Automatic extraction of relevant nodes in biochemical networks

Sébastien Vast, Pierre Dupont, Yves Deville

Department of Computing Science and Engineering
Université catholique de Louvain
Place Sainte Barbe, 2
B-1348 Louvain-la-Neuve - Belgium
{svast, pdupont, deville}@info.ucl.ac.be
http://www.info.ucl.ac.be/~{svast,pdupont,yde}

Abstract : In this paper we describe a novel method for extracting a set of nodes that best capture the connections between k given nodes of interest in a biochemical network. This method relies on the projection of the nodes of the network, seen as an undirected graph, into an euclidean space. Euclidean distances between nodes in the projected space correspond to their commute time distances in the original graph, a measure based on a random walk model on the graph. Commute time reflects the distance between two nodes while considering all paths connecting them. Results on artificial data illustrate the interest of this approach.

Keywords: biochemical network analysis, subgraph mining, commute time distance, spectral graph analysis

1 Introduction

Biochemical networks model interactions between biochemical entities within cells. Metabolism can be viewed as a network of chemical reactions catalyzed by enzymes, and connected via their substrates and products; a metabolic pathway is then a coordinated series of reactions. Other types of biochemical networks include regulatory or signal transduction networks. Several models exist to represent biochemical networks (Deville *et al.*, 2003). In most cases, these networks can be viewed as directed or undirected graphs. The present work is part of the BioMaze project which aims to produce computer tools for analyzing biochemical networks. BioMaze extends the Amaze project which aims to build a biochemical database integrating the three types of networks mentioned above and to provide dedicated query tools (van Helden *et al.*, 2000).

Atelier Apprentissage et BioInformatique, 7^e Conférence francophone sur l'apprentissage automatique, CAp 2005, Nice, pp. 21-31, 2005.

The specific problem we address here is the extraction of a relevant subgraph of an undirected[1] graph, which best explains the relations between k given nodes of interest in this graph. Assume, for instance, we are analyzing the synthesis of *pyruvate* from *glucose* and would like to study the possible influence of the expression of a given *gene* on a protein, say *phospho-fructokinase-2*, in the context of the regulation of this metabolic pathway. In this case, we have 4 nodes of interest in a possibly very large graph of interactions and we would like to extract a relevant subgraph explaining the relations between these 4 nodes. The methods described in this paper are also applicable to other practical domains.

This paper presents a novel approach to this problem. It relies on the projection of the nodes of the graph into an euclidean space. Euclidean distances between nodes in the projected space correspond to their *commute time distances* in the original graph, a measure based on a random walk model on the graph (Saerens *et al.*, 2004). Commute time reflects the distance between two nodes while considering all paths connecting them. This contrasts with simpler approaches which would extract only specific paths between each pair of nodes of interest, such as shortest distance or maximal flow paths. Here the goal is the extraction of a relevant subgraph as this is considered to be more informative. An inspiring approach to this problem was presented recently in (C. Faloutsos & Tomkins, 2004) but the problem was restricted to 2 nodes of interest. We adopt here a different point of view allowing for a direct solution to the general problem with any number of nodes of interest. We propose to solve the problem in two steps: the extraction of a subset of relevant nodes in the graph followed by the construction of a subgraph connecting them. The present contribution focuses on the first step.

Section 2 proposes a formal statement of the problem we address. Some possible methods to solve it are discussed and contrasted with our approach. The theory behind the notion of commute time distance is summarized in section 3. Section 4 details how to use commute time distances in order to extract a subset of relevant nodes in a graph. Practical experiments are presented in section 5.

2 The problem of extracting a subset of relevant nodes

Problem statement: **Given** a connected undirected graph $G = (V, E)$, where V denotes a set of nodes (or vertices) and E denotes a set of weighted edges, a non-empty set $K \subseteq V$ of nodes of interest and s a strictly positive integer, **find** a set $S \subseteq V \setminus K$ of nodes, with $|S| = s$, optimizing a goodness function $g(S, K)$. The goodness function $g(S, K)$ measures how well the s *extracted nodes* explain the relations between the $k = |K| \geq 2$ *nodes of interest* in the graph.

The goodness function should measure how well the nodes of interest are connected through paths to which the extracted nodes belong. A naive approach to this problem consists in extracting nodes belonging to shortest paths between pairs of nodes of interest. Consider, for instance, the graph depicted in Figure 1 and assume this graph represents a road map between cities A and B (*i.e.* $k = 2$, in the present case). The

[1] Even though there is a direction of flow in a metabolic pathway, the type of graph analysis considered here does not require directed edges.

shortest distance[2] approach would typically select nodes C and D belonging to the highway connecting A and B. However, as soon as one edge is removed along this path (e.g. in case of a traffic jam) no alternative route from A to B goes through C or D. Nodes included in the dashed circle are more relevant here as they belong to many alternative routes connecting A and B, even though none of these routes might be shorter than the highway. Thus the goodness function should take into account many alternatives routes, possibly all of them.

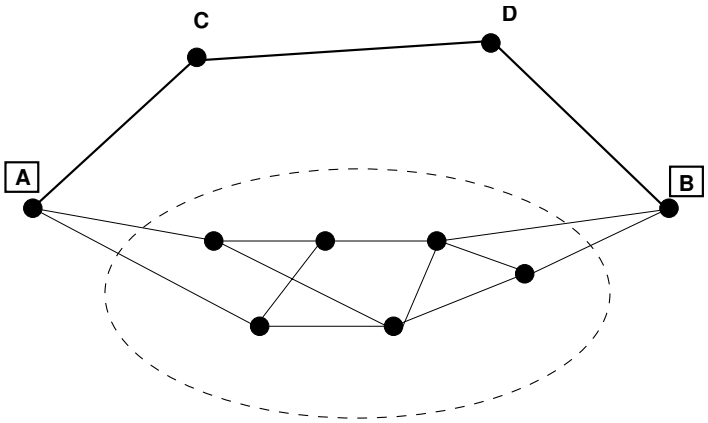


Figure 1: Nodes that best capture the connections between A and B are in the dashed circle as they belong to many alternative routes from A to B, or conversely.

Faloutsos et al. proposed an interesting approach to the more general problem of extracting a relevant subgraph (C. Faloutsos & Tomkins, 2004). This approach can directly be applied to our problem of extracting a subset of the graph nodes. They restrict their attention to the case for which $k = 2$. The goodness function $g(S, K)$ is based on an electrical analogy. The 2 nodes of interest are respectively considered to be the source and the sink of an electrical current. The algorithm searches the paths followed by the current flow and maximizes the sum of current flow in the extracted subgraph. In addition, each node includes some current loss in order to penalize long paths and very highly connected nodes (hubs)[3]. This approach takes into account several paths between the nodes of interest. The fraction of current captured by the subgraph depends on the number and weight of such paths. One drawback of this method is that, due to the current loss, the solution depends on which of the 2 nodes of interest is chosen to be the current source. We propose in the present work an alternative method which deals with any number of nodes of interest with no preference a priori defined between them.

[2]This distance may correspond to the travel time in this particular case.

[3]While it is interesting to extract nodes offering alternative routes to the nodes of interest, hubs do not explain well the specific relations between the nodes of interest as they are well connected to most nodes.

3 Euclidean commute time distance

As motivated by the discussion in section 2, we are looking for a measure describing how well several nodes are connected in a graph by considering all possible paths connecting them. This measure will then be applied to the extraction of a subset of relevant nodes in a graph as detailed in section 4.

The proposed measure relies on a random walk model on the graph. This model assigns transition probabilities to the edges, so that a random walker will jump from one node to another with a probability proportional to the weight of the edge connecting them. The *average commute time*[4] between nodes i and j computes the average time taken by a random walker for reaching node j from node i, and coming back to i. The square root of this quantity is a distance measure between any two nodes called the *euclidean commute time distance* (ECTD). Most of the theory, summarized in the present section, was introduced in (Saerens *et al.*, 2004). The application of this distance measure to the extraction of a subset of relevant nodes in a graph is detailed in section 4.

Section 3.1 introduces some notations and, in particular, the Laplacian matrix $\mathbf{L}$ of a graph. Section 3.2 details how to compute the ECTD from $\mathbf{L}$.

3.1 The Laplacian matrix of a weighted graph

We consider a weighted undirected graph $G = (V, E)$ with strictly positive weights between each pair of connected nodes. The graph order $|V|$ is also denoted n in the sequel. The larger the weight w_{ij} of the edge connecting node i to node j, the easier the communication between i and j is assumed to be. Moreover, the weights are required to be symmetric ($w_{ij} = w_{ji}$). The *adjacency matrix* $\mathbf{A}$ is defined in the usual way:

$$a_{ij} = \begin{cases} w_{ij} & \text{, if node } i \text{ is connected to node } j \\ 0 & \text{, otherwise.} \end{cases}$$

The diagonal *degree matrix* $\mathbf{D}$ is defined as follows. $d_{ii} = \sum_{l=1}^{n} a_{il}$ and $d_{ij} = 0$, if $i \neq j$. A related quantity is the *graph volume*, that is the sum of node degrees: $D_G = \sum_{i=1}^{n} d_{ii}$.

The *Laplacian matrix* $\mathbf{L}$ of the graph is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. When G has a single connected component, the rank of $\mathbf{L}$ is $n - 1$. Moreover, one can easily show that $\mathbf{L}$ is symmetric and positive semidefinite (Chung, 1997).

3.2 Computation of the commute time distances

Klein and Randic proposed in (Klein & Randic, 1981) a distance measure between graph nodes, called *resistance distance* which has the property of decreasing when the number of paths between two nodes increase. As shown by Chandra (Chandra *et al.*,

[4]This notion of commute *time* is equivalent to the average *number of steps* a random walker would make on average to commute between both nodes, since the random walker is assumed to make one step at each time clock.

1989), this measure can be expressed in terms of the random walk model described below.

A random walk on a graph is a Markov chain describing the sequence of nodes visited by a random walker. A state of the Markov chain is associated with every node of the graph. A random variable $X(t)$ represents the current state of the Markov chain at time t. The probability of transiting to state j at time $t+1$, given the current state is i at time t, is given by:

$$P(X(t+1) = j | X(t) = i) = p_{ij} = a_{ij}/d_{ii}.$$

Thus, from any state i, the probability to jump to a state j is proportional to the weight a_{ij} of the edge between i and j. The transition matrix $\mathbf{P} = [p_{ij}]$ of the Markov chain is related to the degree and adjacency matrices as $\mathbf{P} = \mathbf{D^{-1}A}$.

The *average first-passage time* $m(j|i)$ is defined as the average number of steps a random walker, starting in state i, will take to reach state j for the first time. These measure can be computed by the following recurrence (Norris, 1997):

$$\begin{cases} m(j|i) = 1 + \sum_{l=1, l\neq j}^{n} p_{il}\, m(j|l) \text{ for } i \neq j \\ m(j|j) = 0 \end{cases} \quad (1)$$

A closely related measure is the *average commute time*, $q(i,j)$, defined as the average number of steps a random walker, starting in state i, will take to enter state j for the first time, and go back to state i for the first time: $q(i,j) = m(j|i) + m(i|j)$. Note that, in general, $m(i|j) \neq m(j|i)$, while the average commute time is symmetric by definition. As shown by several authors, the average commute time is a distance (Klein & Randic, 1981; Gobel & Jagers, 1974). Moreover the square root of the average commute time defines an euclidean distance (Saerens *et al.*, 2004).

A first method for computing euclidean commute time distances is based on the iterative solving of the recurrences (1). An alternative approach derives from the Moore-Penrose pseudoinverse of the Laplacian $\mathbf{L}$, denoted by $\mathbf{L}^+$, as proposed in (Saerens *et al.*, 2004):

$$q(i,j) = D_G(l^+_{ii} + l^+_{jj} - 2l^+_{ij}) \quad (2)$$

If we further define $\mathbf{e_i}$ as the ith column of the $n \times n$ identity matrix, equation (2) can be rewritten as

$$q(i,j) = D_G(\mathbf{e}_i - \mathbf{e}_j)^T L^+ (\mathbf{e}_i - \mathbf{e}_j), \quad (3)$$

where each node i is represented by a unit base vector $\mathbf{e}_i$. These nodes can be mapped into an euclidean space that preserves the commute time distances as $\mathbf{L}^+$ is positive semidefinite. Indeed, every positive semidefinite matrix can be transformed to a diagonal matrix (see, e.g., (Meyer, 2000)), $\mathbf{\Lambda} = \mathbf{U}^{\mathrm{T}}\mathbf{L}^+\mathbf{U}$, where $\mathbf{U}$ is an orthonormal matrix made of the eigenvectors of $\mathbf{L}^+$. Hence, the commute time distances can be rewritten as:

$$q(i,j) = D_G\, (\mathbf{x}'_i - \mathbf{x}'_j)^{\mathrm{T}} (\mathbf{x}'_i - \mathbf{x}'_j) \quad (4)$$

where the following transformations have been applied: $\mathbf{x}_i = \mathbf{U}^{\mathrm{T}}\mathbf{e}_i$, and $\mathbf{x}'_i = \mathbf{\Lambda}^{1/2}\mathbf{x}_i$.

So, in this n-dimensional Euclidean space, the transformed node vectors, $\mathbf{x}'_i$, are exactly separated by euclidean commute time distances (up to the scaling factor D_G). Close points in this *ECTD space* represent nodes well connected in the original graph

G according to any possible paths between them, and the euclidean distance between them measures this connectivity in the original graph.

The ECTD space has dimensionality n, the graph order, but projection to a subspace preserving as much information as possible can reduce computation time. The so-called spectral (or eigenvector) decomposition of $\mathbf{L}^+$ is given by:

$$\mathbf{L}^+ = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{T}} = \sum_{l=1}^{n-1} \lambda_l \, \mathbf{u}_l \mathbf{u}_l^{\mathrm{T}} \tag{5}$$

where $\lambda_1 > \lambda_2 > \ldots > \lambda_{n-1} > \lambda_n = 0$ are the eigenvalues of $\mathbf{L}^+$, and $\mathbf{u}_l$ the associated eigenvectors. The eigenvector expansion of $\mathbf{L}^+$ can be computed up to $m < n-1$, by considering only the m largest eigenvalues of $\mathbf{L}^+$. This gives rise to an m-dimensional subspace where the commute time distances are approximately preserved.

Finally, since $\mathbf{L}$ and $\mathbf{L}^+$ have the same set of eigenvectors but inverse (non zero) eigenvalues, we do not need to explicitly compute the pseudoinverse of $\mathbf{L}$. It is only necessary to compute the smallest non zero eigenvalues of $\mathbf{L}$, which correspond to the largest eigenvalues of $\mathbf{L}^+$, and their associated eigenvectors. Fast iterative methods exists for this purpose (Golub & Loan, 1996; Sorensen, 1996). The complexity for computing one eigenvalue/eigenvector is $O(n^2)$ and the overall complexity for this method is thus $O(mn^2)$.

4 Node subset minimizing euclidean commute time

The relevant node subset problem can be easily formulated and solved using the euclidean commute time distances between any graph nodes and the nodes of interest. More specifically, we consider the following goodness function :

$$g(S, K) = \sum_{i \in S} d_r(i, K) \tag{6}$$

with

$$d_r(i, K) = \min_{W \subseteq K, |W| = r} \sum_{j \in W} q(i, j)$$

Thus, the contribution of each extracted node to the goodness of the subset S is the sum of the commute time distances to its r ($1 \leq r \leq k$) closest nodes of interest in the ECTD (sub-)space. In the experiments reported below, we considered the distances to the two closest nodes of interest, for each extracted node ($r = 2$). The choice $r = k$ would correspond to considering the distances to all nodes of interest. On one hand, this would allow to take into account the connectivity to all nodes of interest. On the other hand, as this measure would be more global, the extracted nodes might not be particularly well connected to any specific node of interest. We will further study this trade-off in our future work.

Computing an optimal S, which minimizes g for a given number s of nodes to be extracted, is straightforward once the commute time distances between any node of interest and the other nodes of the graph have been computed. It simply amounts to

compute $d_r(i, K)$ for each possible node i of the graph (except the k nodes of interest themselves) and to return the nodes with the s smallest values.

Figure 2 presents the graph of Figure 1 with nodes indexed in increasing order according to $d_2(i, K)$ (here, $K = \{A, B\}$). Unit weight edges were considered in this example.

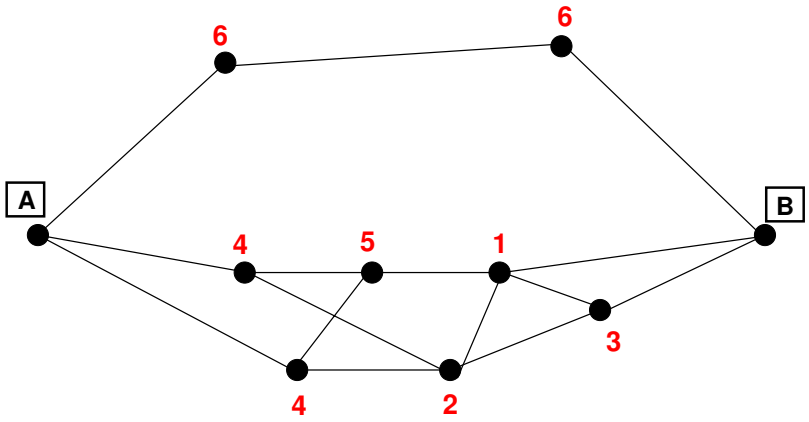


Figure 2: The nodes of this graph are labeled in increasing order (ties are assigned the same rank) according to the sum of their commute time distances to A and B respectively.

5 Experiments

The ultimate objective of this work is to provide a method for a biologist to automatically extract a subset of relevant nodes related to given nodes of interest in a large biochemical network. In order to assess the performance of the proposed method, preliminary experiments with artificial graphs are reported here.

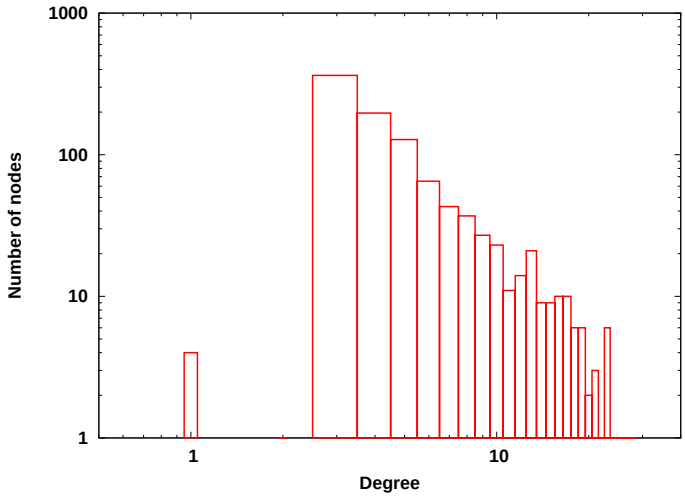


Figure 3: Average degree distribution of graphs used for testing.

A set of 10 graphs of 100 nodes were randomly generated using a power-law graph generator (Barabasi *et al.*, 2000). For each graph, five nodes were used as initial seeds. Next, 95 nodes were iteratively added and randomly connected to 3 nodes of the current graph. At each step, the probability of an existing node to be connected to the new node

is proportional to its current degree. Each generated graph contains a single connected component and all edges have a unit weight. The degree distribution (averaged over all generated graphs) is depicted (in log scales) in Figure 3.

For each graph tested, 10 sets of k nodes of interest were randomly selected. Results are reported for $k = 2, 4$ and 8. In each case, an increasing number of s nodes were extracted. The distance measure $D = \frac{\sum_{i \in S} d_2(i,K)}{\sum_{i \in V \setminus K} d_2(i,K)}$ is the cumulated distance of the subset S of extracted nodes relative to the distance of the total set $V \setminus K$ of nodes which can possibly be extracted. As we aim at minimizing a distance in this case, the smaller D the better.

Comparative results with the method proposed by Faloutsos et al. (C. Faloutsos & Tomkins, 2004) are possible when $k = 2$. These results are presented in Figure 4. Both approaches perform very similarly in this setting, showing that they capture essentially the same information (at least for the tested graphs). However, Faloutsos method cannot extract more than 29 % of nodes in this case (28 out of the 98 nodes which can potentially be extracted with our approach). This comes from the fact that this method only extracts nodes on loopless paths between the 2 nodes of interest. Hence a significant fraction of the graph nodes (here 71 %) may not respect this constraint. On one hand, this illustrates an advantage of our approach. On the other hand, Faloutsos method is more general as it does not only extract a node subset but a connected subgraph. Extension of our method to deal with this more general problem is part of our future work.

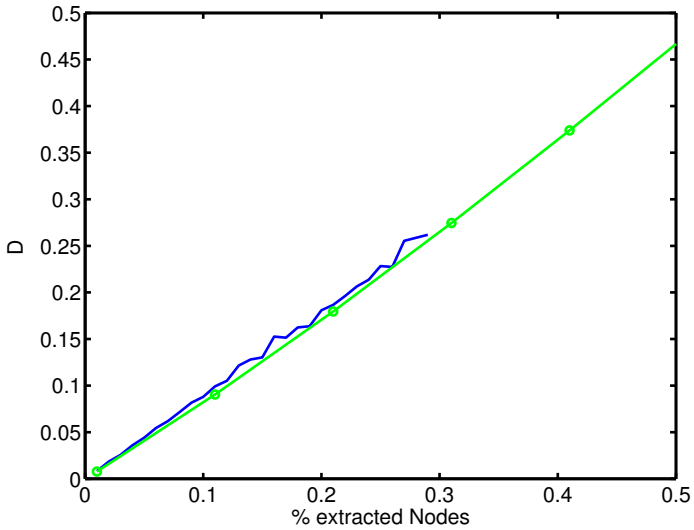
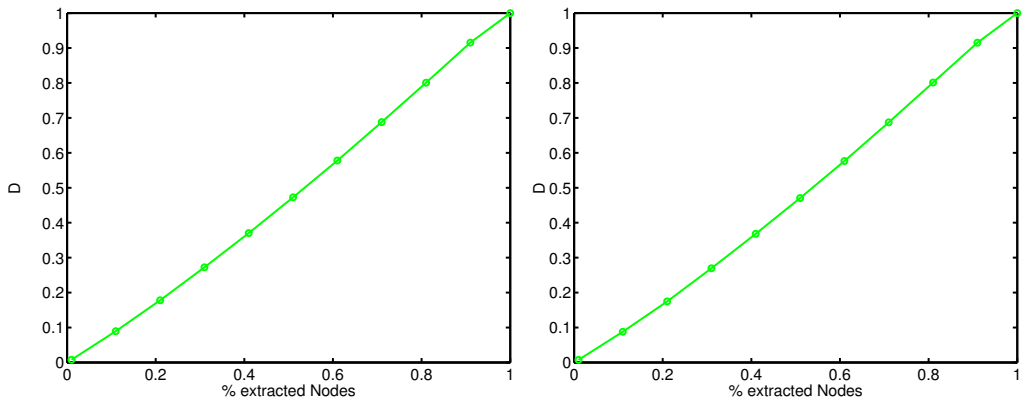


Figure 4: Distance of the extracted subsets for increasing number of extracted nodes. The x axis gives the value of $\frac{s}{n-k}$, that is the percentage of extracted nodes. The green curve (circles) corresponds to our approach minimizing commute times, while the blue curve corresponds to the method of Faloutsos. Results are obtained for $k = 2$ and averaged over 100 tests.

Figure 5 illustrates the results of our approach for $k = 4$ and 8 on the same graphs. Both curves behave similarly and illustrate the generalization of our approach to larger sets of nodes of interest. Note that the computational complexity remains essentially the same as it is dominated by the computation of the same commute time distances in all cases.



(a) $k = 4$ (b) $k = 8$

Figure 5: Distance of the extracted subsets for increasing number of extracted nodes for $k = 4$ or $k = 8$.

In all results presented so far, all edge weights were assumed to be equal (standard deviation $\sigma = 0$). Figure 6 presents the extraction results for another set of 10 graphs of 200 nodes for which a normal distribution on edge weight with a much larger standard deviation ($\sigma > 200$) was defined. As slightly better performance is obtained for these graphs as the distribution of commute time distances is sharper in this case.

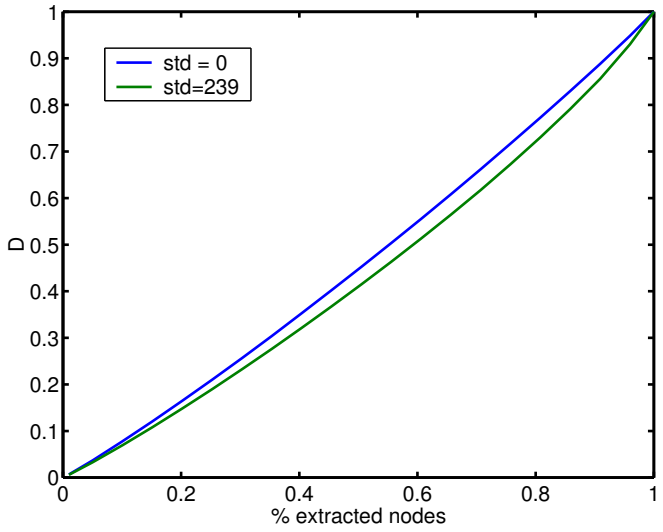


Figure 6: Extraction results for equal weight for all edges ($\sigma = 0$) or large standard deviation ($\sigma = 239$) for a normal weight distribution. Average results for 100 tests with $k = 2$ (10 randomly selected pairs of nodes of interest for each graph).

6 Conclusion and future work

We propose in this paper a novel approach to the extraction of nodes in a biochemical network which best explain the connections between k given nodes of interest in this network. This approach uses commute time distances between nodes, a measure of how well two nodes are connected in a graph by considering all possible paths between them. It is based on the projection of the nodes of the network, seen as an undirected graph, into an euclidean space. Euclidean distances between nodes in the projected ECTD space correspond to their commute time distances in the original graph.

Several questions need to be addressed in the future.

1. The commute time distances can be approximated if the nodes of the original graph are projected into a subspace of the full ECTD space. A lower dimension subspace corresponds to a coarser approximation to the actual commute times while reducing the computational complexity. We will study the trade-off between this complexity and the quality of the set of extracted nodes.

2. Our goodness measure for the extracted node subset is based on the commute time distance from each extracted node to its two closest nodes of interest. As discussed in section 4, alternative goodness measures will be investigated.

3. The more general problem of a relevant subgraph extraction will be considered. Starting from the set of extracted nodes, some edge selection in the original graph has to be designed. This should be derived from the fraction of edges responsible for the largest part of the commute time distances between nodes.

4. Actual experiments on real biochemical networks and result interpretations by biologists are also part of the current project. Comparisons between extracted subgraphs and known pathways could be performed in this regard.

References

BARABASI A., ALBERT R., JEONG H. & BIANCONI G. (2000). Power-law distribution of the world wide web. *Science*, **287**(12).

C. FALOUTSOS K. M. & TOMKINS A. (2004). Fast discovery of connection subgraphs. In *10th ACM Conference on Knowledge Discovery and Data Mining (KDD)*, volume 2, p. 118–127.

CHANDRA A. K., RAGHAVAN P., RUZZO W. L. & SMOLENSKY R. (1989). The electrical resistance of a graph captures its commute and cover times. In *STOC '89: Proceedings of the twenty-first annual ACM symposium on Theory of computing*, p. 574–586: ACM Press.

CHUNG F. (1997). *Spectral graph theory*. American Mathematical Society.

DEVILLE Y., GILBERT D., VAN HELDEN J. & WODAK S. J. (2003). An overview of data models for the analysis of biochemical pathways. *Briefings in Bioinformatics*, **4**(3), 246–259.

GOBEL F. & JAGERS A. (1974). Random walks on graphs. *Stochastic Processes and their Applications*, **2**, 311–336.

GOLUB G. H. & LOAN C. F. V. (1996). *Matrix Computations, 3rd edition*. Johns Hopkins University Press.

KLEIN D. & RANDIC M. (1981). Resistance distance. *Journal of Mathematical Chemistry*, **12**, 81–95.

MEYER C. D. (2000). *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics.

NORRIS J. (1997). *Markov Chains*. Cambridge University Press.

SAERENS M., FOUSS F., YEN L. & DUPONT P. (2004). The princial components analysis of a graph, and its relationships to spectral clustering. In *Proceedings of the 15th European Conference on Machine Learning (ECML 2004)*, volume 3201 of *Lecture Notes in Artificial Intelligence*, p. 371–383: Springer-Verlag.

SORENSEN D. C. (1996). *Implicitly restarted Arnoldi/Lanczos methods for large scale eigenvalue calculations*. Rapport interne TR-96-40, Rice University.

VAN HELDEN J., NAIM A., MANCUSO R., ELDRIDGE M., WERNISCH L., GILBERT D. & WODAK S. (2000). Representing and analyzing molecular and cellular function using the computer. *Biol. Chem.*, **381(9-10)**, 921–935.