Bioinformatics Advance Access published October 23, 2006

An Algorithm for Modularity Analysis of Directed and Weighted Biological Networks Based on Edge-Betweenness Centrality

Jeongah Yoon[1], Anselm Blumer[2], and Kyongbum Lee[1,*]

Departments [1]Chemical and Biological Engineering and [2]Computer Science, Tufts University, Medford, MA 02155.

Associate Editor: Alvis Brazma

ABSTRACT

Motivation: Modularity analysis is a powerful tool for studying the design of biological networks, offering potential clues for relating the biochemical function(s) of a network with the 'wiring' of its components. Relatively little work has been done to examine whether the modularity of a network depends on the physiological perturbations that influence its biochemical state. Here, we present a novel modularity analysis algorithm based on edge-betweenness centrality, which facilitates the use of directional information and measurable biochemical data.

1 INTRODUCTION

A common feature of large, complex biological networks is that they are organized into smaller sub-networks consisting of directly interacting, or 'connected,' molecular components. Recent studies have suggested that these sub-networks correspond to biologically meaningful, functional units, or 'modules' (Hartwell, et al., 1999). In this light, one approach to understanding the design of biological networks is to examine their modularity. For example, comparative analyses of structurally similar modules across different species may identify mutually shared functions, associate a modular structure with a new function, and provide insight into the evolution of various network structures (Sharan and Ideker, 2006). One issue that remains to be addressed is whether particular structures are inherent to a network or dependent on its functional state. This issue can be addressed by incorporating experimental and derived measures that correlate the functional state of a biological network with the extents of interactions, or 'connection strengths,' between the many molecular components (Patil and Nielsen, 2005). In recent years, analytical technologies have emerged enabling parallel measurements on the most common types of biochemical processes. For example, the DNA micro-array technology is now widely used to comprehensively profile the transcriptional activity of a gene network (di Bernardo, et al., 2005). Recent reports have also described the use of isotopomer modeling and metabolomic technologies for high-throughput analyses of metabolic reaction fluxes in intact cells (Fischer and Sauer, 2005).

In this application note, we describe an algorithm for data-driven modularity analysis, with the principal aim of disseminating the source code. A novel feature of this analysis is that it incorporates functional information on the interactions between the network's components. Our core algorithm extends the edge betweenness analysis algorithm (Newman and Girvan, 2004) to partition directed graphs with non-uniform edge costs. Additional components of our algorithm consist of well-known techniques for graph (Freeman, 1979) and vector space calculations. Our algorithm is general with respect to the type of connections between network components, and should be applicable to a variety of biological networks, such as transcriptional regulatory and protein-protein interaction networks. The inputs of the algorithm are an adjacency matrix describing the 'static' connectivity of the network components and a weight matrix describing the extents of interactions between these components. Here, we briefly describe the application of our modularity analysis to a metabolic network represented as a directed, compound graph with reaction edges, where the edge costs are supplied by metabolic profiling and flux analysis. In this analysis, the modularity of the network quantitatively reflects the connection diversity, i.e. reaction engagements, between the network components, i.e. metabolites.

2 BACKGROUND

Vertex betweenness centrality - A network is conveniently modeled as a graph *G{V,E}* consisting of a set of vertices (*V*) and edges (*E*), where each edge connects a pair of vertices (Cormen, et al., 2001). Vertex centrality refers to the significance of a vertex in determining the layout of the graph. There are three different measures of centrality based on degree, closeness, or betweenness. Among these, betweenness centrality has been shown to best reflect the variation in vertex centrality among distinguishable graphs (Freeman, 1979). Betweenness centrality is defined in terms of a probability. If $\sigma_{st}(v)$ is the number of the shortest paths (geodesics) from a vertex *s* to *t* that contains the vertex *v*, and σ_{st} is the number of shortest paths from *s* to *t*, then $b_{st}(v) = \sigma_{st}(v)/\sigma_{st}$ is the probability that vertex *v* falls on a randomly selected shortest path connecting *s* with *t*. The overall betweenness centrality of a vertex *v* is obtained by summing up its partial betweenness values for all unordered pairs of vertices $\{(s,t) \mid s,t \in V, s \neq t \neq v\}$:

$$C_B(v) = \sum_{s \neq v \neq t \in V} b_{st}(v) \qquad (1)$$

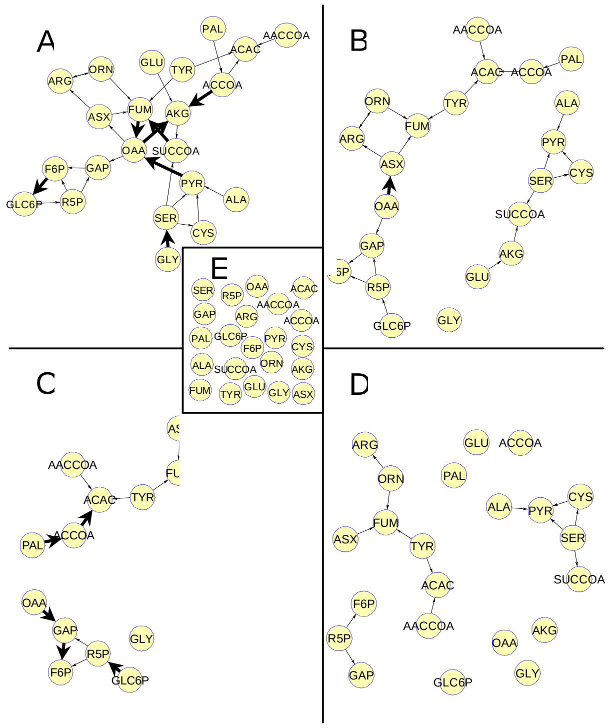
This index reflects the amount of control exerted by a given vertex over the interactions between the other vertices in the network. In general, the "vertex" betweenness centrality index is costly to compute for large networks. Recently, a faster algorithm has been developed applicable for large, but also very sparse networks, such as social networks (Brandes, 2001). We have adapted this faster algorithm to calculate the "edge" betweenness centrality index for a metabolic reaction network based on shortest paths.

Edge-betweenness centrality – Unlike many conventional clustering methods, which are agglomerative, the edge-betweenness algorithm is a

[*]To whom correspondence should be addressed.

top-down, divisive method for grouping network components into modules. Edge-betweenness centrality is the frequency of an edge that places on the shortest paths between all pairs of vertices. Analogous to Equation (1), the betweenness centrality of an edge in a network is given by the sum of the edge betweenness values for all source vertices. As suggested by (Newman and Girvan, 2004), the edges with highest betweenness values are most likely to lie between sub-graphs, rather than inside a sub-graph. Consequently, successively removing edges with the highest edge-betweenness will eventually isolate sub-graphs consisting of vertices that share connections only with other vertices in the same sub-graph.

3 ALGORITHM



s were represented as nodes, and reactions as edges. Edge weights were derived from reaction flux data as noted in the text. Figure panels show the network partitions generated at a few, selected stages of the algorithm: graph representation ('view') of the original network (A), view 5 (B), view 6 (C), view 8 (D), and view 9 (insert E). The view numbers refer to algorithm iterations. Modules were first observed at view 5. After 9 iterations (E), all edges were removed and all nodes separated. All programs were implemented in MATLAB (version 7.0.4, MathWorks, Natick, MA). The graph views were drawn using the Bioinformatics toolbox. Note that the drawn edges do not reflect the length adjustments supplied by the edge-weight matrix due to a limitation of the visualization software. The bold arrows highlight the highest betweenness edges removed in the subsequent iteration.

In its original implementation, which was developed for un-weighted, un-directed networks, the edge-betweenness analysis used the breadth-first search (BFS) algorithm (Newman and Girvan, 2004). Here, we extended this prior work to enable the edge-betweenness analysis of directed, weighted networks. The algorithm steps are as follows:

Step 1. Shortest paths through the network are calculated using Dijkstra's algorithm. Usually, the shortest paths differ for undirected and directed graphs. The shortest path calculation also critically depends on an edge weight matrix, which adjusts the relative distances between the network nodes (graph vertices) based on the strengths of the biochemical interactions represented by the corresponding edges. For example, a weight matrix holding metabolic flux data is used to adjust the distance between a pair of reactant and product nodes based on the activity of the intervening reaction. In the limiting case of an infinitesimal flux, the corresponding edge-cost is infinite, and thus unavailable to any shortest paths, reflecting a non-active component of a metabolic reaction network. In general, the dimensions and contents of the user-defined weight matrix (**W**) will depend on the available data.
ivity data is only partially available or altogether
our algorithm permits the assignment of a default
ix with uniform edge costs. The outputs of step 1 are: a
h number matrix (**Ssigma**), a predecessor matrix
d a shortest distance matrix (**Ddist**) (see supplementary

he edge-betweenness centrality index is calculated for
s previously suggested (Newman and Girvan, 2004).
ith the highest index value is removed, forming a new,
nodular, graph representation of the original network.
and 2 are repeated iteratively until no more edges
first iteration finds all possible shortest paths of the
twork, calculates the edge-betweenness values of each
ne network, and removes the highest betweenness edges.
first partition, the algorithm iterations recalculate the
paths and edge-betweenness index values subsequently.
ps 1 and 2 are performed over a variable space of O(n+m),
is the number of vertices and m is the number of edges. In
st case, steps 1 and 2 will require, respectively,
log$n$) and O(nm) steps per edge removal. At completion,
l edges have been removed, the algorithm will have
m×(O(nm+n^2log$n$) + O(nm)) steps.
illustration, we show the application of the algorithm to a
liver central carbon metabolism (Figure 1), for which
flux data had been previously obtained (Lee, et al., 2003).
tium 4 desktop with a CPU clock speed of 2.53 GHz and
RAM of 0.5 GB, the calculations for this example application required 9 iterations and lasted 8 to 9 seconds, indicating an average run time of 1 sec per iteration.

REFERENCES

Brandes, U. (2001) A faster algorithm for betweenness centrality, *J Math Soci*, **25**, 163-177.

Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C. (2001) *Introduction to algorithms*. The MIT press, Cambridge, Massachusetts.

di Bernardo, D., Thompson, M.J., Gardner, T.S., Chobot, S.E., Eastwood, E.L., Wojtovich, A.P., Elliott, S.J., Schaus, S.E. and Collins, J.J. (2005) Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks, *Nat Biotechnol*, **23**, 377-383.

Fischer, E. and Sauer, U. (2005) Large-scale in vivo flux analysis shows rigidity and suboptimal performance of Bacillus subtilis metabolism, *Nat Genet*, **37**, 636-640.

Freeman, L.C. (1979) Centrality in social networks: conceptual clarification, *Social Net*, 215-239.

Hartwell, L.H., Hopfield, J.J., Leibler, S. and Murray, A.W. (1999) From molecular to modular cell biology, *Nature*, **402**, C47-52.

Lee, K., Berthiaume, F., Stephanopoulos, G.N. and Yarmush, M.L. (2003) Profiling of dynamic changes in hypermetabolic livers, *Biotechnol Bioeng*, **83**, 400-415.

Newman, M.E. and Girvan, M. (2004) Finding and evaluating community structure in networks, *Phys Rev E Stat Nonlin Soft Matter Phys*, **69**, 026113.

Patil, K.R. and Nielsen, J. (2005) Uncovering transcriptional regulation of metabolism by using metabolic network topology, *Proc Natl Acad Sci U S A*, **102**, 2685-2689.

Sharan, R. and Ideker, T. (2006) Modeling cellular machinery through biological network comparison, *Nat Biotechnol*, **24**, 427-433.