Inference of pathways from metabolic networks by subgraph extraction

Karoline Faust[1], Jérôme Callut[2], Pierre Dupont[3], and Jacques van Helden[4]

[1,4]Laboratoire de Bioinformatique des Génomes et des Réseaux (BiGRe), Université Libre de Bruxelles, Campus Plaine - CP263, Boulevard du Triomphe, 1050 Bruxelles, Belgium.
[2,3]UCL Machine Learning Group, Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium.
{kfaust,jacques.van.helden}@ulb.ac.be
jcal@info.ucl.ac.be
pierre.dupont@uclouvain.be

Abstract. In this work, we present different algorithmic approaches to the inference of metabolic pathways from metabolic networks. Metabolic pathway inference can be applied to uncover the biological function of sets of co-expressed, enzyme-coding genes.
We compare the kWalks algorithm based on random walks and an alternative approach relying on k-shortest paths. We study the influence of various parameters on the pathway inference accuracy, which we measure on a set of 71 reference metabolic pathways. The results illustrate that kWalks is significantly faster and has a higher sensitivity but the positive predictive value is better for the pair-wise k-shortest path algorithm. This finding motivated the design of a hybrid approach, which reaches an average accuracy of 72% for the given set of reference pathways.

Key words: metabolic pathway inference, kWalks, k-shortest paths

1 Introduction

The products of co-expressed genes are often involved in a common biological function. In particular, the metabolic response to nutrients is generally regulated at multiple levels, including transcriptional. One approach to understand the function of co-expressed enzymes is to uncover the biological pathways in which they participate. This is usually achieved by mapping reactions associated to enzyme-coding genes on pre-defined metabolic pathways, i.e. [5, 11]. However, this approach does not deal well with transverse pathways (a set of reactions mapping to several pathways) and it fails if reactions belong to a pathway not yet included in the pre-defined set of reference pathways. Another strategy has been to infer metabolic pathways by finding the shortest paths between two reactions (e.g. those catalyzed by two co-expressed enzymes). One problem with this approach is the presence of compounds involved in a large number of reactions (co-factors and side-compounds like H_2O, ATP, NADPH), which tend to be used

as shortcuts to connect any pair of nodes. Thus, a naive path finding approach results in biochemically invalid pathways, which contain co-factors or side compounds as intermediates. In our previous work, we tested different strategies to overcome this problem. First, we excluded a selected subset of highly connected compounds from the network [12, 13]. However, the choice of the compounds to be excluded is an issue, since some among the highly connected compounds participate in pathways as valid intermediate (e.g. purine nucleotide biosynthesis). We therefore introduced weighted networks [3, 4], in order to penalize highly connected compounds without excluding them from the graph. This approach yielded satisfactory results for the two-end linear path finding. In the present work, we extend the approach to *multiple-end pathway inference*: taking as input a set of seed reactions, we extract a subgraph that connects "at best" those seed nodes, according to some relevance criteria. The resulting pathway can correspond to an already known pathway, but it can also be a variant or a combination of known pathways, or even a novel pathway.
Pathway inference is as a more flexible alternative to pathway mapping. For instance, it can be used to infer pathways from operons, co-expressed genes or gene fusion events. It may also be applied in metabolic reconstruction in order to suggest possible pathways from genomic data for organisms with unknown metabolism.
We thoroughly evaluated the pathway inference performance of three algorithms: the *pair-wise k-shortest paths algorithm*, the *kWalks* algorithm [1, 6] and a hybrid algorithm that combines the former two.

2 Materials and Methods

2.1 Metabolic graph

In order to infer metabolic pathways, we need to represent metabolic data as a graph. We selected MetaCyc [10], the well-curated tier of BioCyc [2], as our data source, and constructed a bipartite, directed graph from all small molecule entries and their associated reactions contained in the OWL file of MetaCyc (Release 11.0). The resulting graph consists of 4,891 compound nodes and 5,358 reaction nodes. As discussed in [3], the direction of a reaction depends on physiological conditions in an organism (substrate and product concentrations, temperature). Since our graph is composed of data obtained from several hundred organisms, we considered that each reaction can be traversed either in forward or in reverse direction. Consequently, each reaction was represented as a pair of nodes, for the forward and the reverse directions, respectively. To prevent the k-shortest paths algorithm to cross the same reaction twice, forward and reverse direction are mutually exclusive. After this duplication of reaction nodes, we obtain a directed graph with 15,607 nodes and 43,938 edges. From now on, we will refer to this graph as the *MetaCyc* graph.

2.2 Reference pathways

We obtained a selected set of 71 known *S. cerevisiae* pathways from BioCyc (Release 11.0). All pathways in this reference set consist of at least 5 nodes and are included in the largest connected component of the MetaCyc graph. On average, the pathways are composed of 13 nodes and in addition, more than half of them are branched and/or cyclic.

2.3 Algorithms

All algorithms tested here take as input the nodes of interest (termed seed nodes or seeds) as well as a weighted input graph, and return a subgraph that connects the seeds.

Pair-wise k-shortest paths This approach relies on repetitively calling a k-shortest paths algorithm. K-shortest paths algorithms enumerate all simple paths (paths containing each node only once) between a start and an end node in the order of their length. In a weighted graph, paths are listed in the order of their weight.
In the first step of pair-wise k-shortest paths, a k-shortest paths algorithm [7] is called successively on each pair of seed nodes. A k-shortest instead of a shortest paths algorithm is employed to ensure that all lightest paths between a seed node pair are collected. The resulting path sets are stored in a path matrix. The minimal weight between each node pair is stored in a distance matrix. For the undirected MetaCyc graph, these matrices are symmetric. For the directed MetaCyc graph, the reverse paths between two seeds can be obtained by reversing the order of path nodes and their reaction directions.
In the second step of the algorithm, the subgraph is constructed from the path sets, starting with the lightest path set. Step-wise, more path sets are merged with the subgraph in increasing order of their weight. The process stops if either all seeds belong to one connected component of the subgraph or all path sets have been merged with the subgraph. The resulting subgraph represents the inferred pathway.
This algorithm is time-consuming, since the number of calls to the k-shortest paths algorithm increases quadratically with the seed node number.

kWalks The key idea of kWalks is that some edges in the input graph are more relevant than others to connect the seed nodes. The relevance of an edge is measured as the expected number of times it is visited along random walks connecting seed nodes. These expected passage times can be obtained using basic Markov chain theory [8]. A transition probability matrix P is derived from the adjacency matrix of the graph using simple edge weight normalization. For each seed node x, the submatrix ${}^{x}P$ is defined by considering only the lines and columns of P corresponding to x and all non-seed nodes. The fundamental matrix ${}^{x}N = (I - {}^{x}P)^{-1}$ contains useful information for computing the desired

expectation. The entry ${}^{x}N_{xi}$ gives the number of times node i has been visited during walks starting in node x and ending when any seed node (except x) is reached. The expected number of passage time ${}^{x}E(i,j)$ on an edge $i \rightarrow j$ is obtained by multiplying ${}^{x}N_{xi}$ by the probability P_{ij} of transiting from node i to node j. Finally, the relevance of an edge $i \rightarrow j$ is given by averaging ${}^{x}E(i,j)$ over all seed nodes x. This technique is time-consuming since it relies on matrix inversions, which are generally performed with a cubic time complexity in the number of nodes in the graph.
An alternative approach considers random walks of a bounded length, i.e. only walks up to a prescribed length are allowed. The passage time expectations during such walks can be computed in linear time with respect to the number of graph edges and the maximum walk length using forward-backward recurrences [1, 6]. Moreover, bounding the walk length controls the level of locality while connecting seed nodes, which can be useful for pathway recovery.
Once the edge relevance has been obtained, a subgraph can be extracted by adding edges in the order of their relevance with respect to seed nodes, until either all seed nodes are connected or all edges have been added.
The output of kWalks is a list of edge relevance values. We can replace the original edge weights by these relevances and iterate kWalks by re-launching it on the input graph with updated weights.
In contrast to pair-wise k-shortest paths, the pathways inferred by kWalks may contain branches ending in non-seed nodes. We remove these branches in a post-processing step.

Hybrid approach The hybrid approach combines kWalks with the pair-wise k-shortest paths algorithm. First, kWalks is launched to extract a fixed percentage of the input graph. The final pathway is then extracted from the kWalks subgraph using the pair-wise k-shortest paths algorithm.

2.4 Parameter combinations

The performance of kWalks, pair-wise k-shortest paths and the hybrid approach was evaluated with a number of different parameter values.

Iteration number For the kWalks and hybrid algorithm, we ran 1, 3 or 6 iterations of the kWalk algorithm.

Graph weight We weight the metabolic graph to avoid highly connected compounds. As in our previous work [3, 4], we assign to each compound node its degree as weight (compound degree weight) or use an un-weighted graph for comparison (unit weight). In addition, we test a weighting scheme where compound node weights are taken to the power of two (inflated compound degree weight). Since the pair-wise k-shortest paths and kWalks assume weights on edges rather than nodes, the initial degree-based node weights are transformed

into edge weights by taking the mean of the weights of the nodes adjacent to an edge.

Re-use of kWalks edge relevances In the hybrid approach, we may either use the weights from the input graph or the edge relevances computed by kWalks to weight the extracted subgraph. In addition, when iterating kWalks, we may modify the edge relevances by inflating them (taking them to the power of a positive integer) to increase the difference between relevances. We tested all combinations resulting from these options.

Directionality In order to support reaction reversibility, we represent each reaction by two nodes, one for the direct and one for the reverse direction. In addition, we also constructed an undirected version of the MetaCyc graph, where each reaction is represented by only one node, which is connected to compound nodes by undirected edges.

Fixed subgraph extraction In the hybrid approach, after the last kWalks iteration we extract a subgraph of fixed size from the input graph. The size of this subgraph has been varied from 0.1% to 10% of the edges ranked by relevance. The subgraph obtained by fixed size extraction may consist of more than one component.
The subgraph size optimization has been performed in the directed, compound-weighted MetaCyc graph without iterating kWalks or inflating edge relevances. The input graph weights rather than the edge relevances were fed into the second step of the hybrid algorithm.

2.5 Evaluation procedure

For each pathway, several inferences are tested, with increasing seed node number, in order to test the impact of the seed number on the accuracy of the result. For each of the 71 reference pathways, we first select the terminal reactions as seeds, we infer a pathway that interconnects them, and we compare the nodes of the inferred pathways with those of the annotated pathway. Then, we progressively increase the number of seed reactions by adding randomly selected nodes of the reference pathway, and re-do the inference and evaluation, until all reactions of the pathway are selected as seeds.
We define as one experiment the set of all the pathway inferences performed for a given parameter value combination (e.g. pair-wise k-shortest paths on directed graph with compound node weights). We did 82 such experiments to find the optimal parameter value combination for each algorithm.

Scores The accuracy of an inferred pathway is calculated based on the correspondence between its non-seed nodes and those of the reference pathway. We define as true positive (TP) a non-seed node that is present in the reference as

well as the inferred pathway. A false negative (FN) is a non-seed node present in the reference but missing in the inferred pathway and a false positive (FP) is a non-seed node absent in the reference but found in the inferred pathway. The sensitivity (Sn) is defined as the ratio of inferred true instances versus all true instances, whereas the positive predictive value (PPV) gives the ratio of inferred true instances versus all inferred instances.

$$Sn = \frac{TP}{(TP + FN)} \tag{1}$$

$$PPV = \frac{TP}{(TP + FP)} \tag{2}$$

We can combine sensitivity and positive predictive value to calculate the accuracy as their geometric mean.

$$Acc_g = \sqrt{Sn * PPV} \tag{3}$$

3 Results

3.1 Study case aromatic amino acid biosynthesis

To illustrate the idea of pathway inference, we will discuss the aromatic amino acid biosynthesis pathway. Figure 1A shows the pathway as annotated in BioCyc. This pathway is active in *E. coli* and produces aromatic amino acids (tyrosine, tryptophan and phenylalanine) from erythrose-4-phosphate. The first part of this pathway is linear and ends in chorismate. From chorismate onwards, the pathway splits into three branches, one leading to tryptophan and the other bifurcating to phenylalanine and tyrosine respectively. The entire pathway, excluding the terminal compounds, is made up of 34 compound and reaction nodes.
The aromatic amino acid pathway is tightly regulated on the transcriptional level. In presence of one of the end products (that is an aromatic amino acid), the corresponding synthesis branch is down-regulated. The linear part of the pathway is also subject to regulation (on the enzymes catalyzing the first, fifth and sixth reaction) integrating feed-back loops from the three end-products. From the set of transcriptionally regulated reactions, we selected DAHPSYN-RXN, SHIKIMATE-KINASE-RXN, PRAISOM-RXN, PHEAMINOTRANS-RXN, TYRAMINOTRANS-RXN and RXN0-2382 (BioCyc identifiers) as seed nodes. In our previous 2-end path finding approach [3, 4] we were restricted to only two seed nodes. To simulate this situation, we applied the pair-wise k-shortest paths algorithm on the start (DAHPSYN-RXN) and one of the end reactions (RXN0-2382). The resulting pathway, shown in Figure 1B, connects the two seed reactions via a shortcut, bypassing a major part of the reference pathway. The resulting linear path fits the branched reference pathway with a low accuracy (14%). However, if we repeat the inference with the full seed node set, we recover the reference pathway with an accuracy of 97% (Figure 1C). Thus,

without surprise, we observe that multi-seed subgraph extraction is more appropriate to infer branched pathways than 2-end path finding. In the next section, we evaluate various algorithms and parametric choices for the multi-seed subgraph extraction.

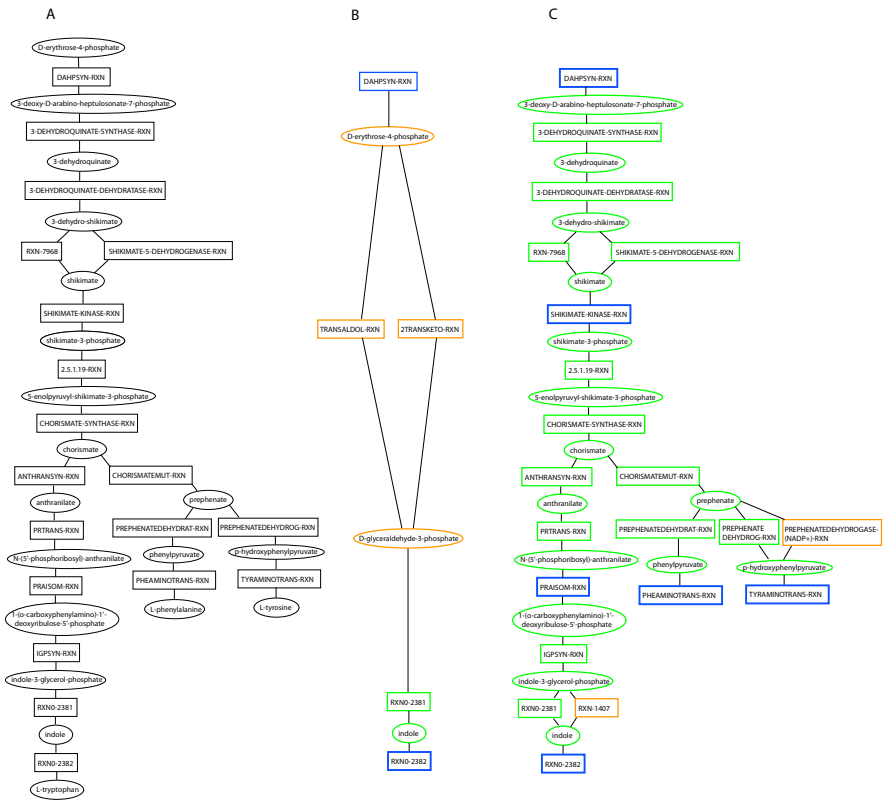

Fig. 1. Aromatic amino acid biosynthesis pathway as annotated in BioCyc (A), inferred with two seed reactions (B) and inferred with 6 seed reactions (C). For both inferences, the pair-wise k-shortest paths algorithm has been run on the compound-weighted, directed MetaCyc graph. Seed reactions have a blue border, false positive nodes an orange border and true positive nodes a green border. Compound nodes are represented as ellipses and labeled with their names, whereas reaction nodes are drawn as rectangles and labeled with their BioCyc identifiers.

3.2 Parameter combinations

The performance of the pair-wise k-shortest paths algorithm, kWalks and the hybrid approach has been evaluated for 82 parameter value combinations ac-

Table 1. The ten pathway inference experiments (parametric combinations) resulting in the highest geometric accuracy in our evaluation. For each experiment, its parameter values, its runtime and its geometric accuracy (averaged over all inferences) are displayed.

Algorithm	Iteration number	Weighting scheme	Inflation after iteration	Directed graph	Edge relevances used as weight	Geometric accuracy	Runtime in seconds
Hybrid	6	Compound degree weight	False	True	False	0.6822	636
Pair-wise k-shortest paths	0	Compound degree weight	False	True	False	0.6803	445
kWalks	3	Compound degree weight	True	True	False	0.6796	130
kWalks	6	Inflated compound degree weight	False	True	False	0.679	309
Hybrid	3	Compound degree weight	False	True	False	0.6786	393
kWalks	6	Compound degree weight	True	True	False	0.6778	323
kWalks	6	Compound degree weight	False	True	False	0.6773	312
Hybrid	0	Compound degree weight	False	True	False	0.6757	183
Hybrid	6	Compound degree weight	True	True	False	0.6738	744
Hybrid	3	Compound degree weight	True	True	False	0.6724	431

cording to the Sn, PPV and Acc_g criteria described above.

Table 1 lists the top ten experiments, with geometric accuracies averaged over all inferences done for each experiment. Interestingly, all three algorithms are present in Table 1. In agreement with our previous analysis [3, 4], directed, compound-weighted graphs yield highest pathway inference accuracies. The performance of the hybrid approach increases if the original graph weights rather than the edge relevances obtained by kWalks are fed into the pair-wise k-shortest paths algorithm. Inflating edge relevances after kWalks iterations does not have a significant impact on pathway inference accuracy.

It is worth noting that kWalks without iteration is not among the top ten experiments. Figure 2A shows a summary of all inferences done for kWalks without iteration in the directed, un-weighted MetaCyc graph. For comparison, Figure 2B displays the geometric accuracies obtained for kWalks with three iterations under the same conditions. Closer inspection of the geometric accuracy heat maps

reveals that iterating kWalks improves geometric accuracies for the tryptophan biosynthesis, isoleucine biosynthesis I and UDP-N-acetylgalactosamine biosynthesis pathways. The average geometric accuracy of kWalks increases from 62% for one iteration to 64% for three iterations. This illustrates that calling kWalks iteratively improves pathway inference accuracy. Not surprisingly, Table 1 lists only kWalks experiments in which this algorithm has been iterated three or six times.

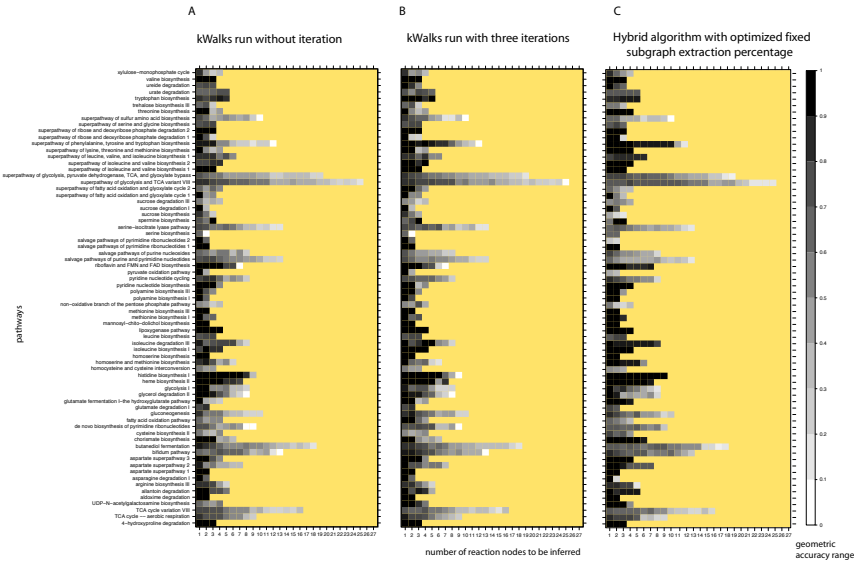

Fig. 2. The geometric accuracies obtained for each pathway as a heat map. The x-axis indicates the number of non-seed reaction nodes (those that the algorithm needs to infer). Each row corresponds to one reference pathway. The geometric accuracy is reflected by a gray scale from white ($Acc_g = 0$) to black ($Acc_g = 1$). A. Heat map obtained for kWalks in the un-weighted, directed MetaCyc graph without inflation or iteration. B. Heat map obtained under the same conditions, but with kWalks iterated three times. C. Summary of the inferences done for the hybrid approach in the directed, compound-weighted MetaCyc graph. The size of the subgraph extracted by kWalks was set to 0.5%, which was the optimal value found by our evaluation.

3.3 Hybrid algorithm optimization

In the evaluation shown in Table 1, we set the fixed subgraph size for the hybrid approach to 5%. However, subsequent variation of the subgraph size in a percentage range from 0.1% to 10% showed that 0.5% is the optimal percentage

for subgraph extraction in the hybrid approach. When the subgraph extraction percentage is set to 0.5%, the average geometric accuracy of the hybrid approach reaches 72%, which is the highest percentage obtained for any experiment. Figure 2C shows the geometric accuracy of each inference in a heat map.

4 Discussion

4.1 Parameters

Graph directionality The directed MetaCyc graph yields higher geometric accuracies than the undirected one. In the undirected MetaCyc graph, it is possible to traverse the graph from substrate to substrate or from product to product, which is prevented in the directed graph.

Compound weighting In our previous studies [3, 4], we showed that the weight is the most determinant parameter for inferring relevant pathways by 2-end path finding. As expected, node weighting also exerts a strong impact on the performances of multi-seed pathway inference, but this impact depends on the algorithm used. The pair-wise k-shortest paths performed best in the compound-weighted MetaCyc graph. But surprisingly, kWalks without iteration resulted in higher accuracies when applied to the un-weighted rather than to the weighted MetaCyc graph. Interestingly, kWalks automatically induces weights that favor relevant compounds. Actually, the kWalks relevance score can be interpreted as a context-specific betweenness index, and we can thus understand that it penalizes highly connected compounds, thereby explaining the good results obtained by relevance weighting. In the hybrid approach, the resulting average geometric accuracy for the un-weighted graph is higher when we pass the kWalks induced weights (edge relevances) rather than the original weights to the second step of the algorithm. However, if we run the hybrid approach on the weighted graph, the kWalks induced weights decrease the accuracy compared to the original weights. Inflation of kWalks induced weights does not improve results significantly.
For most parameters, we determined optimal values and their combination (with respect to the reference pathways) by an exhaustive search. However, compound weights were chosen heuristically and may be optimized in future by a machine learning approach.

4.2 Algorithms

Although the pair-wise k-shortest paths algorithm is slow (7 minutes per pathway inference in average), its average geometric accuracy figures among the top ten. In contrast, kWalks without iteration runs in seconds, but yields unsatisfactory accuracies. Upon closer inspection, it became apparent that kWalks results in high sensitivities and low positive predictive values. The positive predictive value of the kWalks algorithm can be increased by invoking it iteratively or by combining it with pair-wise k-shortest paths in the hybrid algorithm. Both

strategies are paid with a longer runtime.
The highest average geometric accuracy was obtained with the optimized hybrid approach. This shows that pair-wise k-shortest paths and kWalks are complementary. The focus of kWalks is to capture the part of the input graph most relevant for connecting the seeds, reducing the number of false negatives at the cost of increasing the number of false positives. Pair-wise k-shortest paths can then discard false positives introduced by kWalks.

4.3 Similar approach to subgraph extraction

Recently, Koren and co-workers [9] designed a proximity measure that avoids dead-end nodes and takes node degree and multiple paths between seeds into account. Based on this measure, they describe a subgraph extraction approach that relies on finding the k shortest paths between seed nodes. The resulting paths are combined in such a way that proximity between seeds is maximized while minimizing the subgraph size.
This extraction procedure aims at capturing the paths that contribute to the proximity of nodes. It is worth noting that two nodes with many paths between them are considered closer than two nodes connected by a few paths. Therefore, this algorithm will likely return more alternative paths between seeds than our algorithms do. This is very interesting when one wants to explore the metabolic neighborhood of a set of seed nodes, but less desirable when a metabolic pathway should be predicted.

5 Conclusion

We have presented three different algorithmic approaches to infer metabolic pathways from metabolic graphs: kWalks, pair-wise k-shortest paths, and a hybrid that combines both.
The former two algorithms have complementary strengths and weaknesses. Our evaluation on 71 yeast pathways has shown that their combination in the hybrid approach yields the highest geometric accuracies.

In future, we will apply these algorithms to microarray data to infer metabolic pathways from co-expressed enzyme-coding genes.

Acknowledgments

KF is supported by Actions de Recherches Concertes de la Communaut Franaise de Belgique (ARC grant number 04/09-307). The BiGRe laboratory is a member of the BioSapiens Network of Excellence funded under the sixth Framework program of the European Communities (LSHG-CT-2003-503265). The work was supported by the Belgian Program on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, project P6/25 (BioMaGNet).

We would like to thank the referees for their helpful comments, in particular for pointing to the article of Koren and co-workers.

References

1. Callut, J.: First Passage Times Dynamics in Markov Models with Applications to HMM Induction, Sequence Classification, and Graph Mining. PhD Thesis Dissertation, Université catholique de Louvain (2007)
2. Caspi, R. et al.: The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. Nucleic Acids Research 36, D623–D631 (2008)
3. Croes, D., Couche, F., Wodak, S., van Helden, J.: Metabolic PathFinding: inferring relevant pathways in biochemical networks. Nucleic Acids Research 33, W326–W330 (2005)
4. Croes, D., Couche, F., Wodak, S., van Helden, J.: Inferring Meaningful Pathways in Weighted Metabolic Networks. J. Mol. Biol. 356, 222–236 (2006)
5. Dahlquist, K.D., Salomonis, N., Vranizan, K., Lawlor, S.C., Conklin, B.R.: GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways. Nature Genetics 31, 19–20 (2002)
6. Dupont, P., Callut, J., Dooms, G., Monette, J.-N., Deville, Y.: Relevant subgraph extraction from random walks in a graph. Research Report UCL/FSA/INGI RR 2006-07 (2006-07)
7. Jimenez, V.M., Marzal, A.: Computing the K-shortest Paths: a New Algorithm and an Experimental Comparison. In: Proc. 3rd Int. Worksh. Algorithm Engineering (WAE 1999) 1668, pp. 15–29 (1999)
8. Kemeny, J.G., Snell, J.L.: Finite Markov Chains. Springer-Verlag (1983)
9. Koren, Y., North, S.C., Volinsky, C.: Measuring and extracting proximity in networks, In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 245–255 (2006)
10. Krieger, C.J. et al.: MetaCyc: a multiorganism database of metabolic pathways and enzymes. Nucleic Acids Research 32, D438–D442 (2004)
11. Paley, S.M., Karp, P.D.: The Pathway Tools cellular overview diagram and Omics Viewer. Nucleic Acids Research 34, 3771–3778 (2006)
12. van Helden, J., Gilbert, D., Wernisch, L., Schroeder, M., Wodak, S.: Applications of regulatory sequence analysis and metabolic network analysis to the interpretation of gene expression data. In: Lecture Notes in Computer Science 2066, pp. 155–172 (2001)
13. van Helden, J., Wernisch, L., Gilbert, D., Wodak, S.: Graph-based analysis of metabolic networks. In: Ernst Schering Res Found Workshop, pp. 245-74. Springer-Verlag (2002)