

## Alignment of Metabolic Pathways

Ron Y. Pinter<sup>a,\*</sup>, Oleg Rokhlenko<sup>a</sup>, Esti Yegeer-Lotem<sup>a</sup>,  
Michal Ziv-Ukelson<sup>a</sup>

<sup>a</sup>Dept. of Computer Science, Technion - Israel Institute of Technology,  
Haifa 32000, Israel

### ABSTRACT

**Motivation:** Several genome-scale efforts are underway to reconstruct metabolic networks for a variety of organisms. As the resulting data accumulates, the need for analysis tools increases. A notable requirement is a pathway alignment finder that enables both the detection of conserved metabolic pathways among different species as well as divergent metabolic pathways within a species. When comparing two pathways, the tool should be powerful enough to take into account both the pathway topology as well as the nodes' labels (*e.g.* the enzymes they denote), and allow flexibility by matching similar — rather than identical — pathways.

**Results:** *MetaPathwayHunter* is a pathway alignment tool that, given a query pathway and a collection of pathways, finds and reports all approximate occurrences of the query in the collection, ranked by similarity and statistical significance. It is based on a novel, efficient graph matching algorithm that extends the functionality of known techniques. The program also supports a visualization interface with which the alignment of two homologous pathways can be graphically displayed.

We employed this tool to study the similarities and differences in the metabolic networks of the bacterium *Escherichia coli* and the yeast *Saccharomyces cerevisiae*, as represented in highly curated databases. We reaffirmed that most known metabolic pathways common to both species are conserved. Furthermore, we discovered a few intriguing relationships between pathways that provide insight into the evolution of metabolic pathways. We conclude with a description of biologically meaningful meta-queries, demonstrating the power and flexibility of our new tool in the analysis of metabolic pathways.

**Keywords:** metabolic pathways, pathway analysis, conserved pathways, approximate tree matching.

**Availability:** Code and data upon request.

**Contact:** pinter@cs.technion.ac.il

### INTRODUCTION

Genome-scale metabolic networks are now being reconstructed for a variety of organisms such as *Escherichia coli*, *Saccharomyces cerevisiae*, and human. The wealth of information regarding the chemical reactions that take place within a cell and the corresponding enzymes that catalyze these reactions is currently stored in several public databases, including KEGG (Kanehisa and Goto, 2000), EcoCyc (Karp *et al.*, 2004) and SGD (Christie *et al.*, 2004). These databases maintain information about complex cellular processes, such as metabolism, signal transduction and cell cycle, by storing the corresponding networks of interacting molecules in digital forms, often as graphical pathway diagrams. The majority of these databases provide tools for pathway visualization and for queries on pathway components such as substrates, products and reactions. However, the need arises for good tools capable of searching for homologues to a query pathway in a collection of known pathways, and of aligning two pathways to locate conserved pathway fragments.

Pathway alignments should reflect both the similarity (rather than identity) between the enzymes that participate in the aligned pathways, as well as between their topologies. The need for advanced tools for pathway analysis will increase over the next several years as biologists begin not only to inspect existing pathways but also to redirect and re-engineer metabolic pathways. The latter objective, called *Metabolic Pathway Tinkering* (Newgard, 2002), requires thorough analysis of metabolic pathways, and brings up the need for formalizing specific, flexible queries on pathway databases.

Work to date on pathway searching has been limited to heuristics that try to capture certain properties of the underlying graphs and use them as measures of similarity, as in (Ogata *et al.*, 2000), and to visual inspection, sometimes aided by tools such as described in (Schreiber, 2003). Another attempt was undertaken by Tohsato *et al.* (Tohsato *et al.*, 2000) who proposed a method for multiple alignment of metabolic pathways, but restricted the pathways' topology to chains (or strands). Related work, which data-mines chains in protein-protein networks, is described in (Kelley *et al.*, 2003). Recently, Koyutürk *et al.* (Koyutürk *et al.*, 2004) presented a related mining approach where frequently occurring patterns

\*To whom correspondence should be addressed

(that can be general graphs) are detected in biological networks. Still, they do not address the search scenario, and — moreover — they state that the issue of *approximate* (rather than exact) matching is an important open problem.

In order to comprehensively search and mine metabolic pathways we developed *MetaPathwayHunter*, a novel tool for pathway alignment which is based on a powerful and efficient approximate pattern matching algorithm for labeled graphs. Our alignment model, the algorithm supporting it, and its implementation are described in the *System and Methods* section. We employed *MetaPathwayHunter* to conduct a study on the similarities and variations in the metabolic networks of two organisms, *E. coli* and *S. cerevisiae*, that serve as model organisms for pro- and eukaryotes, respectively, and observed several biologically interesting findings. Furthermore, we provide a description of meta-pathway queries that enable the user to probe the metabolic pathways database in a most flexible yet powerful manner. The experiments, their results, and the usage of meta-pathway queries are described in the *Results* section. We conclude with a brief discussion and some suggestions for future work.

## SYSTEM AND METHODS

In order to compare pathways to each other using a quantitative measure, we must represent them as mathematical objects that lend themselves to effective computation. Here we represent a pathway by a graph whose nodes correspond to enzymes that catalyze the pathway's reactions, and the edges connect two nodes if for the corresponding enzymes the product of one serves as the substrate of the other. When computing the similarity between metabolic pathways, we take into account both the resemblance between any two corresponding nodes in the pathway graph as well as the likeness between the pathways' network structure. The former reflects the similarity between matched enzymes, based on functional homology, and the latter checks for topological similarity between the graphs in a biologically meaningful way.

When comparing two pathways we try to *align* them to each other as best we can. Similarly to the alignment of genomic and proteomic sequences, we match pathways up in such a way that similar ingredients are paired with each other while minimizing the differences between them. These differences pertain both to the nodes, where enzymes of similar function are deemed close to each other, as well as to the connections between the nodes, namely the edges and paths that form the structure of the pathway.

As in sequence alignment, the closeness between two pathways is reflected by a *score* that is obtained by computing a function that measures the distance in a meaningful manner. Our method exhaustively computes all optimal solutions under a given scoring model. Furthermore, suboptimal solutions (up to a predefined threshold score) are reported,

ranked by their statistical significance. This is clearly preferable both to naive visual inspection which is expensive and prone to human errors, as well as to standard heuristic search methods which are likely to overlook some of the relevant results.

In this section we first describe our graph similarity measure and define the alignment score. They are limited to tree-like graphs in order to allow efficient alignment based on graph matching algorithms which are described next. Then we show how this method is highly applicable to metabolic pathways and explain how we compute the statistical significance of the score. We conclude this section with a few details concerning the implementation of our tool.

## Model

The topology of a metabolic pathway, similarly to other biological networks, can be represented as a graph. Thus the structural similarity among pathways can be naively revealed using techniques for solving various subgraph isomorphism and homeomorphism problems (Garey and Johnson, 1979) (formally defined in the next paragraph). Unfortunately, both problems are NP complete, rendering their solution intractable. Dealing with the association of individual nodes with *e.g.* similar rather than identical enzymes would then make the algorithmic problems even more complicated and computationally intensive. Still, an approach which utilizes typical properties of metabolic pathway graphs to simplify the problem at hand leads to tractable, efficient solutions. Our study shows that the topology of most metabolic pathways can be easily cast as multi-source trees or transformed to them without much loss of generality, as cycles are quite rare in this data. A *multi-source* tree is a directed acyclic graph (DAG), whose underlying undirected graph is a tree (see Figure 1), where some of the nodes can have several incoming as well as several outgoing edges.

There are several, increasingly complex yet tractable ways to model the problem of comparing trees to each other. A starting point is the *subtree isomorphism problem* (Matula, 1968, 1978; Shamir and Tsur, 1999): Given a pattern tree  $P$  and a text tree  $T$ , find a subtree of  $T$  which is isomorphic to  $P$ , *i.e.* find if some subtree of  $T$  that is identical in structure to  $P$  can be obtained by removing entire subtrees of  $T$ , or decide that there is no such tree. The *subtree homeomorphism problem* (Chung, 1987; Reyner, 1977; Valiente, 2003) is a variant of the former problem, where degree-2 nodes can be deleted from the text tree (see Figure 1).

We base our metabolic pathway alignment engine on the subtree homeomorphism model for reasons that are both biologically and computationally driven. Biologically, a single enzyme in one pathway may replace a few consecutively acting enzymes in another pathway. The replacement can take place if the replacing enzyme is multifunctional and can thus catalyze several consecutive reactions, or if the enzyme uses an alternative catalysis that leads directly from the initial

substrate to the final product. Note that enzymes that catalyze just a single reaction are more likely to be replaced than those that catalyze more reactions, for both biochemical and parsimony-related reasons. Translating this biological description into graph terms implies that degree-2 nodes may be deleted from the graph, a behavior which is perfectly captured by subtree homeomorphism.

Computationally, the advantage of subtree homeomorphism over the more complex models (such as Kilpelainen and Mannila, 1995) is in that it has tractable solutions. Complicating the model by e.g. allowing the deletions from both sides would render the problem intractable.

The model we employ extends previously known exact tree matching models which allowed nodes to be matched only if their labels were identical. Our model, on the other hand, is based on an approximate pattern matching algorithm *i.e.* it enables matching two nodes with distinct labels, and scores the match according to the similarity between the nodes.

**Definitions.** Let  $\Delta$  denote a predefined node-to-node similarity score table and  $\delta$  denote a predefined (usually negative) score for deleting a node from a tree (see Figure 1). A mapping  $\mathcal{M}[T_1, T_2]$  from  $T_1$  to  $T_2$  is a partial one-to-one map from the nodes of  $T_1$  to the nodes of  $T_2$  that preserves the ancestor relations of the nodes. We define the following similarity measure for two homeomorphic trees.

**Definition 1.** Consider two labeled trees  $T_1$  and  $T_2$ , such that  $T_2$  is homeomorphic to  $T_1$ , and let  $\mathcal{M}[T_1, T_2]$  denote a node-to-node, homeomorphism-preserving mapping from  $T_1$  to  $T_2$ . The **Labeled Subtree Homeomorphism score** of  $\mathcal{M}[T_1, T_2]$ , denoted  $LSH(\mathcal{M}[T_1, T_2])$ , is

$$LSH(\mathcal{M}[T_1, T_2]) = \delta(|T_2| - |T_1|) + \sum_{\forall(u,v) \in \mathcal{M}} \Delta[u, v].$$

Correspondingly,

**Definition 2.** The **Approximate Labeled Subtree Homeomorphism (ALSH) problem** is, given two undirected labeled trees  $P$  and  $T$ , and a scoring table which specifies the similarity scores between the label of any node appearing in  $T$  and the label of any node appearing in  $P$ , as well as a predefined node deletion (gap) penalty, to find a homeomorphism-preserving mapping  $\mathcal{M}[P, t]$  from  $P$  to some subtree  $t$  of  $T$ , such that  $LSH(\mathcal{M}[P, t])$  is maximal.

We observe that the ALSH problem on directed multi-source trees is a sparse instance of ALSH on unrooted unordered<sup>1</sup> trees (the fact that the edges are directed reduces the number of possible mappings). Thus, an algorithm for directed multi-source trees can be obtained by extending the *Approximate Subtree Homeomorphism* algorithm of (Pinter *et al.*, 2004) without increasing the algorithm's complexity.

This algorithm combines the node-to-node similarity measures with the topological distance between the pattern and the text to produce a single, comprehensive score expressing how close they are to each other.

## The Alignment Algorithm

The alignment algorithm employs a bottom-up dynamic programming approach and computes optimal alignments between  $P$  and any homeomorphic subtree  $t$  of  $T$ , which maximizes the *LSH* score between  $P$  and  $t$ . It is based on the close relationship between subtree homeomorphism and weighted assignments in bipartite graphs (see the code in Procedure *ComputeAlignmentScores*). The ALSH problem is recursively translated into a collection of smaller ALSH problems, which are solved using weighted assignment algorithms. This approach yields an  $O(m^2n/\log m + mn \log n)$  algorithm for solving ALSH on directed multi-source trees, where  $m$  and  $n$  are the number of vertices in  $P$  and  $T$ , respectively. For simplicity of presentation, we first describe the basic ALSH algorithm for rooted unordered trees (where both the pattern and text trees are rooted and edge direction is ignored). This is done in the next section, where the basic algorithm flow is described and exemplified. Then, in the subsequent section, we show how to extend the basic algorithm to multi-source trees (where both the pattern and text trees are unrooted and edge direction is taken into account).

**Algorithm Flow.** Let  $T^r = (V_T, E_T, r)$  be the text tree which is rooted in  $r$ , and  $P^{r'} = (V_P, E_P, r')$  be the pattern tree which is rooted in  $r'$ , respectively. Let  $p_u^{r'}$  denote a subtree of  $P^{r'}$  which is rooted in node  $u$  of  $P^{r'}$ , and  $t_v^r$  denote a subtree of  $T^r$  which is rooted in node  $v$  of  $T^r$ . Let  $y_1, \dots, y_{c(v)}$  be the children of  $v \in T^r$ , and let  $x_1, \dots, x_{c(u)}$  be the children of  $u \in P^{r'}$ . (Note that  $c(u) \leq c(v)$ , as no deletions are allowed from the pattern.) We define  $AlignmentScores[u \in V_P, v \in V_T]$  as follows.

**Definition 3.** For each node  $v \in V_T$  and for each node  $u \in V_P$ ,  $AlignmentScores[u, v]$  is the maximal *LSH* similarity score between any subtree  $p_u^{r'}$  of  $P^{r'}$  and a corresponding homeomorphic subtree  $t_v^r$  of  $T^r$ , if such exists. Otherwise,  $AlignmentScores[u, v]$  is  $-\infty$ .

The computation of  $AlignmentScores[u, v]$  is done recursively, in a *postorder* traversal of  $T^r$ . First,  $AlignmentScores[u, v]$  are computed for all leaf nodes of  $T^r$  and  $P^{r'}$ . Next,  $AlignmentScores[u, v]$  are computed for each node pair ( $u \in V_P, v \in V_T$ ), based on the values of the previously computed scores for all children of  $u$  and  $v$  as follows. Let  $u$  be a node of  $P^{r'}$  with children  $x_1, \dots, x_{c(u)}$  and  $v$  be a node of  $T^r$  with children  $y_1, \dots, y_{c(v)}$ . After computing  $AlignmentScores[x_i, y_j]$  for  $i = 1, \dots, c(u)$  and  $j = 1, \dots, c(v)$ , a bipartite graph  $G$  is constructed with bipartition  $X$  and  $Y$ , where  $X$  is the set of children of  $u$ ,  $Y$  is the set of children of  $v$ , and each node in  $X$  is connected to each node in  $Y$ . Edge  $(x_i, y_j)$  of  $G$  is annotated with weight  $AlignmentScores[x_i, y_j]$  (see Figure 2).

<sup>1</sup> An *unrooted tree* is an undirected, acyclic, connected graph. A tree is said to be *ordered* if the relative order of its subtrees in each node is fixed. Otherwise, a tree is *unordered*.

$AlignmentScores[u, v]$  is then computed, using procedure  $ComputeAlignmentScores(u, v)$  (see the code in Figure 3) as the maximum between the following two terms:

1. The node-to-node similarity value  $\Delta[u, v]$ , plus the sum of the weights of the matched edges in the maximal assignment over  $G$ . Recall that this term is only computed if  $c(u) \leq c(v)$ .
2. The weight  $AlignmentScores[u, y_j]$  for the comparison of  $u$  and the best scoring child  $y_j$  of  $v$ , updated with the penalty for deleting  $v$ .

Figure 2 exemplifies a single call to procedure  $ComputeAlignmentScores$  for aligning a pair of subtrees. Upon the call to procedure  $ComputeAlignmentScores(u, v)$ , the scores for comparing each subtree of  $u$  (rooted at either  $x_1$  or  $x_2$ ) with each subtree of  $v$  (rooted at  $y_1, y_2$  or  $y_3$ ) have already been computed and stored in the corresponding cells of the dynamic programming table  $DP$ . Note that the score for comparing  $x_1$  with  $y_3$  is 1, obtained as the sum of two terms: a score of 2 for aligning an “a” with an “A” plus a penalty of -1 for deleting a “D” (see the scoring table  $\Delta$  in Figure 1). Similarly, the score for aligning  $x_2$  with  $y_3$  is 1, which is the score of aligning a “b” with a “D”. (By definition of subtree homeomorphism, there is no penalty for deleting the subtree of  $y_3$  that is labelled with an “A”). In the same manner, the best score for aligning of  $x_1$  with  $y_2$  is -3, obtained by matching an “a” directly to a “C” and deleting the subtrees of  $y_2$  which are labeled with an “F”. Similarly, the score for aligning  $x_2$  with  $y_2$  is -2. All other subtree pairs are composed of two leaf nodes and therefore their score has been previously set by direct lookup in the scoring table  $\Delta$ . Procedure  $ComputeAlignmentScores(u, v)$  constructs the bipartite graph  $G$  shown in Figure 2 with bipartition  $X$  and  $Y$ , where  $X = \{x_1, x_2\}$  is the set of children of  $u$ ,  $Y = \{y_1, y_2, y_3\}$  is the set of children of  $v$ , and each node in  $X$  is connected to each node in  $Y$ . The weight of an edge connecting vertices  $x_i$  and  $y_j$  is set to the previously computed value  $DP[x_i, y_j]$  which is the score for the alignment of the subtree rooted at  $x_i$  with the subtree rooted at  $y_j$ .

The value  $DP[u, v]$  is then computed as the maximum between the following two terms:

1. The node-to-node similarity value  $\Delta[u, v] = +2$ , plus the assignment score for  $G$  which is also +2, obtained by matching  $x_1$  with  $y_3$  and  $x_2$  with  $y_1$ . This term yields a total score of +4.
2. The score for comparing node  $u$  with the best child of  $v$  is -7 and the penalty for deleting node  $v$  is -1, so this term yields a total score of -8.

Since the term contributed by the bipartite matching yields a score which is better than the score suggested by deleting node  $v$ , entry  $DP[u, v]$  will finally be set to the value of +4.

**Extensions to Directed Multi-Source Trees.** The Approximate Labeled Subtree Homeomorphism algorithm described above can be easily extended to support unrooted, unordered trees as follows. Let  $T = (V_T, E_T)$  and  $P = (V_P, E_P)$  be two unrooted trees. The ALSH between  $P$  and  $T$  could be computed in a naive manner as follows. Select an arbitrary node  $r$  of  $T$  to obtain the rooted tree  $T^r$ . Next, for each node  $u \in P$  compute the rooted ALSH between  $P^u$  and  $T^r$ . Clearly, such a strategy entails the computation of alignments of subtree pairs  $(p_u^r, t_v^r)$  for each  $u \in P$  and  $v \in T$ . We refer the interested reader to Pinter et al., 2004 for a more sophisticated variation of this algorithm.

We next turn to handle multi-source trees. Such trees are DAGs whose underlying structure is an unrooted, unordered tree, and therefore alignments corresponding to potential mappings between subtree pairs  $(p_u^r, t_v^r)$ , such that  $u \in P$  and  $v \in T$ , will be considered. However, here we filter-out subtree alignments that map together edges of conflicting direction. For example, consider the potential mapping between subtrees  $t_v^r$  and  $p_u^r$  in Figure 1. The following hierarchy is defined on the neighbors of node  $u$  in  $p_u^r$ : node  $r'$  is denoted the “parent” of  $u$  while nodes  $x_1$  and  $x_2$  are denoted the “children” of  $u$ . Similarly, in  $t_v^r$  node  $r$  is the parent of node  $v$  and nodes  $y_1$  and  $y_2$  are the children of  $v$ . Note that node  $u$  has two incoming edges to its children  $x_1$  and  $x_2$  in  $p_u^r$ , while in  $t_v^r$  node  $v$  has one incoming edge from child  $y_2$  and one outgoing edge to child  $y_1$ . When computing ALSH for multi-source trees, a mapping between two nodes is forbidden if the directions of the edges connecting each node to its designated parent disagree. Furthermore, by definition of subtree homeomorphism, each child of  $u$  must be mapped to a child of  $v$ , and therefore the algorithm for ALSH on multi-source trees will set the alignment score for the subtree pair  $(p_u^r, t_v^r)$  to  $-\infty$ . Thus, the additional edge-direction information in multi-source trees restricts the number of possible mappings by adding the requirement that both the number of the incoming edges of  $u$  and the number of outgoing edges of  $u$  must be smaller than or equal to the numbers of the incoming and outgoing edges of  $v$ , respectively.

As for legitimate subtree mappings, the weighted bipartite matching computation is updated as follows to utilize the edge-direction information in multi-source trees: consider the bipartite graph  $G = \{X \cup Y, E\}$ , where  $X$  denotes the children of  $v$  in  $t_v^r$  and  $Y$  denotes the children of  $u$  in  $p_u^r$ . A vertex  $(x_i, y_j)$  will now be included in  $E$  if and only if the direction of the edge connecting  $x_i$  to  $u$  is similar to the direction of the edge connecting  $y_j$  to  $v$ . Therefore, we get a sparse bipartite graph, which could actually be split into two separate, smaller bipartite graphs: one corresponding to matchings of incoming-edge neighbors of  $u$  and  $v$ , and the other for matching outgoing-edge neighbors.

## Application to Metabolic Pathway Analysis

In this section we first describe our method and data sources and then analyze their significance.

**Metabolic Data Sets.** Metabolic pathways of *E. coli* were extracted from the EcoCyc (Karp *et al.*, 2004) database and metabolic pathways of the yeast *S. cerevisiae* were extracted from SGD (Christie *et al.*, 2004). Both databases combine automatic pathway creation based on gene annotations as well as manual curation. Our dataset contained all pathways composed of two reactions or more that appear in these databases for these organisms (113 for *E. coli* and 151 for *S. cerevisiae*).

Note that the text graph rarely contains pathways whose underlying undirected graph is cyclic. In the seldom case of directed cycles (fewer than 10 per organism), we generated alternative multi-source trees that cover all the possible cycle-splitting variations. In the special case of DAGs which cannot be cast as multi-source trees, duplication and splitting is performed on those vertices where two ingoing edges meet. This fits well with biology as the distinct paths correspond to alternative metabolic pathways.

**Alignment Scoring.** Similarly to sequence alignment, the suggested notion of pathway alignment is based on edit operations that include node substitution and node deletion (the latter relating only to the text). Alignment scoring is composed of node substitution scores that are rated by a label substitution table, and node deletion scores modelling gaps in the pattern which entail a fixed penalty. Below we describe the scoring scheme used for these two operations.

To build a label substitution table we associated each enzyme with its EC (Enzyme Commission) classification - a numbering system consisting of four sets of numbers that categorize the type of the catalyzed chemical reaction. Since an EC classification is functional, enzymes with similar EC classifications are functional homologues, but do not necessarily possess any sequence similarity. The actual values of the label substitution table were determined according to the following definition from (Tohsato *et al.*, 2000):

**Definition 4.** For an enzyme class  $h$ ,  $C(h)$  denotes the number of enzymes whose classes are included under  $h$ .  $I(h)$ , the **information content** of  $h$ , is defined as

$$I(h) = -\log_2 C(h)$$

For two enzymes  $e_i$  and  $e_j$ , if their lowest common upper class is  $h_{ij}$ , then we consider  $I(h_{ij})$  to express the similarity between  $e_i$  and  $e_j$ .

Note that we look for the smallest common subtree that contains both enzymes. Therefore if two enzymes are far apart in the EC classification their smallest common subtree will contain many leaves and thus their similarity level will be low. Otherwise their smallest common subtree will contain only a few leaves and their similarity level will be higher. Hence  $I(h)$  increases with the similarity.

The node deletion score (*i.e.* gap penalty) reflects the tradeoff between a gap and a mismatch. As the gap penalty increases, the algorithm tends to match distant enzymes to avoid gaps. Conversely, a gap penalty of zero enables alignments of evolutionary remote pathways, where only bits of the pathways are conserved, to score highly. As different values may suit different needs our tool enables users to set this parameter per execution.

**Statistical Significance of Alignments.** The statistical significance of each alignment is based on  $p$ -value calculation. The  $p$ -value of an alignment of a pathway query with score  $s$  was computed by executing the same query against 100 random pathway graphs, and counting the fraction of graphs containing an alignment that received score  $s$  or higher. A random pathway graph is a graph containing the same set of nodes and the same number of edges as the original graph, such that the degree of each node in the random graph is equal to its degree in the original graph. Random pathway graphs were generated from the original pathway graph by a long series of random edge switches, as described in (Maslov and Sneppen, 2002).

The  $p$ -value cutoff used in our analysis is 0.01. We denote pathway pairs with at least one statistically significant alignment between them as *significantly aligned pathway pairs*. To assess whether the number of significantly aligned pathway pairs in the inter-species comparison and in the intra-species comparison deviate significantly from the number expected by pure chance at a cutoff of 0.01, we used the exact binomial test ( $k, n, p$ ) per comparison. This test computes the probability of having at least  $k$  successes in  $n$  Bernoulli experiments with probability  $p$  for success. Here  $k$  is the number of significantly aligned pathway pairs,  $n$  is the total number of aligned pathway pairs, and  $p$  is 0.01. This test was performed using the *R* project for Statistical Computing (<http://www.r-project.org>).

## Implementation Details

The algorithm was implemented as a prototype program using a combination of C++ code and a Java-based GUI in order to allow web applet-based usage. It runs on any Intel Pentium-based computer under the Microsoft Windows operating system (Version 2000 and higher). It does not require any special purpose hardware or other licensed software.

For each query, the program reports the 5 best matches per pathway, sorted by score and statistical significance, and produces an HTML file that graphically superimposes the query upon the aligned metabolic pathway. As for determining the gap penalty, manual inspection of the data revealed that most pathway alignments include at most one gap in a row. Considering that the worst mismatch in the EC classification is scored  $-8.17$ , we set the default gap penalty to  $-3$ , which allows for two consecutive gaps followed by a mismatch between closely-related enzymes, or for one gap and a mismatch between more distant enzymes.

## RESULTS

We applied our approach to the genome-scale metabolic networks of the bacterium *E. coli* and the yeast *S. cerevisiae*, as these are the two extensively studied model organisms representing the prokaryotic and eukaryotic kingdoms, respectively. We ran all-against-all alignments, namely taking each metabolic pathway as a query and aligning it against all other pathways in our dataset. We also used our tool to data-mine the pathway database with a meta-pathway query.

The runtime of the all-against-all benchmark, where query sizes ranged from 2 to 41 nodes, was measured. The entire process, including the I/O overhead of reading the pathways and recording the alignment information for successful matches, took 3.66 hours to complete on a regular desktop machine (Pentium 4, 2.6GHz clock, 512MB RAM). This yields an average of 47 seconds per query.

Below we describe results relating to both inter- and intra-species alignments, and conclude by demonstrating the power of metapathway queries in biologically relevant scenarios.

### Inter-species Alignments

We performed all possible alignments between the 113 *E. coli* pathways and the 151 *S. cerevisiae* pathways. This analysis resulted in 610 pathway pairs that had at least one statistically significant alignment between them ( $p \leq 0.01$ ). This number was statistically significantly greater than the randomly expected number of  $113 \times 151 \times 0.01 = 171$  pathway pairs (applying the exact binomial test to 610 pathways yields  $p < 2.2e^{-16}$ ). The significant alignments span most types of metabolic pathways, such as amino acid biosynthesis and fatty acid degradation, as 63% of the *E. coli* pathways and 66% of the *S. cerevisiae* pathways had at least one statistically significantly aligned pair-mate from the other species. In order to evaluate more carefully the degree of conservation between the metabolic networks of the two species we examined the alignments of the analogous metabolic pathways in *E. coli* and *S. cerevisiae*. Out of the 80 analogous pathways 62 pathways were found to be statistically significant ( $p \leq 0.01$ ). This implies that, despite the evolutionary distance between *E. coli* and *S. cerevisiae*, a considerable fraction of their metabolic networks is conserved.

The conservation between species is not limited to small pathways, as demonstrated by the alignment of the analogous metabolic pathways of phenyl-alanine, tyrosine, and tryptophan biosynthesis in *E. coli* and *S. cerevisiae* ( $s = -4.28$ ,  $p < 0.01$ ). This pathway consists of 17 enzymes arranged in a star-like topology, turning the substrate erythrose-4 phosphate into one of the three amino acids phenyl-alanine, tyrosine, or tryptophan (see Figure 4.1). In spite of its size the pathway is almost identical between the two species, implying a common ancestral pathway. Indeed, it has been suggested that the major amino acid biosynthesis pathways were established before ancient organisms diverged into the

three kingdoms of Archaea, Bacteria, and Eukaria (Hochuli et al., 1999).

The analogous pathways of phenyl-alanine, tyrosine, and tryptophan biosynthesis in *E. coli* and *S. cerevisiae* provide a stimulating example for the power of our tool in discovering interesting biological phenomena. Inspection of their alignment reveals that the two pathways are identical except for a single mismatch within an intermediate enzyme in the biosynthesis of tyrosine, carried out by TyrA in *E. coli* (labeled 1.3.1.13) and Tyr1 in *S. cerevisiae* (labeled 1.3.1.12). The two enzymes catalyze almost identical reactions however TyrA uses NAD<sup>+</sup> as an acceptor while the *S. cerevisiae* enzyme uses NADP<sup>+</sup> instead. Intriguingly, upon aligning their protein sequences using BLAST no significant sequence similarity was found between the two enzymes. The two enzymes appear to be true functional orthologs, resulting either from convergent evolution where non-homologous proteins converged to a similar function, or else from divergent evolution that changed the protein sequences but maintained their function. This example asserts our choice of EC classification as our scoring scheme since only by using a functional classification, in contrast to sequence based classification, could such a phenomenon be detected.

Gaps in the alignment of two pathways may hint to additional intriguing evolutionary phenomena. An example is the gap found upon comparing homoserine to methionine biosynthesis in *E. coli* vs. *S. cerevisiae* ( $s = -13.15$ ,  $p < 0.01$ ), depicted in Figure 4.2. In *S. cerevisiae* this pathway consists of a chain of three reactions catalyzed by three different enzymes. In *E. coli* the pathway consists of a chain of four reactions catalyzed by four different enzymes. The middle reaction in *S. cerevisiae*, catalyzed by Met17, is analogous to the succession of the two middle reactions in *E. coli*, catalyzed by MetB and MetC. Biologically, this implies that the functionality of Met17 in *S. cerevisiae* is comparable to the combined functionality of two enzymes MetB and MetC in *E. coli*. Moreover, all three enzymes are sequence homologues. This may hint to an interesting case of either gene fusion in *S. cerevisiae* or gene duplication in *E. coli*. Further investigation is needed to uncover the biological scenario that led to this incident; however, the finding that these enzymes participate in a common metabolic pathway provides a first step in this direction.

### Intra-species Alignments

Intra-species alignments may provide researchers with the ability to trace the evolution of metabolism within a species. For example, the finding that pathways within a species resemble each other may imply that they arose during evolution due to instances of gene duplication followed by divergence. To demonstrate the abilities of our tool we executed all-against-all intra-species queries, where each pathway was aligned against all other pathways within the same species.

The all-against-all alignments in *E. coli* and in *S. cerevisiae* resulted in 187 significantly aligned pathway pairs in *E. coli*, and 262 such pairs in *S. cerevisiae* ( $p \leq 0.01$ ). The number of such pathways in *E. coli* is statistically significantly greater than the randomly expected number of  $113 \times 112 \times 0.01 = 127$  pathway pairs (yielding  $p < 4.2e^{-07}$  using the exact binomial test). The same computation for *S. cerevisiae* resulted in  $151 \times 150 \times 0.01 = 227$  expected pathway pairs, and the corresponding statistical significance of our result is  $p < 0.02$ . Statistically significant alignments were found for 66% of the pathways in *E. coli* and 62% of the pathways in *S. cerevisiae*.

The pathways of biosynthesis of the amino acids valine, leucine, and isoleucine (see Figure 5.1) provide an example for the power of intra-species alignments. The three amino acids belong to the class of hydrophobic amino acids. Valine and leucine are synthesized from the same substrate and share most of the pathway; isoleucine is synthesized from a different substrate. The intra-species alignments revealed that valine and isoleucine have identical biosynthesis pathways ( $s = 0$ ,  $p < 0.01$ ) in both *E. coli* and *S. cerevisiae*, and even employ the same set of enzymes. This substantiates the hypothesis that the biosynthesis of the three amino acids arose from a common ancestral amino-acid biosynthesis pathway (Klipcan and Safro, 2004). Moreover, the degradation of the three amino acids, similarly to their biosynthesis, involves identical enzymes. Hence the entire metabolism of these three amino acids seems to stem from a single ancestral pathway.

### MetaPathway Queries

So far we have discussed cases in which a user provides a specific metabolic pathway as a query. However in some cases a user may query the tool using only a partial skeleton of a certain pathway. The output of the pathway alignment tool may then identify the entire pathway scheme. One approach that is likely to benefit from this option is metabolic pathway tinkering (Newgard, 2002), where metabolic pathways are redirected and re-engineered in order to supply certain products. To answer such needs and others we provide the possibility to form and pose a meta-pathway query.

A meta-pathway query is a pattern containing the essential enzymes as nodes and a suggested structure of their (not necessarily direct) interactions. Note that in our model no deletions are allowed in the pattern, hence it is important for all putative enzymes to appear in the pattern. Furthermore, our notion of homeomorphism allows us to represent indirect interactions as single edges in the pattern; the gap penalties must be adjusted when using the algorithm in this mode so as to increase the chances of finding chain reactions.

Meta-pathway queries may be of significant value in two likely scenarios. The first is when a user wishes to discover if two or more enzymes of interest are metabolically connected. This may serve to understand the effect of a mutation

in one enzyme on the performance of another, for example upon analyzing functional profiles of gene-deletion mutants (Giaever *et al.*, 2004). A second scenario is when a user has limited knowledge of a certain pathway, and would like to uncover the entire pathway.

An example for the latter is given in Figure 6, where the query consisted of a hub enzyme and its adjacent enzymes (see Figure 6.1). The tool reported two significant alignments (see Figure 6.2), the *E. coli* allantoin degradation pathway and the *S. cerevisiae* ureide degradation pathway. Both pathways degrade the same substrate to three different products in *S. cerevisiae* and to two of these three in *E. coli* (note that the gap penalty was set to zero to allow for maximal degrees of freedom during the search). The ability to detect these related but not identical pathways through a common core demonstrates the power of meta-queries where knowledge of the entire pathway and its homologues is lacking.

## DISCUSSION

We have presented a new formulation for an emerging problem in bioinformatics, namely the need to find pathway patterns in larger metabolic pathway texts. Our formulation includes a score that combines both topological as well as naming similarities in a comprehensive manner. Moreover, this formulation gives rise to efficient algorithms (Pinter *et al.*, 2004) that are able to deal with more complicated network structures than have been handled to date.

We have implemented these algorithms and embodied them in a working tool that can be effectively used by life science researchers. Our new tool yields more comprehensive queries than those supported by previous tools, which were restricted to chain topology and therefore could not capture the more complex, tree-like homologies. Furthermore, we demonstrated the utility of our tool by analyzing a large number of metabolic pathways of *E. coli* and *S. cerevisiae*, thus revealing new biological insights into pathway evolution. These results in themselves are of interest and open the way to similar studies.

We intend to extend the tool to more general network topologies, such as directed acyclic graphs, graphs with limited tree-width, and graphs that have simple cycle decompositions. Another open issue is to incorporate a variable scoring scheme, to *e.g.* represent affine gap penalties. We also propose to analyze hypergraphs: hyperedges can be used to represent reactions that involve several enzymes. Finally, we plan to make our tool available through the emerging platforms for biological data exchange, providing the necessary interfaces.

## ACKNOWLEDGEMENTS

We thank Elad Ben-Yosef and Moshe Itzhaki for their dedicated work in coding the algorithm, and Hanah Margalit,

Roded Sharan, and the anonymous referees for their helpful comments. This work was supported by the Aly Kaufman Post-Doctoral Fellowship to M. Z.-U.; the Bar-Nir Bergreen Software Technology Center of Excellence to R. Y. P. and M. Z.-U.; the Yeshaya Horowitz Association through the Center for Complexity Science and the Planning and Budgeting Committee of the Council for Higher Education in Israel to E. Y.-L.

## REFERENCES

- Garey,M.R. and Johnson,D.S. (1979) *Computers and Intractability*. Freeman, San Francisco.
- Christae,K.R. et al. (2004) Saccharomyces Genome Database (SGD) provides tools to identify and analyze sequences from Saccharomyces cerevisiae and related sequences from other organisms. *Nucl. Acids Res.*, **32**, D311-D314.
- Chung,M.J. (1987)  $O(N^{2.5})$  time algorithms for the subgraph homeomorphism problem on trees. *J. Algorithms*, **8**, 106-112.
- Giaever,G., Chu,A.M., Ni,L., Connely,C., Riles,L. et al. (2004) Functional profiling of the saccharomyces cerevisiae genome. *Nature*, **418**, 387-91.
- Hochuli,M., Palzelt,H., Oesterheld,D., Wuthurich,K. and Szyper-sky,T. (1999) Amino acid biosynthesis in the halophilic archaeon haloarcula hispanica. *J. Bacteriol.*, **81**, 3226-37.
- Kanehisa,M. and Goto,S. (2000) KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.*, **28**, 27-30.
- Karp,P.D., Arnaud,M., Collado-Vides,J., Ingraham,J., Paulsen,I.T. and Saier,M.H. Jr. (2004) The e. coli EcoCyc Database: No Longer Just a Metabolic Pathway Database. *ASM News*, **70**, 25-30.
- Kelley,B.P., Sharan,R., Karp,R.M., Sittler,T., Root,D.E., Stockwell,B.R. and Ideker,T. (2003) Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *PNAS*, **100**, 11394-11399.
- Koyutürk,M., Grama,A. and Szpankowski,W. (2004) An efficient algorithm for detecting frequent subgraphs in biological networks. *Bioinformatics*, **20**, i200-i207.
- Kilpelainen,P. and Mannila,H. (1995) Ordered and unordered tree inclusion. *SIAM J. Comput.*, **24**, 340-356.
- Klipcan,L. and Safro,M. (2004) Amino acid biogenesis, evolution of the genetic code and aminoacyl-trna synthetases. *J Theor Biol*, **228**, 389-96.
- Maslov,S. and Sneppen,K. (2002) Specificity and stability in topology of protein networks. *Science*, **296**, 910-913.
- Matula,D.W. (1968) An algorithm for subtree identification. *SIAM Rev.*, **10**, 273-274.
- Matula,D.W. (1978) Subtree isomorphism in  $O(n^{5/2})$ . *Ann. Discrete Math.*, **2**, 91-106.
- Newgard,C.B. (2002) While tinkering with the beta-cell...metabolic regulatory mechanisms and new therapeutic strategies. *Diabetes.*, **51**, 3141-50.
- Ogata,H., Fujibuchi,W., Goto,S. and Kanehisa,M. (2000) A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. *Nucleic Acids Research*, **28**, 4021-4028.
- Pinter,R.Y., Rokhlenko,O., Tsur,D. and Ziv-Ukelson,M. (2004) Approximate labelled subtree homeomorphism, *Proceedings of 15th Annual Symposium of Combinatorial Pattern Matching*, Lecture Notes in Computer Science, **3109**. Springer-Verlag, pp. 59-73.
- Reyner,S.W. (1977) An analysis of a good algorithm for the subtree problems. *SIAM J. Comput.*, **6**, 730-732.
- Schreiber,F. (2003) Comparison of metabolic pathways using constraint graph drawing. *Proceedings of the Asia-Pacific Bioinformatics Conference (APBC'03)*, Conferences in Research and Practice in Information Technology, **19**. pp. 105-110.
- Shamir,R. and Tsur,D. (1999) Faster subtree isomorphism. *Journal of Algorithms*, **33** 267-280.
- Tohsato,Y., Matsuda,H. and Hashimoto,A. (2000) A multiple alignment algorithm for metabolic pathway analysis using enzyme hierarchy. *Proc. 8th International Conference on Intelligent Systems for Molecular Biology (ISMB 2000)*, pp. 376-383.
- Valiente,G. (2003) Constrained tree inclusion. *Proceedings of 14th Annual Symposium of Combinatorial Pattern Matching*, Lecture Notes in Computer Science, **2676**. pp. 361-371.

---

### Procedure *ComputeAlignmentScores*( $u, v$ )

---

**Input:** A DP table with all values up to cell  $(u, v)$  already set. A Label-to-Label Scoring Table  $\Delta$ .

**Output:** The score to be set to entry  $(u, v)$  of the DP table.

$k$  : the out-degree of node  $u$ ;

$l$  : the out-degree of node  $v$ ;

**if**  $k > l$  **then**

return  $-\infty$ ;

**else**

$G$  : a bipartite graph with node bipartition  $X$  and  $Y$ ;

$X$  : the set of children  $\{x_1, \dots, x_k\}$  of  $u$ ;

$Y$  : the set of children  $\{y_1, \dots, y_l\}$  of  $v$ ;

node  $x_i \in X$  is connected to node  $y_j \in Y$  via an edge

whose weight  $w(x_i, y_j)$  is set to  $DP[x_i, y_j]$ ;

$AS(G)$ : the weighted assignment score of  $G$ ;

$AS(G) \leftarrow \max_{(i,j) \in M} DP[x_i, y_j]$

where  $M$  is a maximum matching;

**end**

$BestChild(u, v)$ : the child of node  $v$  whose ALSH score with  $u$  is the highest;

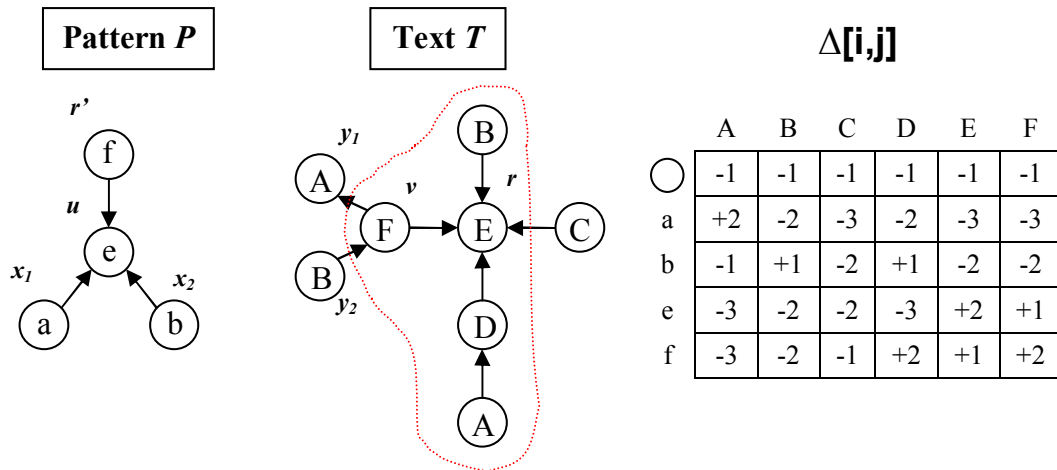
$BestChild(u, v) \leftarrow \max_{j=1}^l DP[u, y_j]$ ;

$\delta$  : the deletion penalty from  $\Delta$ ;

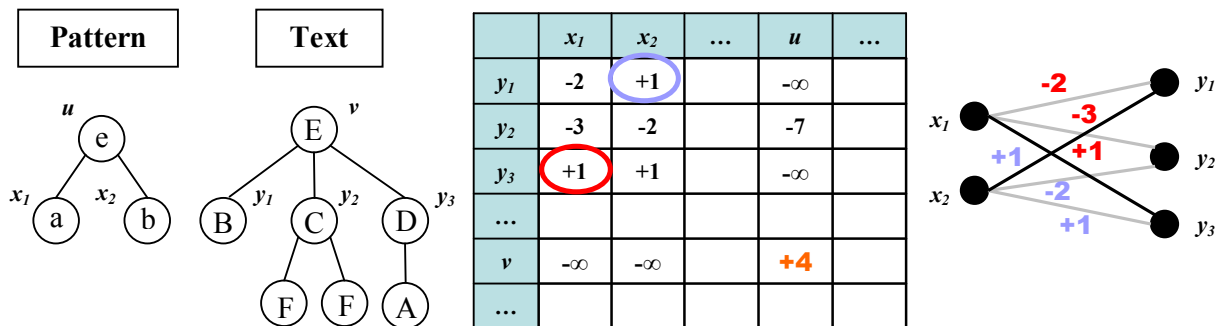
return  $\max\{\Delta[u, v] + AS(G), BestChild(u, v) + \delta\}$ ;

---

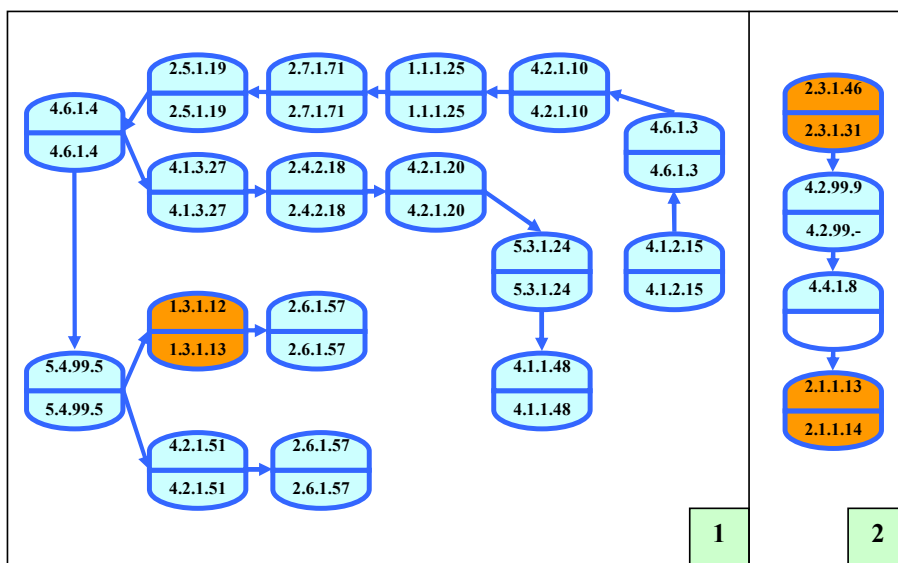
**Fig. 3.** Procedure *ComputeAlignmentScores*( $u, v$ )



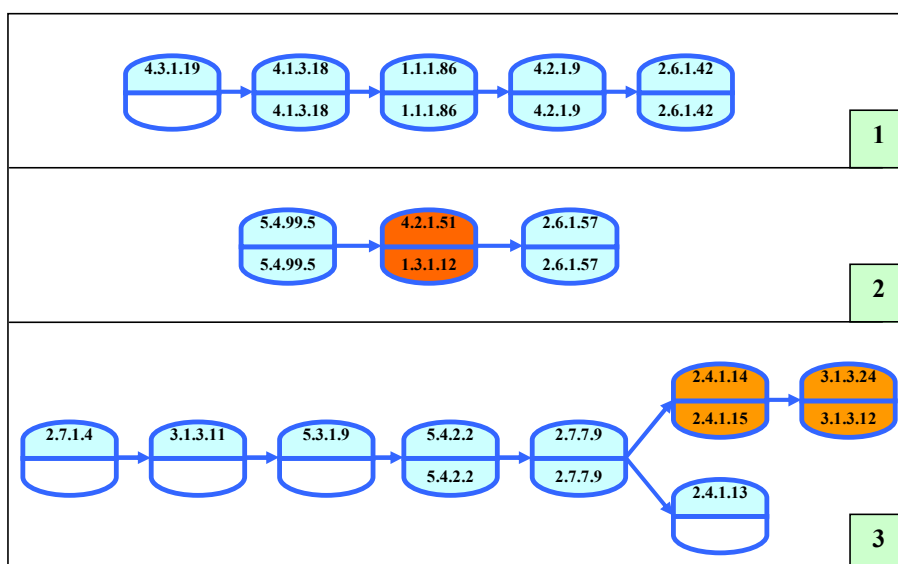
**Fig. 1.** Approximate Labeled Subtree Homeomorphism. For each node, the label is written inside the circle and the variable name assigned to the node is written externally. The node-label similarity scores are specified in Table  $\Delta$ . Note that deletion score is set to  $-1$ . A subtree in the text that is homeomorphic to the pattern is circled by the dashed line. The LSH score for this alignment is 7.



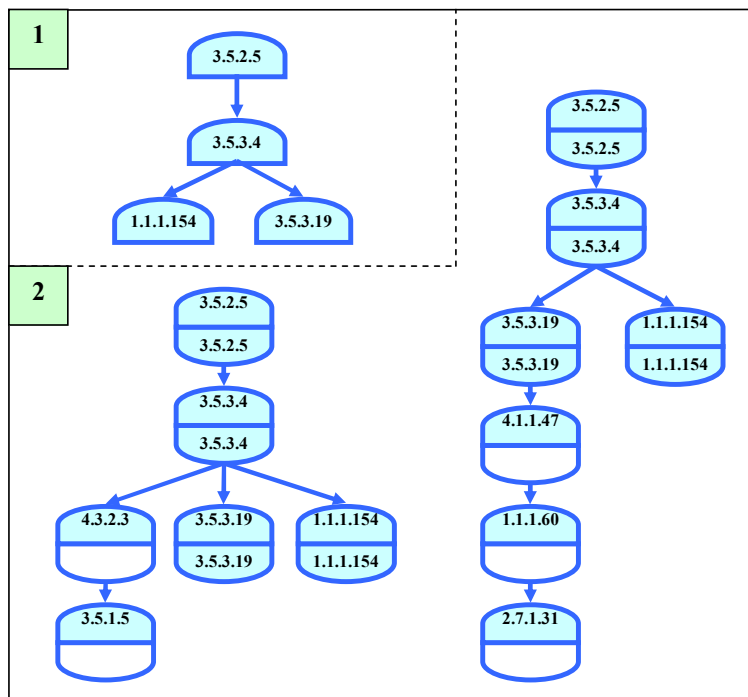
**Fig. 2.** The work done by the ALSH algorithm during the alignment of the subtree  $P^u$  with the subtree  $T^v$ . The score for entry  $(u, v)$  of the above DP table is computed via the corresponding weighted bipartite graph. The node-label similarity scores used in this example are specified in Table  $\Delta$  of Figure 1.



**Fig. 4.** The top-scoring *inter* species alignments. Each node represents a match: the upper part represents the query enzyme, and the lower part represents the text enzyme. Color shades reflect enzyme homology. [1] The phenylalanine, tyrosine and tryptophan pathway of *E.coli* vs. *S. cerevisiae* ( $s = -4.28, p < 0.01$ ). [2] homoserine and methionine biosynthesis of *E. coli* vs. *S. cerevisiae* ( $s = -13.15, p < 0.01$ ).



**Fig. 5.** The top-scoring *intra* species alignments. [1] The isoleucine vs. valine biosynthesis pathways of *S. cerevisiae* ( $s = 0, p < 0.01$ ) alignment. [2] The trehalose anabolism pathways of *S. cerevisiae* vs. sucrose biosynthesis pathway of *S. cerevisiae* ( $s = -9.58, p < 0.01$ ). [3] The tyrosine biosynthesis of *E.coli* vs. the phenylalanine biosynthesis of *E. coli* ( $s = -8.23, p < 0.01$ ).



**Fig. 6.** The meta-pathway query alignment. [1] A meta-query. [2] The alignment of a meta-query with the ureide degradation pathway of *S. cerevisiae* (left,  $s = 0, p < 0.01$ ) and with the alantoine degradation pathway of *E. coli* (right,  $s = 0, p < 0.01$ ).