Clustering and Classifying Enzymes in Metabolic Pathways: Some Preliminary Results

Sen Zhang
Computer Science Dept.
New Jersey Inst. Tech.
Newark, NJ 07102, USA

Li Liao
Jean-Francois Tomb
DuPont Experimental Station
Wilmington, DE 19880, USA

Jason T. L. Wang
Computer Science Dept.
New Jersey Inst. Tech.
Newark, NJ 07102, USA

ABSTRACT

In this paper we study data mining problems arising in the analysis of metabolic pathways. Each pathway is represented as a set of enzymes. These pathways are clustered according to their co-occurrence in various organisms. Each enzyme in a pathway is represented by a phylogenetic tree. By observing that pathways co-occurring in many organisms tend to have common enzymes, we propose new tree matching algorithms to cluster the pathways by clustering their enzymes (trees). Using the tree matching algorithms, we also develop a technique to classify enzymes in the pathways. We expect to apply these techniques to further studying evolution of metabolism.

1. INTRODUCTION

As more genomes are sequenced and metabolic pathways of organisms reconstructed, it becomes possible to perform detailed pathway comparisons, thus gaining insights into the evolution of metabolism. Often, metabolic pathways are profiled according to their presence and absence in organisms with completed genome sequences. A pathway is said to be present in an organism if all enzymes that are required for the pathway to function are found in the organism. Here, we are not concerned with whether a pathway will be really activated (i.e., producing the end product). In general, even if all its enzymes exist, the pathway may still not be active. In other words, being present here is different from being active. Notice also that each step in a pathway, representing one chemical reaction, normally performs a different and unique function in producing the end product. Putting in the context of a pathway network, there have to be stoichiometical relations to be conserved. However, we are not concerned with these details for the purpose of this study.

Thus, each pathway can be considered as a string of zeros and ones, corresponding to its absence and presence, respectively, in a set of organisms. In the study presented here, we consider 31 organisms. Thus, each pathway is a string of 31 bits, where a '0' value at the ith bit indicates that the pathway is absent in the ith organism and a '1' value at the jth bit indicates that the pathway is present in the jth organism. Such profiles have been utilized to compare genomes by using various scoring schemes, developed to compare profiles bearing hierarchical structures [2]. As pointed out in [2], these profiles can also be used to compare metabolic pathways. The scheme for comparing pathways in completed genomes consists of two steps. First, based on their profiles, co-occurrence of metabolic pathways in different organisms is determined. These pathways are then clustered based on their co-occurrence in the organisms, i.e. pathways that co-occur in the same set of organisms are grouped into one cluster. For example, Table 1 illustrates profiles of presence and absence of metabolic pathways $P_1, \ldots, P_5$ in organisms $Org_1, \ldots, Org_4$, respectively. P_1, P_3 and P_5 co-occur in Org_2 and Org_4 and hence are clustered into the same group. P_2 and P_4 co-occur in Org_1 and Org_3 and are clustered into another group. In the second step, co-evolution of enzyme members of these pathways is evaluated.

	Org_1	Org_2	Org_3	Org_4
P_1	0	1	0	1
P_2	1	0	1	0
P_3	0	1	0	1
P_4	1	0	1	0
P_5	0	1	0	1

Table 1: Profiles of presence and absence of metabolic pathways $P_1, \ldots, P_5$ in organisms $Org_1, \ldots, Org_4$.

Co-evolution of metabolic pathways has been mainly studied from substrate specificity [5]. In this paper we approach it from a different perspective, namely by comparing the phylogenetic trees of proteins involved in such pathways. In general, a pathway is an ordered sequence of enzymes. For our purpose, we ignore the order among the enzymes. For example, there are 4 enzymes in a pathway called "4HPPYRO2FUMAAC.CAT - 4-hydroxyphenylpyruvate, _O(,2)–fumarate, _acetoacetate_catabolism", with the following order: 1.13.11.27, 1.13.11.5, 5.2.1.2, 3.7.1.2. Here 1.13.11.27 refers to the enzyme with the enzyme commission number

```
     AP AG TH MJ PO PH AA BS BB CJ CQ CT CA DR EC HI HP ML MT MG MP NM PA RP CY TM TP UU AT CE SC
>16  N  Y  Y  Y  N  Y  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N
  PRPPGLNIMPATP10FTHMP.ANA −
 5−phosphoribosyl_1−diphosphate,_glutamine−−IMP_anabolism_(ATP,_10−formyltetrahydrometh
  2.4.2.14  6.3.5.3  4.1.1.21  6.3.2.6  4.3.2.2  6.3.4.13  6.3.3.1  2.1.2.2  2.1.2.3  3.5.4.10
  SAHMET5MTHMPT.CAT −
 'S'−adenosylhomocysteine−−methionine_(5−methyltetrahydromethanopterin)_catabolism
  3.3.1.1
  PRPPHISARC.ANA − phosphoribosyldiphosphate−−histidine_anabolism_[Archaea]
  2.4.2.17  3.6.1.31  3.5.4.19  5.3.1.16  4.2.1.19  2.6.1.9
  EF2HISEF2DADPIPLM.ANA −
  elongation_factor−2_L−histidine−−elongation_factor_2−diphthamide,_orthophosphate_anabolis
  2.1.1.98  6.3.2.22
```

Figure 1: Pathways and enzymes in cluster 16.

(EC#) 1.13.11.27. We represent this pathway simply as a set of the four enzymes, ignoring the order among these enzymes.

2. GENOME AND PATHWAY DATA

We considered 2719 pathways selected from 31 organisms in the WIT database [3]. Table 2 lists these genomes. By applying the co-occurrence criterion described in the previous section, we clustered these pathways into 69 groups. For instance, cluster 16, which is shown in Figure 1, consists of 4 pathways that all appear in A. fulgidus (AG), P. horikoshii (PH), M. jannaschii (MJ), and M. thermoautotrophicum (TH), but not in any other of the 31 organisms.

A phylogenetic tree is built for each of the enzymes present in each of the metabolic pathways. For example, proteins that are present in the four organisms AG, PH, MJ and TH, and belong to the enzyme with the enzyme commission number (EC#) 2.4.2.14 are first aligned using Clustal W. A phylogenetic tree is then reconstructed using the neighbor joining method from the result of multiple sequence alignment. This phylogenetic tree will be used to represent the enzyme 2.4.2.14 in our study (see Figure 2).

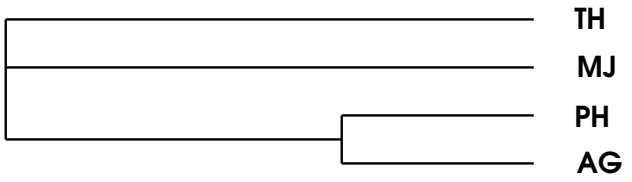

Figure 2: The phylogenetic tree for enzyme 2.4.2.14.

We hypothesize that the "similarity" among phylogenetic trees of enzyme members of clustered metabolic pathways indicates co-evolution and that the "dissimilarity" among the phylogenetic trees might be clues for other evolutionary phenomena, such as lateral transfer. In the next section we propose (dis)similarity measures for comparing two phylogenetic trees.

3. (DIS)SIMILARITIES OF TREES

A phylogenetic tree is a labeled, unordered tree where the order among siblings is unimportant and each node has a label. From Figure 2 it can be seen that in phylogenetic trees, node labels only occur in leaves (where each label represents a genome or organism name) and interior nodes do not have labels. We present two (dis)similarity measures for comparing these trees. Let T_1 and T_2 be two trees. Let S_1 be the set of leaves of T_1 and let S_2 be the set of leaves of T_2. Define the similarity between T_1 and T_2, denoted $d_x(T_1, T_2)$, to be

$$d_x(T_1, T_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$$

where $|.|$ denotes the set cardinality. Thus, $d_x(T_1, T_2)$ measures the difference between the leaves of T_1 and T_2.

The second metric is to measure the structural difference of T_1 and T_2. We propose a parameterized distance, denoted $d_y(T_1, T_2)$, which is an extension of the degree-2 editing distance between two unordered trees previously reported in [4; 7; 8]. We number the nodes in a tree based on preorder traversal and use a dynamic programming algorithm to calculate $d_y(T_1, T_2)$. Specifically, let $t_1[i]$ be the ith node in T_1 and let $t_2[j]$ be the jth node in T_2. Let $t_1[i_1], \ldots, t_1[i_m]$ be the children of $t_1[i]$ and let $t_2[j_1], \ldots, t_2[j_n]$ be the children of $t_2[j]$. Let $T_1[i]$ be the subtree rooted at $t_1[i]$ and let $T_2[j]$ be the subtree rooted at $t_2[j]$. There are three cases to be considered when calculating $d_y(T_1[i], T_2[j])$.

1. $T_1[i]$ is matched with $T_2[j_t]$, for some t, $1 \leq t \leq n$. In this case, all the subtrees rooted at the children of $t_2[j]$ (except $t_2[j_t]$) must be inserted. Hence

$$d_y(T_1[i], T_2[j]) = c \times \min_{1 \leq t \leq n} \{d_y(T_1[i], T_2[j_t])$$

Code	Class	Name, Number of Orfs, and Length
AP	Archaea	Aeropyrum pernix, 1631 ORF's, 1669 Kb
AG	Archaea	Archaeoglobus fulgidus, 2491 ORF's, 2178 Kb
TH	Archaea	Methanobacterium thermoautotrophicum, 1866 ORF's, 1751 Kb
MJ	Archaea	Methanococcus jannaschii, 1811 ORF's, 1739 Kb
PO	Archaea	Pyrococcus abysii, 1874 ORF's, 1765 Kb
PH	Archaea	Pyrococcus horikoshii, 1825 ORF's, 1738 Kb
AA	Bacteria	Aquifex aeolicus, 1744 ORF's, 1590 Kb
BS	Bacteria	Bacillus subtilis, 4093 ORF's, 4214 Kb
BB	Bacteria	Borrelia burgdorferi, 1666 ORF's, 1519 Kb
CJ	Bacteria	Campylobacter jejuni, 1633 ORF's, 1641 Kb
CQ	Bacteria	Chlamydia pneumoniae CWL029, 993 ORF's, 1230 Kb
CT	Bacteria	Chlamydia trachomatis D/UW-3/Cx, 867 ORF's, 1057 Kb
CA	Bacteria	Clostridium acetobutylicum, 3967 ORF's, 4030 Kb
DR	Bacteria	Deinococcus radiodurans, 3103 ORF's, 3284 Kb
EC	Bacteria	Escherichia coli, 4285 ORF's, 4639 Kb
HI	Bacteria	Haemophilus influenzae, 1846 ORF's, 1830 Kb
HP	Bacteria	Helicobacter pylori, 1547 ORF's, 1667 Kb
ML	Bacteria	Mycobacterium leprae, 2940 ORF's, 3807 Kb
MT	Bacteria	Mycobacterium tuberculosis, 3924 ORF's, 4411 Kb
MG	Bacteria	Mycoplasma genitalium, 532 ORF's, 580 Kb
MP	Bacteria	Mycoplasma pneumoniae, 674 ORF's, 816 Kb
NM	Bacteria	Neisseria meningitidis ser. A (str. Z2491), 1838 ORF's, 2168 Kb
PA	Bacteria	Pseudomonas aeruginosa, 5626 ORF's, 6246 Kb
RP	Bacteria	Rickettsia prowazekii, 854 ORF's, 1111 Kb
CY	Bacteria	Synechocystis sp., 3226 ORF's, 3573 Kb
TM	Bacteria	Thermotoga maritima, 1846 ORF's, 1860 Kb
TP	Bacteria	Treponema pallidum, 1031 ORF's, 1138 Kb
UU	Bacteria	Ureaplasma urealyticum, 646 ORF's, 751 Kb
AT	Eucaryota	Arabidopsis thaliana, 9460 ORF's, 36506 Kb
CE	Eucaryota	Caenorhabditis elegans, 16639 ORF's, 100096 Kb
SC	Eucaryota	Saccharomyces cerevisiae, 6259 ORF's, 12057 Kb

Table 2: A list of genomes used in this study.

$$+ \sum_{1 \le q \le n, q \ne t} d_y(\emptyset, T_2[j_q])\} + \mu(\lambda, t_2[j])$$

Here $d_y(\emptyset, T_2[j_q])$ is the cost of inserting the subtree $T_2[j_q]$, and $\mu(\lambda, t_2[j])$ is the cost of inserting the node $t_2[j]$. In contrast to the formulas in [4; 7; 8], we introduce here a real number c as a weighting parameter.

2. $T_2[j]$ is matched with $T_1[i_s]$, for some s, $1 \le s \le m$. In this case, all the subtrees rooted at the children of $t_1[i]$ (except $t_1[i_s]$) must be deleted. Hence

$$d_y(T_1[i], T_2[j]) = c \times \min_{1 \le s \le m} \{d_y(T_1[i_s], T_2[j])$$

$$+ \sum_{1 \le p \le m, p \ne s} d_y(T_1[i_p], \emptyset)\} + \mu(t_1[i], \lambda)$$

Here $d_y(T_1[i_p], \emptyset)$ is the cost of deleting the subtree $T_1[i_p]$, and $\mu(t_1[i], \lambda)$ is the cost of deleting the node $t_1[i]$.

3. $t_1[i]$ is matched with $t_2[j]$. In this case, we can construct a weighted bipartite graph between the children of $t_1[i]$ and the children of $t_2[j]$ and find the optimal matching between the children as described in [4; 7; 8]. Let C_1 be the cost incurred by matching $t_1[i]$ with $t_2[j]$ and let C_2 be the cost incurred by matching the children of $t_1[i]$ and the children of $t_2[j]$ obtained from the optimal bipartite matching. Then

$$d_y(T_1[i], T_2[j]) = C_1 + c \times C_2$$

The distance $d_y(T_1[i], T_2[j])$ is obtained from the minimum of the above three cases. This recurrence formula suggests to use a dynamic programming algorithm to calculate $d_y(T_1, T_2)$. As in [7; 8], we use the unit cost for all editing operations (insert, delete, and relabel nodes).

Figure 3 illustrates the parameterized distances between trees. When we set the parameter c value to 0.5, the distance between D_1 and p is 1.875, and the distance between D_2 and p is 1.8125. When we set the c value to 1, which becomes the degree-2 editing distance [7; 8], the distance between D_1 and p, and the distance between D_2 and p, is 6, respectively (representing the cost of deleting the 6 nodes not touched

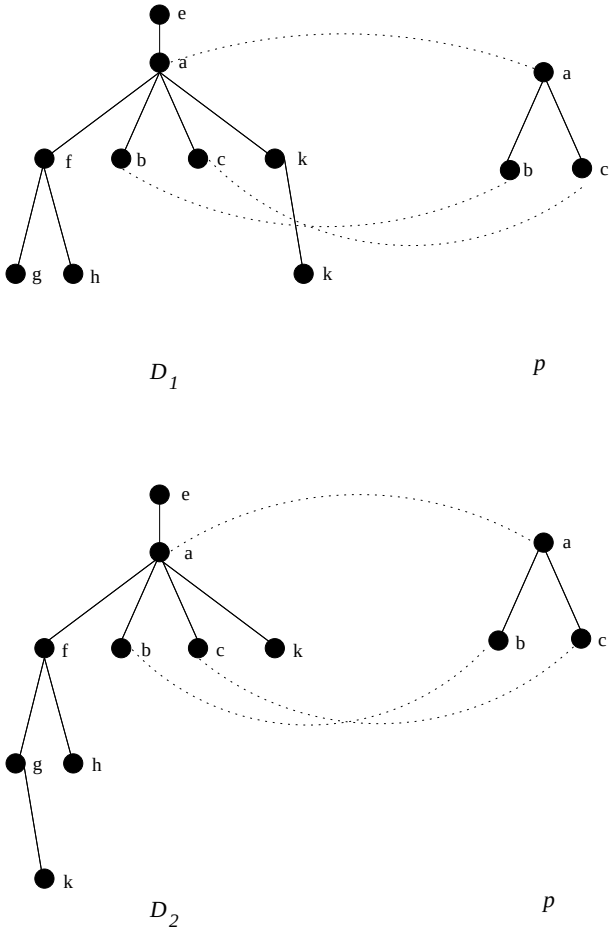

Figure 3: Illustration of parameterized distances between trees.

by the dotted mapping lines in the figure). Note D_1 differs from D_2 topologically. This example shows that the proposed parameterized distance better reflects the structural difference between trees than the degree-2 editing distance when the c value is properly chosen.

Remark: Our algorithm considers node labels for both leaves and non-leaves. In dealing with phylogenetic trees, we assign a label to each interior node of a phylogenetic tree by concatenating the labels occurring in its children.

4. CLUSTERING AND CLASSIFYING ENZYMES

After describing how to calculate the (dis)similarity values between two phylogenetic trees, we now turn to the description of the algorithms for clustering and classifying these trees (enzymes). Both of the algorithms are *distance-based*. They use the two distance measures $[d_x, d_y]$ together to compare the trees.

Our clustering algorithm is based on an agglomerative hierarchical clustering technique [1], which proceeds by iteratively considering all pairs of clusters built so far, and merging the pair that exhibits the greatest similarity into a single group (which then becomes a node of the dendrogram). The algorithm continues the merging until a pre-determined condition is met. In our case, each phylogenetic tree itself is a group initially. We use the group-average linkage method to determine which two groups should be linked (merged). Let group G_1 contain enzymes (trees) $P_1, \ldots, P_p$ and let group G_2 contain enzymes (trees) $Q_1, \ldots, Q_q$. The similarity of the two groups is obtained from averaging the d_x values as follows:

$$\frac{\sum_{1 \le i \le p} \sum_{1 \le j \le q} d_x(P_i, Q_j)}{p \times q}$$

We pick and merge the two groups with the greatest similarity. Suppose there are several pairs of groups that tie (i.e., they have the same similarity value). We then consider the distance between these groups. Specifically, the distance between group G_1 and group G_2 is obtained from averaging the d_y values as follows:

$$\frac{\sum_{1 \le i \le p} \sum_{1 \le j \le q} d_y(P_i, Q_j)}{p \times q}$$

Among the pairs of groups that tie on their similarity values, we pick and merge the pair of groups with the smallest distance. Intuitively we first consider the difference between leaves of trees in two groups, and if there is a tie, we consider the difference between structures of the trees in the two groups.

We also develop a nearest neighbor classifier [6] to classify the phylogenetic trees by utilizing the pair of (dis)similarity values $[d_x, d_y]$. A test tree T is assigned to a class C if C contains a training tree that has the largest d_x value to T. If there are several classes satisfying this condition, we assign

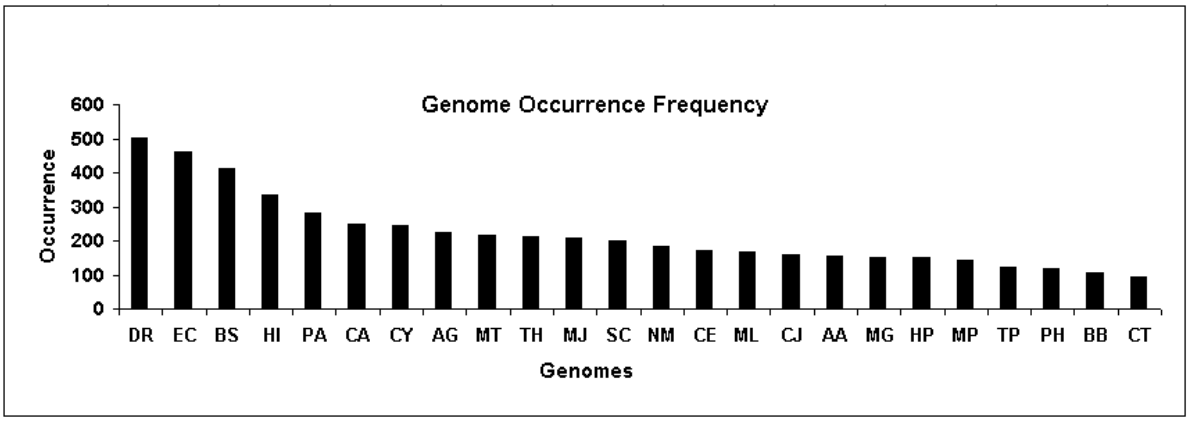

Figure 4: Occurrence frequency of genomes in the leaves of trees.

d_x	d_y				
	$c = 0.2$	$c = 0.6$	$c = 0.8$	$c = 1$	$c = 2$
0.778	1.804	6.724	14.114	30	879

Table 3: The distance and similarity values for two enzyme trees.

T to one of these classes, C^*, where C^* contains a training tree that has the smallest d_y value from T. This technique can be easily generalized to classify metabolic pathways. If the majority of the enzymes (trees) of a pathway is assigned to a class, then the pathway should belong to that class.

5. EXPERIMENTS AND RESULTS

We carried out a series of experiments on the pathway data to evaluate the performance of our approach. The programs were written in c/c++ programming languages and run on a Sun Ultra60 workstation under the Solaris operating system 5.8.

There were 523 phylogenetic trees distributed into 69 groups. Figure 4 shows the occurrence frequency of genomes in the leaves of these trees. It can be seen that the genome DR occurs most frequently in the trees. Pairwise distances and similarities for the trees were calculated. For example, Table 3 shows the d_x and d_y values for two trees (enzyme 3.4.23.46 in group 28 and enzyme 6.3.2.17 in group 46) with respect to different c values.

In the experiments, we set the c value to 0.6 for calculating d_y. Our experimental results show that our algorithms achieved a 100% correct rate in both clustering and classifying the phylogenetic trees.

6. CONCLUSION AND FUTURE WORK

In this paper we have presented two metrics for measuring the similarity and distance of two enzyme trees. We then use these metrics to cluster and classify the enzymes in metabolic pathways. Our experimental results showed a 100% correct rate, indicating the significance of our approach. Future work includes the development of new distance measures for comparing metabolic pathways and for studying pathway evolution.

7. ACKNOWLEDGMENTS

This work was supported in part by NSF grant IIS-9988636. The authors thank Professors Dennis Shasha and Kaizhong Zhang for useful discussions.

8. REFERENCES

[1] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis.* John Wiley & Sons, Inc., 1990.

[2] L. Liao, S. Kim, and J.-F. Tomb. Genome comparison based on profiles of metabolic pathways. In *Special Session on Machine Learning in Bioinformatics, Knowledge-Based Intelligent Information and Engineering Systems*, Crema, Italy, September 2002.

[3] R. Overbeek, N. Larsen, G. D. Pusch, M. D'Souza, E. Selkov Jr, N. Kyrpides, M. Fonstein, N. Maltsev, and E. Selkov. Wit: integrated system for high-throughput genome sequence analysis and metabolic reconstruction. *Nucleic Acids Research*, 28(1):123–125, 2002.

[4] D. Shasha, J. T. L. Wang, K. Zhang, and F. Y. Shih. Exact and approximate algorithms for unordered tree matching. *IEEE Transactions on Systems, Man and Cybernetics*, 24(4):668–678, April 1994.

[5] L. Sun, I. P. Petrounia, M. Yagasaki, G. Bandara, and F. H. Arnold. Expression and stabilization of galactose oxidase in escherichia coli by directed evolution. *Protein Engineering*, 14:699–704, 2001.

[6] C. W. Therrien. *Decision Estimation and Classification*. John Wiley & Sons, Inc., 1989.

[7] J. T. L. Wang, K. Zhang, G. Chang, and D. Shasha. Finding approximate patterns in undirected acyclic graphs. *Pattern Recognition*, 35(2):473–483, 2002.

[8] 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):43–58, 1996.