MJA	NME	HIN	LIN
CPE	0.0	0.89	0.86	0.88	0.79	0.88	0.8	0.83
MMU	0.89	0.0	0.78	0.86	0.82	0.86	0.88	0.88
RNO	0.86	0.78	0.0	0.84	0.82	0.82	0.87	0.82
AFU	0.88	0.86	0.84	0.0	0.55	0.8	0.85	0.86
MJA	0.79	0.82	0.82	0.55	0.0	0.82	0.71	0.84
NME	0.88	0.86	0.82	0.8	0.82	0.0	0.83	0.82
HIN	0.8	0.88	0.87	0.85	0.71	0.83	0.0	0.79
LIN	0.83	0.88	0.82	0.86	0.84	0.82	0.79	0.0

Table 11: Distance matrix for the dataset of 8 organisms using carbohydrate metabolism.

In both trees, the bacteria CPE, HIN and LIN are clustered together. 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. Figure 8 depicts the phylogenetic trees computed for these organisms using the two pathway groups.

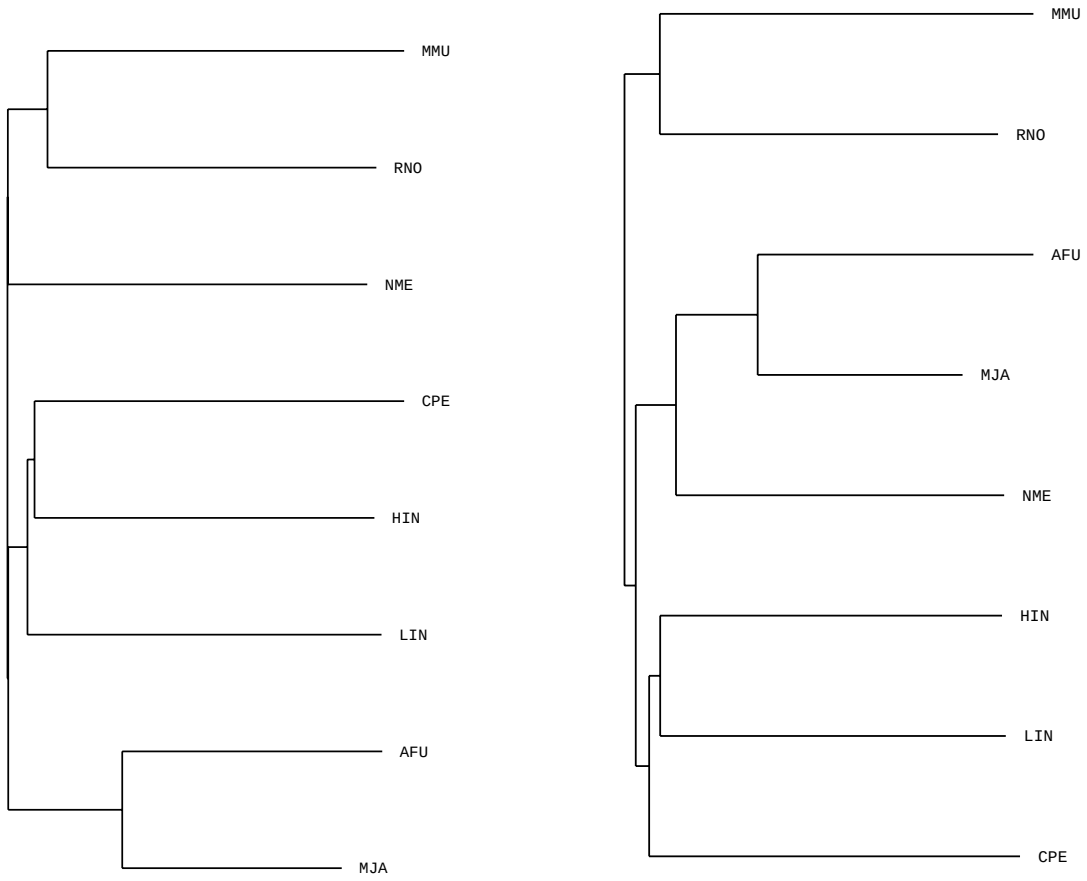
Finally, we evaluated our technique and the NCE technique for the two groups of pathways. We measured the correspondences of the generated phylogenies to the NCBI taxonomy. The similarity measures are shown in table 13. The tree obtained with our method for carbohydrate metabolism outperforms all the other trees. 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.

6 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

	CPE	MMU	RNO	AFU	MJA	NME	HIN	LIN
CPE	0.0	0.88	0.86	0.86	0.74	0.87	0.8	0.81
MMU	0.88	0.0	0.8	0.89	0.87	0.87	0.87	0.88
RNO	0.86	0.8	0.0	0.87	0.78	0.83	0.86	0.82
AFU	0.86	0.89	0.87	0.0	0.54	0.78	0.85	0.85
MJA	0.74	0.87	0.78	0.54	0.0	0.66	0.76	0.83
NME	0.87	0.87	0.83	0.78	0.66	0.0	0.8	0.8
HIN	0.8	0.87	0.86	0.85	0.76	0.8	0.0	0.76
LIN	0.81	0.88	0.82	0.85	0.83	0.8	0.76	0.0

Table 12: Distance matrix for the dataset of 8 organisms using carbohydrate and lipid metabolisms.



(a) Carbohydrate metabolism

(b) Carbohydrate and Lipid metabolisms

Figure 8: Phylogenetic trees for the dataset of 8 organisms built from comparison of (a) carbohydrate metabolism and (b) carbohydrate and lipid metabolisms.

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

Table 13: Similarity measures based on the NCBI taxonomy for the dataset of 8 organisms.

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.

Our experiments so far have considered only the reaction types of the enzymes in defining node similarity. In the future, we will experiment with other notions of distance (viz. sequence/structure similarity) in the proposed research.

Our algorithm for computing similarity between graphs can be extended by including information on the edges (in addition to the information on the nodes) of the graph. The edge labels can be based on the relationship between two enzymes in the enzyme graph, such as the chemical compound shared by the reactions activated by the enzymes. Similarity between chemical compounds can be defined by the "similarity" of their chemical formula. Other information concerning the metabolic pathways such as the inhibitory feedback interaction could also be included during graph comparison.

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 networks. We can either construct a single tree for the entire metabolic network, or combine trees from different metabolic pathways into a single consensus tree [46].

Our algorithm can also be used in the scoring of predicted (Computationally-derived) metabolic pathways. The new pathway could be compared and aligned to existing pathways using our algorithm to judge its validity.

References

[1] `http://www.genome.ad.jp/kegg/`.

[2] `http://www.ecocyc.org/`.

[3] `http://wit.mcs.anl.gov/WIT2/`.

[4] `http://evolution.genetics.washington.edu/phylip.html`.

[5] `http://jade.cs.pusan.ac.kr/phylodraw/`.

[6] `http://www.cs.nyu.edu/cs/faculty/shasha/papers/cousins.html`.

[7] Vincent Blondel and Paul Van Dooren. 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, 2002.

[8] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8):689–694, 1997.

[9] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3-4):255–259, 1998.

[10] G. Chartrand, G. Kubicki, and M. Schultz. Graph similarity and distance in graphs. *Aequationes Mathematicae*, 55(1-2):129–145, 1998.

[11] D.G. Corneil and C.C. Gotlieb. An efficient algorithm for graph isomorphism. *Journal of the ACM*, 17(1):51–64, 1970.

[12] M.L. de Buyser, A. Morvan, S. Aubert, F. Dilasser, and N. El Solh. Evaluation of a ribosomal RNA gene probe for the identification of species and subspecies within the genus Staphylococcus. *J. Gen. Microbiol.*, 138:889–899, 1992.

[13] J. Felsenstein. Phylogenies from molecular sequences: Inferences and reliability. *Annual Rev. Genet.*, 22:521–565, 1998.

[14] S.T. Fitz-Gibbon and C.H. House. Whole genome-based phylogenetic analysis of free-living microorganisms. *Nucleic Acids Research*, 27:4218–4222, 1999.

[15] Christian Forst and Klaus Schulten. Phylogenetic analysis of metabolic pathways. *Journal of Molecular Evolution*, 52:471–489, 2001.

[16] C.V. Forst and K. Schulten. 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)*, pages 174–181. ACM Press, 1999.

[17] C.V. Forst and K. Schulten. Evolution of metabolisms: A new method for the comparison of metabolic pathways using genomic information. *Journal of Computational Biology*, 6(3-4):343–360, 1999.

[18] S. Goto, H. Bono, H. Ogata, W. Fujibuchi, T. Nishioka, K. Sato, and M. Kanehisa. Organizing and computing metabolic pathway data in terms of binary relations. In *Proceedings of the Second Pacific Symposium on Biocomputing (PSB)*, pages 175–186, 1996.

[19] D. Harel and Y. Koren. Clustering spatial data using random walks. Technical Report MCS01-08, Deptartment of Computer Science and Applied Mathematics, The Weizmann Institute of Science, 2001.

[20] J.E. Hopcroft and R.M. Karp. An algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, December 1973.

[21] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, New York, 1990.

[22] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis.* Cambridge University Press, London, 1991.

[23] Glen Jeh and Jennifer Widom. SimRank: A measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, July 2002.

[24] M. Kanehisa. KEGG: From genes to biochemical pathways. In Stan Letovsky, editor, *Bioinfomatics: Databases and Systems*, pages 63–76. Kluwer Academic Publishers, 1999. ISBN 0-7923-8573-X.

[25] P. Karp and M. Riley. Representations of metabolic knowledge. In *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology (ISMB 1993)*, pages 207–215. AAAI Press, 1993.

[26] P.D. Karp and S.M. Paley. Automated drawing of metabolic pathways. In *Proceedings of the Third International Conference on Bioinformatics and Genome Research*, 1994.

[27] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1):83–97, 1955.

[28] R. Lempel and S. Moran. SALSA: Stochastic approach for link-structure analysis and the TKC e ect. *ACM Transactions on Information Systems*, 19:131–160, 2001.

[29] Ming Li, Jonathan H. Badger, Xin Chen, Sam Kwong, Paul Kearney, and Haoyong Zhang. An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, 17:149–154, 2001.

[30] Li Liao, Sun Kim, and Jean-Francois Tomb. Genome comparisons based on profiles of metabolic pathways. In *Sixth International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES 2002)*, Crema, Italy, September 2002.

[31] B.L. Maidak, J.R. Cole, T.G. Lilburn, C.T. Parker, P.R. Saxman, R.J. Farris, G.M. Garrity, G.J. Olsen, T.M. Schmidt, and J.M. Tiedje. The RDP-II (ribosomal database project). *Nucleic Acids Research*, 29:173–174, 2001.

[32] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: A versatile graph matching algorithm. In *Proceedings of Eighteenth International Conference on Data Engineering*, San Jose, California, February 2002.

[33] M. Nei. Phylogenetic analysis in molecular evolutionary genetics. *Annu. Rev. Genet.*, 30:371–403, 1996.

[34] Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB). Enzyme nomenclature recommendations of the nomenclature committee of the international union of biochemistry and molecular biology on the nomenclature and classification of enzyme-catalysed reactions. `http://www.chem.qmul.ac.uk/iubmb/enzyme/`.

[35] H. Ogata, H. Bono, W. Fujibuchi, S. Goto, and M. Kanehisa. Analysis of binary relations and hierarchies of enzymes in the metabolic pathways. *Genome Informatics*, 7:128–136, 1996.

[36] H. Ogata, W. Fujibuchi, S. Goto, and M. Kanehisa. A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. *Nucleic Acids Research*, 28:4021–4028, 2000.

[37] H. Ogata, S. Goto, W. Fujibuchi, and M. Kanehisa. Computation with the KEGG pathway database. *Biosystems*, 47:119–128, 1998.

[38] H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, and M. Kanehisa. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 27:29–34, 1999.

[39] Frank Olken. Phylogenetic tree computation tutorial. `http://pga.lbl.gov/Workshop/April2002/lectures/Olken.pdf`.

[40] A.N. Papadopoulos and Y. Manolopoulos. Structure-based similarity search with graph histograms. In *Proceedings of the DEXA/IWOSS International Workshop on Similarity Search*, pages 174–178. IEEE Computer Society, 1999.

[41] R.C. Read and D.G. Corneil. The graph isomorphism disease. *Journal of Graph Theory*, 1:339–363, 1977.

[42] H.A. Baier Saip and C.L. Lucchesi. Matching algorithms for bipartite graph. Technical Report DCC-03/93, Departamento de Cincia da Computao, Universidade Estudal de Campinas, 1993.

[43] N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 1987.

[44] A. Sanfeliu and K. Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 13:353–362, May 1983.

[45] P.H.A. Sneath and R.R. Sokal. *Numerical Taxonomy*, pages 230–234. W.H. Freeman, San Francisco, 1973.

[46] C. Stockham, L.-S. Wang, and T. Warnow. Statistically based postprocessing of phylogenetic analysis by clustering. In *Proceedings of the Tenth International Conference on Intelligent Systems for Molecular Biology (ISMB 2002)*, 2002.

[47] Yukako Tohsato, Hideo Matsuda, and Akihiro Hashimoto. A multiple alignment algorithm for metabolic pathway analysis using enzyme hierarchy. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB 2000)*, pages 376–383, 2000.

[48] J.R. Ullman. An algorithm for subgraph isomorphism. *Journal of the ACM*, 23(1):31–42, 1976.

[49] K. Zhang, J.T.L. Wang, and D. Shasha. On the editing distance between undirected acyclic graphs. *International Journal of Foundations of Computer Science*, 7(1):43–57, 1996.

7 Appendix

The similarity measures between nodes of graphs defined in section 4 can be computed as the principal eigenvector of some matrix. The mathematical proof follows.
Using the adjacency matrices A and B of G_1 and G_2 respectively, we can rewrite the terms A_1 to A_4 and D_1 to D_4 of section 4 as a product of matrices.

$$A_1^{(k)} = D1_{inA}^{-1} A^T S^k B D1_{inB}^{-1} + D2_{inA}^{-1}(I_1 - A^T) S^k (I_2 - B) D2_{inB}^{-1} \tag{24}$$

where I_1 is a $n_1 \times n_1$ matrix composed of only 1, I_2 is a $n_2 \times n_2$ matrix composed of only 1.
$D1_{inA}$ is a $n_1 \times n_1$ diagonal matrix with $D1_{inA}(a,a)$ ($a \in G_1$) as the indegree of node a if this indegree is different to 0 and 1 otherwise. $D2_{inA}$ is a diagonal matrix with $D2_{inA}^{-1}(a,a) = n_1^{-1}$ ($a \in G_1$) if $deg_{in}(a) = 0$ and $D2_{inA}^{-1}(a,a) = 0$ otherwise. $D1_{inB}$ is a $n_2 \times n_2$ diagonal matrix with $D1_{inB}(b,b)$ ($b \in G_2$) as the indegree of node b if this indegree is different to 0 and 1 otherwise. $D2_{inB}$ is a diagonal matrix with $D2_{inB}^{-1}(b,b) = n_2^{-1}$ ($b \in G_2$) if $deg_{in}(b) = 0$ and $D2_{inB}^{-1}(b,b) = 0$ otherwise.

$$A_2^{(k)} = D1_{outA}^{-1} A S^k B^T D1_{outB}^{-1} + D2_{outA}^{-1}(I_1 - A) S^k (I_2 - B^T) D2_{outB}^{-1} \tag{25}$$

where $D1_{outA}$, $D1_{outB}$, $D2_{outA}$ and $D2_{outB}$ are the corresponding matrices of $D1_{inA}$, $D1_{inB}$, $D2_{inA}$ and $D2_{inB}$ respectively for the outdegree.

$$A_3^{(k)} = D1_{notinA}^{-1}(I_1 - A^T) S^k (I_2 - B) D1_{notinB}^{-1} + D2_{notinA}^{-1} A^T S^k B D2_{notinB}^{-1} \tag{26}$$

where $D1_{notinA}$, $D1_{notinB}$, $D2_{notinA}$ and $D2_{notB}$ are the corresponding matrices of $D1_{inA}$, $D1_{inB}$, $D2_{inA}$ and $D2_{inB}$ respectively for the matrices $I_1 - A$ and $I_2 - B$.

$$A_4^{(k)} = D1_{notoutA}^{-1}(I_1 - A) S^k (I_2 - B^T) D1_{notoutB}^{-1} + D2_{notoutA}^{-1} A S^k B^T D2_{notoutB}^{-1} \tag{27}$$

$$D_1^{(k)} = D1_{inA}^{-1} A^T S^k (I_2 - B) D1_{notinB}^{-1} + D2_{inA}^{-1}(I_1 - A^T) S^k B D2_{notinB}^{-1} \tag{28}$$

$$D_2^{(k)} = D1_{notinA}^{-1}(I_1 - A^T) S^k B D1_{inB}^{-1} + D2_{notinA}^{-1} A^T S^k (I_2 - B) D2_{inB}^{-1} \tag{29}$$

$$D_3^{(k)} = D1_{outA}^{-1} A S^k (I_2 - B^T) D1_{notoutB}^{-1} + D2_{outA}^{-1}(I_2 - A) S^k B^T D2_{notoutB}^{-1} \tag{30}$$

$$A_4^{(k)} = D1_{notoutA}^{-1}(I_1 - A) S^k B^T D1_{outB}^{-1} + D2_{notoutA}^{-1} A S^k (I_2 - B^T) D2_{outB}^{-1} \tag{31}$$

and the equation (12) becomes

$$S^{(k+1)} = 1/4 \times (A_1^{(k)} + A_2^{(k)} + A_3^{(k)} + A_4^{(k)} - D_1^{(k)} - D_2^{(k)} - D_3^{(k)} - D_4^{(k)}\Big) .* Sim(a,b) \tag{32}$$

where $.*$ is the element-wise multiplication between matrix.

We redefine the similarity matrix S as a $n_1.n_2 \times 1$ vector $SV = vec(S^T)$ using the operator vec, where $S(a,b) = SV((a-1) * n_2 + b, 1)$ (the node a has an index between 1 and n_1 and the node b has an

index between 1 and n_2). The operator vec satisfies the property $vec(UXV) = (V^T \otimes U)vec(X)$ where $\otimes$ represent the Kronecker product between matrices (see Lemma 4.3.1 in [22]). The recurrence equations can be reexpressed by the simple form

$$SV^0 = vec(Sim), SV^{(k+1)} = \frac{C.SV^k}{\parallel C.SV^k \parallel_2} \tag{33}$$

where C is a $n_1n_2 \times n_1n_2$ matrix defined by

$$\begin{aligned} C = 1/4 \times SIM \times \Big(& D1^{-1}_{inA}A^T \otimes D1^{-1}_{inB}B^T + D2^{-1}_{inA}(I_1 - A^T) \otimes D2^{-1}_{inB}(I_2 - B^T) \\ & + D1^{-1}_{outA}A \otimes D1^{-1}_{outB}B + D2^{-1}_{outA}(I_1 - A) \otimes D2^{-1}_{outB}(I_2 - B) \\ & + D1^{-1}_{notinA}(I_1 - A^T) \otimes D1^{-1}_{notinB}(I_2 - B^T) + D2^{-1}_{notinA}A^T \otimes D2^{-1}_{notinB}B^T \\ & + D1^{-1}_{notoutA}(I_1 - A) \otimes D1^{-1}_{notoutB}(I_2 - B) + D2^{-1}_{notoutA}A \otimes D2^{-1}_{notoutB}B \\ & - D1^{-1}_{inA}A^T \otimes D1^{-1}_{notinB}(I_2 - B^T) - D2^{-1}_{inA}(I_1 - A^T) \otimes D2^{-1}_{notinB}B^T \\ & - D1^{-1}_{notinA}(I_1 - A^T) \otimes D1^{-1}_{inB}B^T - D2^{-1}_{notinA}A^T \otimes D2^{-1}_{inB}(I_2 - B^T) \\ & - D1^{-1}_{outA}A \otimes D1^{-1}_{notoutB}(I_2 - B) - D2^{-1}_{outA}(I_1 - A) \otimes D2^{-1}_{notoutB}B \\ & - D1^{-1}_{notoutA}(I_1 - A) \otimes D1^{-1}_{outB}B - D2^{-1}_{notoutA}(I_1 - A) \otimes D2^{-1}_{outB}B \Big) \end{aligned}$$

and SIM is a $n_1n_2 \times n_1n_2$ diagonal matrix with $SIM((a-1)*n_2+b, (a-1)*n_2+b) = Sim(a,b)$.

The computation of the similarity values between every pair of nodes (a, b) is therefore equivalent to the computation of the principal eigenvector of the matrix C defined in (34) using the power method.

To insure convergence of the iterations of the power method, the algebraic multiplicity of the principal eigenvalue ρ of the matrix C needs to be equal to one [21]. In this case the matrix has a unique principal eigenvector and the iterative process converges to the solution after few iterations. Determining from a matrix if its principal eigenvalue ρ is unique is not an easy process. Nevertheless, a few properties from the Theory of Matrix handle some special cases. When the matrix C is irreductible, by the Perron-Frobenius theory, the principal eigenvalue has a multiplicity of 1 [21]. In this case, the iterations will always converge. In the case of symmetric nonnegative matrix C, Blondel and Van Dooren state in [7] that the subsequences $SV^{(2k)}$ and $SV^{(2k+1)}$ defined from the sequence $SV^{(k+1)} = C.SV^k / \parallel C.SV^k \parallel_2$ converge both. When $-\rho$ is not an eigenvalue of the matrix C then the sequence SV_k simply converges.

For the enzyme-enzyme relational graphs, the matrix C is symmetric nonnegative when the reactions in the metabolic pathways are considered as reversible. Indeed, in this case the enzyme graphs become undirected and the adjacency matrices associated to them are symmetric. When the graphs G_1 and G_2 are undirected, the matrices A and B are symmetric ($A^T = A$ and $B^T = B$) and $D1_{inA} = D1_{outA}$, $D2_{inA} = D2_{outA}$, $D1_{inB} = D1_{outB}$, $D2_{inB} = D2_{outB}$, $D1_{notinA} = D1_{notoutA}$, $D2_{notinA} = D2_{notoutA}$, $D1_{notinB} = D1_{notoutB}$, $D2_{notinB} = D2_{notoutB}$. By the properties $(A+B)^T = A^T + B^T$ and $(A \otimes B)^T = A^T \otimes B^T$, we can see that in case of undirected graphs, the matrix C is symmetric and nonnegative.