BRIEFINGS IN FUNCTIONAL GENOMICS. VOL II. NO 6. 457–468 doi:10.1093/bfgp/els040

Spectral algorithms for heterogeneous biological networks

Martin McDonald, Desmond J. Higham and J. Keith Vass

Advance Access publication date 30 October 2012

Abstract

Spectral methods, which use information relating to eigenvectors, singular vectors and generalized singular vectors, help us to visualize and summarize sets of pairwise interactions. In this work, we motivate and discuss the use of spectral methods by taking a matrix computation view and applying concepts from applied linear algebra. We show that this unified approach is sufficiently flexible to allow multiple sources of network information to be combined. We illustrate the methods on microarray data arising from a large population-based study in human adipose tissue, combined with related information concerning metabolic pathways.

Keywords: *assortativity; eigenvector; Fiedler vector; Laplacian; meta-analysis; microarray; reordering; singular vector*

INTRODUCTION

Many computational tools are available to simplify and add value to large-scale biological networks. We focus here on *spectral methods*—those that use eigenvectors, singular vectors and generalized singular vectors to cluster or reorder nodes and to visualize patterns in a network. This is an area where essentially the same ideas can be developed and understood from many different viewpoints, including inference/machine learning [1], multi-dimensional scaling [2], graph drawing [3], image segmentation [4], principal component analysis/applied statistics [5] and support vector machines/kernel methods [6]. In this work, starting from first principles, we take a matrix computation/applied linear algebra perspective. In keeping with the aims of the journal, we focus on providing practical help and guidance to the nonspecialist in computerized methodology. After summarizing key ideas in the area, we show how the approach can be extended to the case of two related data sets. This type of meta-analysis issue is gaining importance as the availability of high throughput data increases along with the demand to combine multiple sources of information [7–10].

In the next section, we describe two related networks, based on microarray expression and metabolic pathway data. The third section then motivates and discusses some spectral clustering tools for a single network, and illustrates their use on the microarray data. In the fourth section, we look at a new variant that can incorporate extra information. The technique is validated on synthetic data in fifth section and on the real gene data in the sixth section. The last section gives a summary and points to key challenges.

DATA SETS

Here, we briefly introduce two data sets containing distinct but related information about the behaviour of genes. These will be used in subsequent sections to provide concrete examples. We emphasize however, that the algorithms can be applied generally to complex networks in biology, and beyond.

Microarray

Microarrays measure the transfer, in an individual sample, from deoxyribonucleic acid (DNA, with ~30 000 genes) to messenger ribonucleic acid

Corresponding author. Desmond J. Higham, Department of Mathematics and Statistics, University of Strathclyde, Glasgow G1 1XH, UK. E-mail: d.j.higham@strath.ac.uk

Martin McDonald is a PhD student in Biomedical Engineering.

Desmond J. Higham has a background in stochastic computation. Tel: 0141 548 3716. Fax: 0141 548 3345.

J. Keith Vass is a biochemist and bioinformatician with a particular interest in studying cancer.

(mRNA) [11]. In this work, we illustrate the use of spectral algorithms on public domain adipose tissue microarray data from male Icelandic subjects in DECODE study GSE7965 [12, 13]. This study contained a cohort of 701 individuals. Because we were not interested in determining differences between the sexes, for this study, we selected only the male contingent which is 296 samples. These data may be regarded as a rectangular array whose (i, j) entry records the expression level of gene i in sample j, for 23 765 genes across 296 samples. More precisely, we use the absolute value of the expression data, so that all data entries are nonnegative; with this approach we treat under and over-expression as equivalent, on the grounds that both indicate a deviation from basal behaviour, [14]. Hence, a larger weight is taken to denote a higher level of activity. It is, of course, possible to retain the distinction between under/overexpression using a signed network [15].

Metabolic

Metabolism describes the process through which cells break down and reassemble food and other nutrients. Traditionally, a metabolic network consists of a collection of individual chemicals (the nodes) and their interactions (the edges) [16]. We will construct a different type of metabolic network to obtain information that may be merged with the microarray data described in earlier section. Through the knowledgebase of the Database for Annotation, Visualization and Integrated Discovery (DAVID [17, 18]), we have access to a list of genes with a corresponding KEGG (KEGG:- http://www.genome.jp/kegg/) metabolic pathway identifier—since some genes have end effects in metabolism. The KEGG database contains pathway maps representing interactions and reactions between molecules—these maps are available for individual metabolic processes (e.g. glycolysis). Each individual pathway is comprised of a number of interactions and reactions, and has a unique ID. The list we are using from DAVID associates each gene with a number of these pathway IDs, depending on where the gene products interact—though each gene is counted once per pathway regardless of the number of products present. Then, genes with a pathway ID in common will have product(s) that are involved in a common metabolic process. Using this information, we constructed a gene–gene co-incidence matrix whose (i, j) entry is a nonnegative integer recording the number of times genes i and j appear in the same pathway. The construction of the metabolic pathway matrix with its scaling are the subject of ongoing work. The connections in this analysis are built on counting the total number of edges for each gene—this favours hub genes. The suitability of this as a metric depends on the desired output and can be factored into the interpretation of the results. We treat this metabolic construction as additional information for the microarray data. Rather than containing data on magnitudes of reactions, our metabolic network illustrates how well-connected genes are with each other in the sense of metabolism. This opens up the potential to gain a new perspective on the use of microarray data to make statements about the metabolomics of a disease state.

Finally, to make the data sets compatible, we use only the 4567 genes that appear in both the microarray and metabolic networks.

Overall, we have (a) a non-negative real-valued 4567 by 296 array of gene expression data, and (b) a non-negative integer-valued 4567 by 4567 array of metabolic pathway co-incidence data (with 4.4% of entries nonzero, mean nonzero entry is 1.35 and maximum entry is 45).

SPECTRAL METHODS AND GRAPH LAPLACIANS

In this section, we motivate and explain how the Laplacian and normalized Laplacian can be used to find structure in a network. The next subsection introduces a key result from linear algebra: we refer to [19], and the references therein, for further details. For more general information about the field of spectral graph theory, we recommend [20, 21].

Rayleigh–Ritz theorem

The following lemma, which is a special case of the Rayleigh–Ritz Theorem [22, Theorem 4.2.2], will be used to justify the spectral algorithms

Lemma 1 Let $M \in \mathbb{R}^{N\times N}$ be a symmetric positive semi-definite matrix with eigenvalues $0 = \gamma_1 < \gamma_2 < \gamma_3 \leq \gamma_4 \leq \cdots \leq \gamma_N$, and corresponding eigenvectors $\mathbf{r}^{[1]}, \mathbf{r}^{[2]}, \ldots, \mathbf{r}^{[N]}$. Then the problem

$$\begin{aligned} &\min_{\mathbf{y}\in\mathbb{R}^N} \quad \mathbf{y}^T M\mathbf{y} \\ &\mathbf{y}^T\mathbf{r}^{[1]} = 0 \\ &\mathbf{y}^T\mathbf{y} = 1 \end{aligned} \tag{1}$$

is uniquely solved by $\mathbf{y} = \mathbf{r}^{[2]}$.

Proof. The matrix M has a spectral decomposition $M = R\Gamma R^T$, where $\Gamma \in \mathbb{R}^{N\times N}$ is diagonal with (i, i)th entry γ_i and $R \in \mathbb{R}^{N\times N}$ has jth column $\mathbf{r}^{[j]}$. The eigenvectors are mutually orthogonal, so we may take $R^T R = I$. Letting $\mathbf{z} = R^T\mathbf{y}$, the problem (1) becomes

$$\begin{aligned} \min_{\mathbf{z}\in\mathbb{R}^N} \quad & \mathbf{z}^T\Gamma\mathbf{z}. \\ \mathbf{z}^T R^T\mathbf{r}^{[1]} &= 0 \\ \mathbf{z}^T\mathbf{z} &= 1 \end{aligned}$$

The constraint $\mathbf{z}^T R^T\mathbf{r}^{[1]} = 0$ simplifies to $z_1 = 0$, so the problem becomes

$$\begin{aligned} \min_{\mathbf{z}\in\mathbb{R}^N} \quad & \sum_{i=2}^{N} \gamma_i z_i^2. \\ \mathbf{z}^T\mathbf{z} &= 1 \end{aligned}$$

Because $0 < \gamma_2 < \gamma_3 \leq \gamma_4 \leq \cdots \leq \gamma_N$, it is clear that $z_2 = 1$ and $z_i = 0$ for $i = 3, 4, \ldots, N$ uniquely solves the problem. Hence, we have $\mathbf{y} = \mathbf{r}^{[2]}$ as required.

Clustering and reordering

Let $A \in \mathbb{R}^{N\times N}$ be a symmetric matrix with non-negative entries. From a network perspective, we think of $a_{ij} = a_{ji} \geq 0$ as representing the pairwise similarity between nodes i and j, where a larger value indicates a greater similarity.

Suppose we wish to divide the vertices in two disjoint *clusters*, where a pair of nodes within a cluster are typically well connected and a pair of nodes in different clusters are not. One way to judge the quality of a partition is to count the total weights in the edges that span the two clusters. Introducing the indicator vector $\mathbf{y}$, so that $\gamma_i = -\frac{1}{2}$ if node i is in one set and $\gamma_i = \frac{1}{2}$ if node i is in the other, the total weight across the clusters may be written

$$\frac{1}{2}\sum_{i,j}(\gamma_i - \gamma_j)^2 a_{ij}. \tag{2}$$

In matrix-vector form, this expression becomes

$$\mathbf{y}^T(D - A)\mathbf{y}, \tag{3}$$

where $D \in \mathbb{R}^{N\times N}$ is the diagonal degree matrix with $D_{ii} = \deg_i$, and $\deg_i := \sum_j a_{ij}$ is the degree of node i. Asking for $\mathbf{y}$ to minimize this quantity is not reasonable, because it leads us to the trivial solutions $\gamma_i \equiv \frac{1}{2}$ and $\gamma_i \equiv -\frac{1}{2}$; that is, put all nodes into a single cluster. It therefore makes sense to add a balancing constraint that limits the mismatch between cluster sizes. In general, however, it is not feasible to tackle the discrete problem (3) directly, and hence it is standard practice to allow the γ_i to take any real values; thereby *relaxing* the problem. Using $\mathbf{y} \in \mathbb{R}^N$, a suitable balancing constraint is $\mathbf{y}^T\mathbf{1} = 0$, where $\mathbf{1} \in \mathbb{R}^N$ is the vector with all entries equal to one, and to avoid the trivial solution $\gamma_i \equiv 0$, we add the extra constraint $\mathbf{y}^T\mathbf{y} = 1$. This leads us to the optimization problem

$$\begin{aligned} \min_{\mathbf{y}\in\mathbb{R}^N} \quad & \mathbf{y}^T(D - A)\mathbf{y}. \\ \mathbf{y}^T\mathbf{1} &= 0 \\ \mathbf{y}^T\mathbf{y} &= 1 \end{aligned} \tag{4}$$

As we discuss further in the following subsection, Lemma 1 shows that this problem can be solved via a spectral decomposition; that is, by computing appropriate eigenvectors and eigenvalues.

At this stage, it is worth pointing out that after the relaxation step, where we move from $\gamma_i \in \{-\frac{1}{2}, \frac{1}{2}\}$ to $\mathbf{y} \in \mathbb{R}^N$, we are in the realm where each node is assigned a position on the real line. We can recover clusters by picking a threshold, such as 0, and assigning nodes to the same cluster if they lie on the same side of the threshold. However, rather than interpreting (4) as a problem that approximates a discrete analogue, we could use it as a starting point, and take the viewpoint that nodes are being mapped to points on the real line in such a way that nearby nodes are well connected, i.e. have many or strongly weighted connections between them. Because the solution of (4) may be expressed in terms of a spectral decomposition, this idea may be taken further. Using the fact that the power method iteration converges to a dominant eigenvector, we may argue that solving (4) is equivalent to placing the nodes on the real line in random locations and then iteratively 'shuffling' them, based on their pairwise affinities, until an equilibrium state is reached; see [23] for details.

Rather than taking a hard clustering approach through thresholding, it is also possible to use the real-valued solution $\mathbf{y}$ to relabel the nodes. In this way, a permutation vector $\mathbf{p} \in \mathbb{R}^N$ is constructed, whose components consist of the integers from 1 to N, so that node i gets mapped to position p_i, with

$$p_i \leq p_j \iff \gamma_i \leq \gamma_j. \tag{5}$$

In words, $\mathbf{y}$ places the nodes on the real line, and we relabel them according to their position, the left-most becomes node 1 and the right-most becomes node N. Returning to the matrix

interpretation of the data set A, we have equivalently performed a symmetric permutation that reorders the rows and columns of the matrix. Viewing the reordered matrix is often a very useful way to visualize interesting patterns in the data [10, 19, 24, 25].

In the syntax of the MATLAB language [26], reordering the matrix A using the eigenvector y is achieved by sorting the vector, `[a, p] = sort(y)`, and permuting the matrix, `A (p, p)`.

We also note that spectral reordering methods can be motivated from a different, but closely related, viewpoint, by assuming that the data are an instance of an appropriate random graph and seeking a node ordering that maximizes the likelihood of the network [27, 28].

Graph Laplacian

The matrix $D - A \in \mathbb{R}^{N \times N}$ appearing in (4) is known as the *graph Laplacian* matrix for the network. This symmetric positive semi-definite matrix has smallest eigenvalue 0 and corresponding eigenvector $\mathbf{1}$. We suppose that the network is connected (every pair of nodes may be joined by at least one set of edges with non-zero weights), in which case all other eigenvalues of the Laplacian are positive; see e.g. [29, 30]. We also suppose that there is a unique smallest non-zero eigenvalue, and order the eigenvalues so that $0 = \lambda_1 < \lambda_2 < \lambda_3 \leq \cdots \leq \lambda_N$. We denote the corresponding eigenvectors $\mathbf{v}^{[1]}, \mathbf{v}^{[2]}, \ldots, \mathbf{v}^{[N]}$. These are orthogonal, and we assume that they have Euclidean norms of unity. The eigenvector $\mathbf{v}^{[2]}$ corresponding to the first non-zero eigenvalue of the Laplacian plays an important role in many areas of graph theory and network science, and is referred to as the *Fiedler vector* [20, 21, 31].

It now follows from Lemma 1 that the solution of the relaxed problem (4) is given by the Fiedler vector, $\mathbf{v}^{[2]}$.

An alternative form of clustering and reordering

Next, we note that using the constraint $\mathbf{y}^T\mathbf{1} = 0$ in (3) aims to balance the number of nodes in each group. As an alternative, we may wish to quantify the size of each node i in terms of its degree, $\deg_i$, and aim to balance the overall size of the clusters. An appropriate balancing constraint is then $\mathbf{y}^T D\mathbf{1} = 0$. Further, rather than normalizing with $\mathbf{y}^T\mathbf{y} = 1$, so that all nodes are treated equally in terms of distributing the locations on the real axis, we may prefer $\mathbf{y}^T D\mathbf{y} = 1$, which encourages high degree nodes to be placed nearer the origin. From the reordering viewpoint, this may be interpreted as an attempt to reduce the influence of 'promiscuous' nodes, encouraging them away from the extremes of the ordering range. Such issues of calibration can be important when there is a high degree of variance among the interaction weights, a circumstance that is common for gene expression data. These two changes convert the relaxed problem (4) to

$$\begin{aligned} \min_{\mathbf{y} \in \mathbb{R}^N} \quad & \mathbf{y}^T(D - A)\mathbf{y}. \\ \mathbf{y}^T D\mathbf{1} &= 0 \\ \mathbf{y}^T D\mathbf{y} &= 1 \end{aligned} \tag{6}$$

Changing variable to $\mathbf{x} = D^{\frac{1}{2}}\mathbf{y}$, this problem becomes

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^N} \quad & \mathbf{x}^T D^{-\frac{1}{2}}(D - A)D^{-\frac{1}{2}}\mathbf{x}, \\ \mathbf{x}^T D^{\frac{1}{2}}\mathbf{1} &= 0 \\ \mathbf{x}^T\mathbf{x} &= 1 \end{aligned} \tag{7}$$

where we make the reasonable assumption that all node degrees are non-zero.

Normalized graph Laplacian

The matrix $D^{-\frac{1}{2}}(D - A)D^{-\frac{1}{2}} \in \mathbb{R}^{N \times N}$ appearing in (7) is known as the *normalized graph Laplacian*. Like the (unnormalized) Laplacian in earlier subsection, this symmetric positive semi-definite matrix has an eigenvalue 0 and, in the case of a connected graph, a unique smallest nonzero eigenvalue. The eigenvalues lie in the interval [0, 2], see e.g. [30], and we label them $0 = \mu_1 < \mu_2 < \mu_2 \leq \cdots \leq \mu_N$, with corresponding eigenvectors $\mathbf{w}^{[1]}, \mathbf{w}^{[2]}, \ldots, \mathbf{w}^{[N]}$. By construction, we have $\mathbf{w}^{[1]} = D^{\frac{1}{2}}\mathbf{1}/||D^{\frac{1}{2}}\mathbf{1}||$.

We refer to $D^{-\frac{1}{2}}\mathbf{w}^{[2]}$ as the *normalized Fiedler vector*. Lemma 1 now shows that the relaxed problem (7) is solved by $\mathbf{x} = \mathbf{w}^{[2]}$ and hence the required solution of (6) is the normalized Fiedler vector $\mathbf{y} = D^{-\frac{1}{2}}\mathbf{w}^{[2]}$.

At this stage, it is worth making a few points about the spectral approach.

(1) Eigenvalues and eigenvectors are invariant under permutation, in the sense that

$$A\mathbf{x} = \lambda\mathbf{x} \iff (PAP)P\mathbf{x} = \lambda P\mathbf{x}$$

for $\mathbf{x} \in \mathbb{R}^N$, $\lambda \in \mathbb{R}$ and any symmetric permutation matrix $P \in \mathbb{R}^{N \times N}$. (A symmetric permutation matrix is found by symmetrically permuting the rows of an identity matrix according to some permutation of the numbers 1 to N.)

It follows that spectral algorithms are oblivious to the way that nodes are labelled—e.g. relabelling the nodes, $A \mapsto PAP$, simply reorders the elements of the Fiedler vector accordingly, $\mathbf{v}^{[2]} \mapsto P\mathbf{v}^{[2]}$. As a consequence, when we test spectral algorithms on synthetic data where known structures have been deliberately created, it is reasonable to label the nodes of A in any convenient manner.

(2) Whether we use the vector $\mathbf{y} \in \mathbb{R}^N$ for hard clustering or for reordering, it is clear that we should be unconcerned about two types of transformation

translation: where $\gamma_i \mapsto \gamma_i + c$, for a constant c that is independent of i,

rescaling: where $\gamma_i \mapsto \alpha\gamma_i$, for a constant $\alpha \neq 0$ that is independent of i.

In particular, the map $\mathbf{y} \mapsto -\mathbf{y}$ coincides with swapping labels across the two clusters or to reversing the node ordering, and we note that unit eigenvectors are uniquely defined only up to a $\pm$ factor.

(3) Using the translation and scaling operations above, we can show that the same Fiedler vector solutions arise for a very wide range of balancing constraints—we do not need to ask for nearly equal cluster sizes in the original discrete formulation; see [19].

(4) Because a symmetric matrix has orthogonal eigenvectors, moving beyond the Fiedler cases and using the 'next best directions' $\mathbf{v}^{[3]}$, $\mathbf{v}^{[4]}, \ldots$and $D^{-\frac{1}{2}}\mathbf{w}^{[3]}$, $D^{-\frac{1}{2}}\mathbf{w}^{[3]}, \ldots$to cluster or reorder the data can reveal further information about the data; see [19].

Singular value decomposition

In the case of a bipartite network, we have two separate groups of nodes and the weight a_{ij} represents the pairwise affinity between node i in the first group and node j in the second group. If the groups contain M and N nodes, respectively, then $A \in \mathbb{R}^{M\times N}$. Spectral information is now contained in the Singular Value Decomposition (SVD)

$$A = U\Sigma V^T,$$

where $U \in \mathbb{R}^{M\times M}$ and $V \in \mathbb{R}^{N\times N}$ are orthogonal and $\Sigma \in \mathbb{R}^{M\times N}$ is diagonal with diagonal elements $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$, [32]. The columns of U and V are referred to as the *left* and *right singular vectors* of A, respectively.

Analogously to the development in earlier section, we may introduce two indicator vectors, $\mathbf{p} \in \mathbb{R}^M$ and $\mathbf{q} \in \mathbb{R}^N$, and consider the quantity

$$\frac{1}{2}\sum_{i,j}(p_i - q_j)^2 a_{ij}. \tag{8}$$

After adding appropriate constraints and relaxing to real-valued vectors $\mathbf{p}$ and $\mathbf{q}$, it may be shown that the left and right singular vectors of A can be used to reorder the two groups of nodes. Similarly, the SVD of the normalized data $D_{\text{out}}^{-\frac{1}{2}}AD_{\text{in}}^{-\frac{1}{2}}$ arises if we generalize the $\mathbf{y}^T D\mathbf{1} = 1$ alternative in (6). Here $D_{\text{out}} \in \mathbb{R}^{M\times M}$ and $D_{\text{in}} \in \mathbb{R}^{N\times N}$ are the diagonal in and out degree matrices; that is, $(D_{\text{out}})_{ii} = \sum_{j=1}^{N} a_{ij}$ and $(D_{\text{in}})_{jj} = \sum_{i=1}^{M} a_{ij}$. We refer to [25] for further details.

We note that the left and right singular vectors of $A \in \mathbb{R}^{M\times N}$ are equivalent to the eigenvectors of A^TA and AA^T, respectively, and this forms a natural bridge to the methods described in the earlier subsections. For example, we may regard the operation of forming A^TA as correlating across the second group of nodes to form a pairwise affinity matrix for the first group. A spectral method could then be applied directly to A^TA to cluster or reorder the first group. In the notation used earlier in this section, the columns of matrices U and V are eigenvectors of the matrices AA^T and A^TA, respectively.

Microarray application

We illustrate the SVD reordering approach on the microarray data described in the second section. In the language of the previous section, the matrix A is the rectangular array of microarray data with genes and patients as the rows and columns. Then, the matrices U and V provide the left and right singular vectors—preserving the rectangular form of the data gives us the ability to reorder the samples as well as the genes; this type of bi-clustering is commonly performed on microarray data [14, 25]. Here, we have $N = 4567$ genes and $M = 296$ samples.

Figure 1 shows the components of the left singular vector $\mathbf{u}^{[2]}$ in increasing order. Using this ordering for the genes, we therefore take the view that (a) nearby genes in this ordering exhibit similar behaviour and (b) genes at the ends of the ordering are the most significant in terms of driving the corresponding sample ordering.

The hormone leptin is known to control body weight, and leptin resistance is a good indicator for obesity [33]. Hence, the gene that codes for leptin is

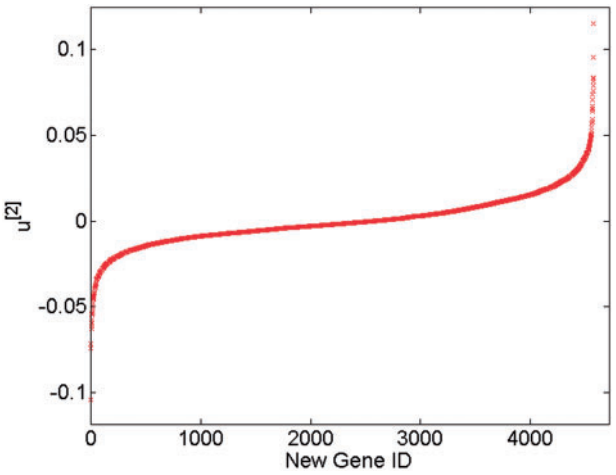

Figure 1: Components of the left singular vector $\mathbf{u}^{[2]}$, in increasing order, from an SVD of the microarray data.

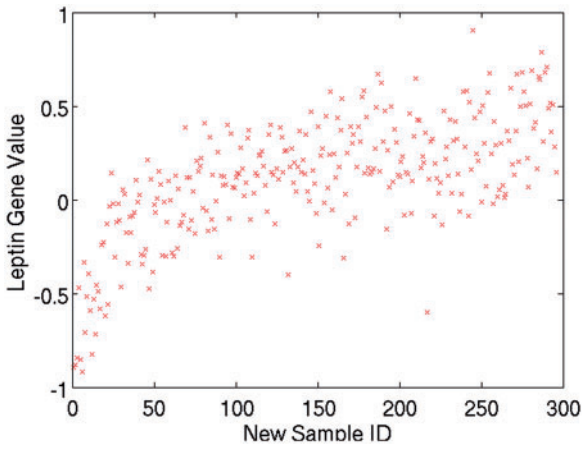

Figure 2: Leptin gene expression values from samples ordered by the right singular vector, $\mathbf{v}^{[2]}$.

of particular interest in this adipose tissue data set. In Figure 1, the gene that codes for leptin appears only three positions away from one end of the singular vector—this identifies the leptin gene as key to explaining the variance in the data set. In Figure 2, we show the expression level of the leptin gene across the samples, where the samples have been reordered according to the right singular vector, $\mathbf{v}^{[2]}$. As expected, there is a clear trend as we move across the ordered sample list. This type of sanity check based on prior information is a useful first step for validating the microarray data.

A NEW NORMALIZATION OF THE GRAPH LAPLACIAN

In the third section, we motivated spectral methods by setting up appropriately constrained optimization problems. This approach offers a lot of flexibility, a fact that we now exploit to derive an alternative Laplacian style matrix.

A network is said to be *assortative* if connections are more likely between nodes of similar degree (where the degree of a node is the number of edges that are connected to it) [34, 35]. Many authors have considered the issue of quantifying the overall level of assortativity in a network, relative to a null model [36, 37]. However, here we consider an inverse problem that also has practical relevance—given a network, can we identify specific patterns of assortativity? More precisely, can we find a set of nodes that

(a) form a strong cluster, and
(b) possess similar degrees?

where a strong cluster is a collection of nodes that shows significance in terms of weight density, as tested for by the cluster quality measure in the forthcoming section. We note that this is a partially local concept—it is possible for a *substructure* of this type to be present in a network that is not categorized as being assortative by a global measure. We also note that this type of substructure has a very natural generalization; the condition (b) could be extended to the case where nodes possess an independent measure of 'size' and we seek clusters that involve nodes of comparable size.

We therefore suppose that a positive weight w_i is associated with each node i. To look for nodes that are well-connected and size-compatible, we may replace the starting point (2) with

$$\frac{1}{2}\sum_{i,j}(\sqrt{w_i}\gamma_i - \sqrt{w_j}\gamma_j)^2 a_{ij}. \tag{9}$$

Letting $D_w \in \mathbb{R}^{N\times N}$ denote the diagonal matrix with iith entry w_i, this expression may be written

$$\mathbf{y}^T D_w^{\frac{1}{2}}(D-A)D_w^{\frac{1}{2}}\mathbf{y}. \tag{10}$$

Here, we emphasize that D is the original diagonal degree matrix arising from the data matrix, but the diagonal matrix D_w may contain any appropriate set of nodal weights.

To focus on the case where we prioritize nodes with large weights, we take $\mathbf{y}^T D_w^{-1}\mathbf{y} = 1$ as our normalizing constraint. This encourages the highly weighted nodes to take values at the extreme ends of the range of γ_i values. Changing variable to $\mathbf{z} = D_w^{\frac{1}{2}}\mathbf{y}$, the expression (10) then becomes

$$\mathbf{z}^T D_w(D-A)D_w\mathbf{z}, \tag{11}$$

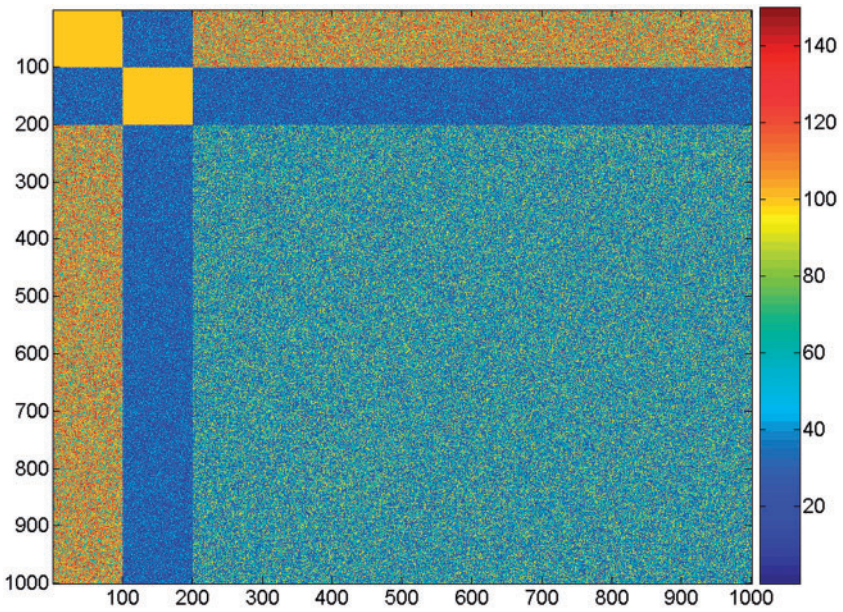

Figure 3: Weight matrix for the synthetic network, with colour bar.

with $\mathbf{z}^T\mathbf{z} = 1$.

We will refer to the matrix

$$L_w := D_w(D - A)D_w \tag{12}$$

appearing in (11) as the *node-weighted Laplacian*. By construction, L_w has a zero eigenvalue with corresponding eigenvector $D_w^{-1}\mathbf{1}$. This vector depends only on w-information; it ignores the network connectivity. Hence, by analogy with the Fiedler vector and normalized Fiedler vector approaches, we propose to reorder/cluster for this generalized notion of assortativity in terms of $\mathbf{x}^{[2]}$, the eigenvector of L_w corresponding to the smallest positive eigenvalue. From Lemma 1, this becomes the required minimizer of (11) if we add the balancing constraint $\mathbf{z}^T D_w^{-1}\mathbf{1} = 0$. In terms of the original variable, y, this balancing constraint is $\mathbf{y}^T D_w^{-\frac{1}{2}}\mathbf{1} = 0$, which further encourages highly weighted nodes away from the origin.

Converting back to $\mathbf{y} = D_w^{\frac{1}{2}}\mathbf{z}$, we therefore propose to take $D_w^{\frac{1}{2}}\mathbf{x}^{[2]}$ as our network reordering vector.

SYNTHETIC TESTING

We now illustrate the use of the new node-weighted graph Laplacian (12) on a synthetic network that is designed to contain an appropriate set of nodes. More precisely, we wish to test whether the node-weighted Laplacian can discover clusters whose nodes also have high degrees. The symmetric network adjacency matrix $A \in \mathbb{R}^{1000\times 1000}$ is shown in Figure 3. We have 1000 nodes, and initially all edge weights are assigned independently at random from a uniform $U(0, 100)$ distribution. We then force nodes 1–100 to become a strong cluster—the edge weights between these nodes are reset to 100. Then, to make the degrees of the nodes in this cluster higher than average, all edges between nodes in this cluster and the rest of the network are increased by 50. To make the test more challenging, another cluster comprising nodes 101–200 is also created in the data, involving nodes that have generally low overall degrees. Here, each edge in this group is reset to 100 and then the weights between this second cluster and the rest of the network are decreased by 50.

We emphasize that the network in Figure 3 is ordered in a natural manner, so that the substructure is readily visible. As mentioned in earlier section, spectral algorithms are invariant to the initial ordering of the data so we are free to choose one that allows for a simple assessment of the results.

The upper left picture in Figure 4 shows the nodes in their original ordering, along with the corresponding degrees. The upper right picture shows the resulting components in the Fiedler vector, $\mathbf{v}^{[2]}$, arising from the graph Laplacian. Similarly, the lower right picture shows the components of the normalized Fiedler vector, $D^{-\frac{1}{2}}\mathbf{w}^{[2]}$, arising from the normalized graph Laplacian. Neither vector distinguishes the high degree–high degree cluster formed by the first 100 nodes. The Fiedler vector treats the first 200

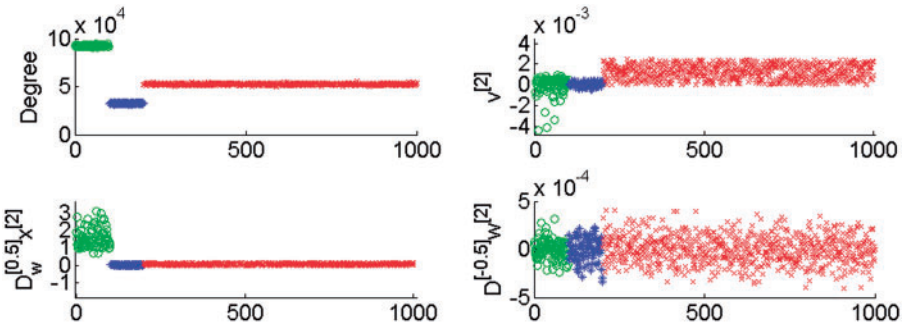

Figure 4: Synthetic network, original node ordering. Upper left: nodal degrees. Upper right: components of the Fiedler vector, $\mathbf{v}^{[2]}$. Lower right: components of the normalized Fiedler vector, $D^{-\frac{1}{2}}\mathbf{w}^{[2]}$. Lower left: components of the vector $D_w^{\frac{1}{2}}\mathbf{x}^{[2]}$ arising from the node-weighted graph Laplacian.

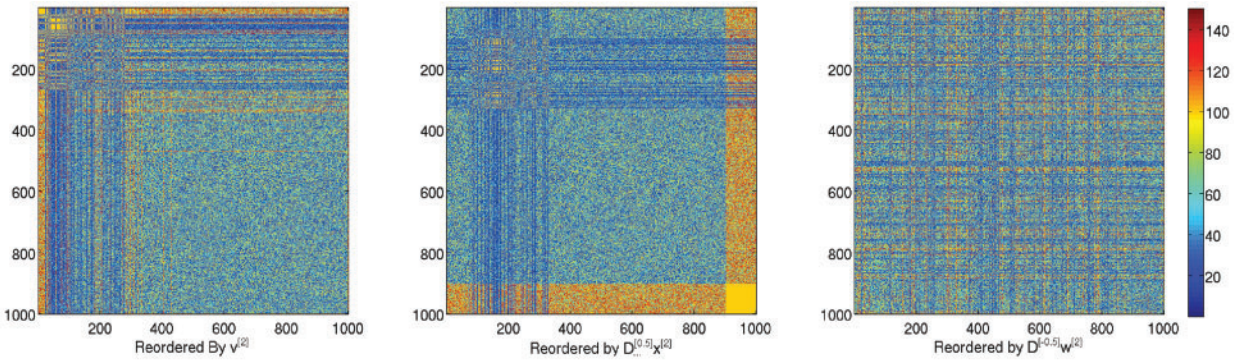

Figure 5: Heat maps for network reorderings applied to the synthetic network in Figure 3. Left: Laplacian. Middle: node-weighted Laplacian. Right: normalized Laplacian.

nodes similarly, with some overlap into the remaining 800, and the normalized Fiedler vector does not reveal any of the built-in structure. The lower left picture shows the values of $D_w^{\frac{1}{2}}\mathbf{x}^{[2]}$ arising from the node-weighted graph Laplacian. In this case, the first 100 nodes are clearly separated from the remainder.

Following on from Figure 4, the heat maps in Figure 5 show the network reordered according to Fiedler vector (left), normalized Fiedler vector (right), and by $D_w^{\frac{1}{2}}\mathbf{x}^{[2]}$ from the node-weighted Laplacian (middle). We see that only the middle picture reveals the cluster of strongly-weighted/high degree nodes.

In summary, this test shows that the node-weighted graph Laplacian (12) can reveal assortativity substructure that is not apparent from the more standard Laplacians.

Synthetic testing of merging two data sets

To complete the synthetic testing, we consider an additional case where D_w is constructed independently of the matrix A. This time D_w is given a range of values in order to test how much of the ordering of node weighted Laplacian is influenced by structure in A versus weights in D_w. The matrix A is constructed as in the previous example, and the first 50 values in D_w are given a high weight, 20, the next 50 given a low weight, 1. This pattern of 50 high and 50 low is repeated for the next 200 nodes; see the middle picture in Figure 6.

The three pictures in Figure 6 use the original, given network node ordering. The left picture shows the node degrees. The middle picture shows the new values we are using in D_w, from an artificial 'second' network. The right picture shows the reordering vector arising from the node weighted Laplacian. We see that this node weighted Laplacian reordering clearly picks out the first 50 nodes—those that were well-connected in matrix A and have high values in D_w. The 50 nodes that are well-connected but have low values in D_w are not separated, illustrating the fact that the result from the node weighted Laplacian uses a combination of both the information in the original network, and the values in the rescaling

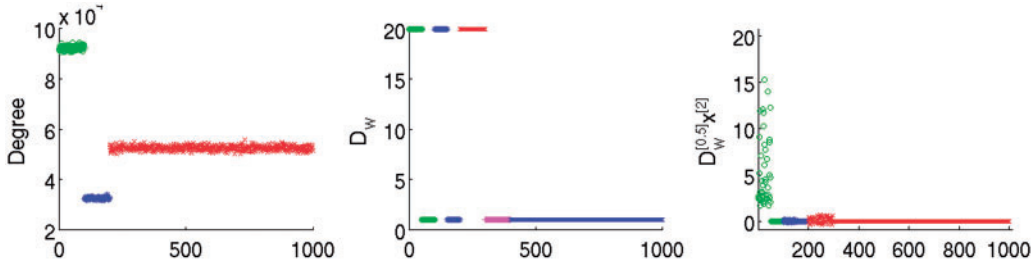

Figure 6: From the synthetic network in the original node ordering. Left: nodal degrees. Middle: components of D_w for the node weighted Laplacian. Right: components of the vector $D_w^{\frac{1}{2}}\mathbf{x}^{[2]}$ arising from the node-weighted graph Laplacian.

vector D_w. In addition to Figure 6, we also show the network reordered according to the node weighted Laplacian in Figure 7. We see here the strongly-weighted/high D_w nodes are pushed to the end of the ordering. The strongly-weighted/low D_w nodes have been pushed away from the end.

In summary, we have shown that the output from the node weighted Laplacian is influenced by both the original network and the values used for the components of D_w.

TEST COMBINING MICROARRAY AND METABOLIC NETWORKS

In this section, we further illustrate the use of spectral methods on the data described in second section. In particular, we show that the type of substructure targeted by the node-weighted graph Laplacian can be found in biological data.

In the notation of (12) our weighted, symmetric adjacency matrix $A \in \mathbb{R}^{4567\times4567}$ has the form MM^T, where $M \in \mathbb{R}^{4567\times296}$ is our 'gene expression against sample' microarray data. To incorporate the metabolic information, we let w_i be the overall degree of gene i in the metabolic network. We note that one difficulty in using w_i is the fact that the metabolic network is limited to the completeness of the database we use to construct it. Our aim is therefore to uncover well connected structures in the gene–gene microarray correlation network that are also strongly active in a metabolic sense.

In the left of Figure 8, we show the degree in the metabolic network, w_i, for the genes, when ordered by the vector $D_w^{\frac{1}{2}}\mathbf{x}^{[2]}$ from the node-weighted Laplacian. We see that the ordering gives preference to genes with high metabolic weights—placing them at the extremes of the list. By contrast, the right hand picture uses the Fiedler vector arising from the Laplacian matrix, which does not incorporate any metabolic information. Naturally, in this case, we do not see any metabolic pattern.

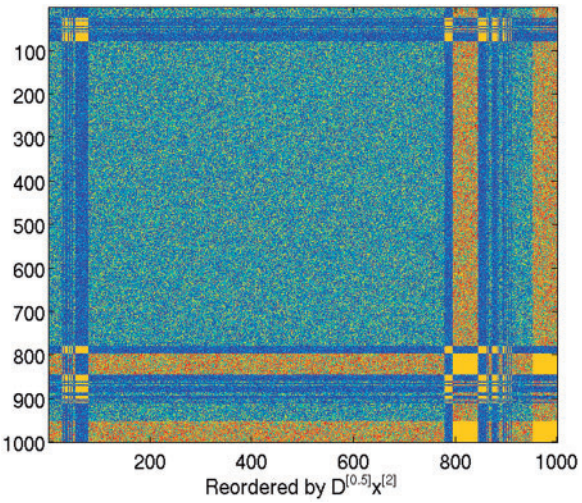

Figure 7: Heat map for network reordering applied to synthetic network in Figure 3, with nodal weights indicated in Figure 6, with the node weighted Laplacian.

Having confirmed that the metabolic information has affected the ordering, we now check whether $D_w^{\frac{1}{2}}\mathbf{x}^{[2]}$ has identified structure in the microarray data. We may do this by first inspecting the reordered microarray correlation matrix and choosing an appropriate range of contiguous nodes from the end of the ordering. In our case, 200 genes appeared to form a strong group. This is our putative cluster, whose quality can then be measured. There are, of course, many competing measures of cluster quality. Here, we follow the approach of [10], which can be summarized as

Step 1. Calculate the ratio of the average weight of edges in the cluster to the average weight of all other edges.

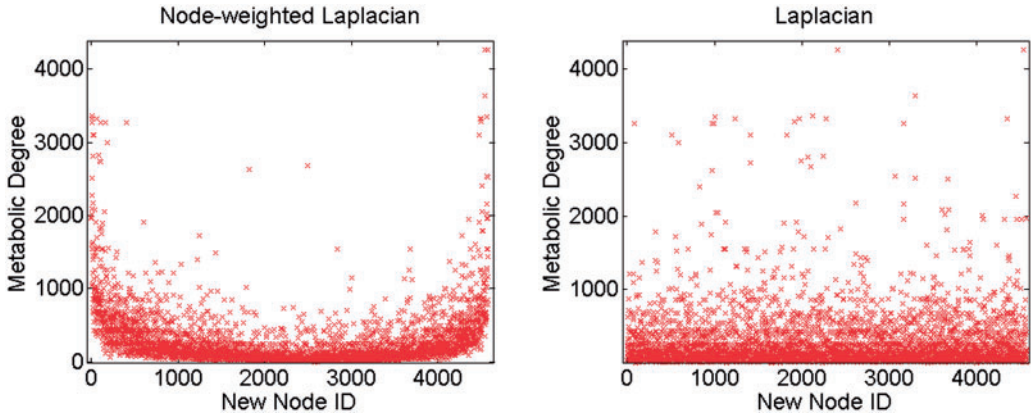

Figure 8: Metabolic degree of reordered genes. Left: ordered by vector from the node-weighted Laplacian. Right: ordered by the Fiedler vector.

Step 2. Randomize the order of the network, regard the first 200 genes as a cluster and calculate the ratio as in Step 1.

Step 3. Perform 999 repetitions of Step 2, and record as a '*P* value' the frequency with which the ratio for the randomized network exceeds that in Step 1.

In this way, the *P* value can be interpreted as the probability that a cluster of at least as highly quality as the discovered cluster would arise by chance.

Using this approach, the 200 gene cluster detected in the microarray data produced a *P* value below 0.01. Overall, this confirms that (a) the data contain a set of nodes with high expression correlation and high metabolic activity, and (b) the customized spectral approach was able to identify this structure.

Interpreting the results

Factorizing metabolic pathway data together with gene-expression data is a way of adding known large-scale biological information to the analysis. This approach does not attempt to prejudice the outcome, but asks if prior knowledge can add any useful information.

We are able to add a biological narrative to some of the observed genes that appear at both ends of the matrix. Along with leptin, a signalling molecule produced in adipose tissue, we find acyl-CoA oxidase 1, palmitoyl, the first enzyme in fatty-acid beta oxidation; malonyl-CoA decarboxylase, involved in both fatty-acid bio-synthesis or, more plausibly here, scavenging odd-length dicarboyxlic acid fatty-acids.

At the other end of the matrix, we find the gene for arginosuccinate lyase, traditionally linked to low food availability. This is implausible in this cohort, both from the social background and internally. Our analysis also finds ketohexokinase; the presence of this enzyme has been linked to a high fructose diet and its role is to use this sugar as both an energy source and, in adipose tissue, as source for precursors of fatty-acids. Ketohexokinase initiates the pathway through which most dietary fructose is metabolised [38, 39]. Traditionally, this was described as an energy store, but now is usually viewed as leading to undesirable fat and obesity. Fructose, in developed countries, is a common ingredient in most diets from the addition of corn syrup.

Our analysis has also led us to discover patterns with high probability of relevance to metabolic syndrome, obesity and type 2 diabetes. The availability of relevant biometric information would allow us to place these observations into more specific biological context.

DISCUSSION

Our aim was to motivate and illustrate spectral methods for network analysis. We used a first principles, linear algebra setting to show that by varying specific choices in the algorithm design, we can generate a range of spectral methods. In particular, we derived a simple, novel extension that can uncover assortative substructure. Due to space limitations, many issues have been omitted, so we finish by mentioning two key areas of current interest. First, for a large complex network, that is perhaps noisily defined, it may be of interest to identify substructures that go beyond simple clusters. For example, algorithms can be devised that discover subpatterns of bi-partivity [40], periodicity [27] or hierarchy [41], using spectral means. Second, a more systematic spectral approach for dealing with two or more related data sets can

be developed through the use of the Generalized Singular Value Decomposition [7–10].

Key Points

- Spectral algorithms in network science can be motivated naturally from a linear algebra perspective.
- The flexibity arising from this viewpoint allows for a variety of algorithms to emerge; in particular, a novel variant that can discover assortive subpatterns in an individual network and between pairs of networks by merging information from multiple sources.
- Such assortative subpatterns can be observed in real microarray/metabolic data sets.

FUNDING

M.M. is supported through an EPSRC-funded Doctoral Training Centre in Medical Devices. D.J.H. is supported by a Leverhulme Fellowship. J.K.V. is supported through an EPSRC-funded Knowledge Transfer Account at the University of Strathclyde.

References

1. Sra S, Nowozin S, Wright SJ. *Optimization for Machine Learning. Neural Information Processing Series*. Boston, MA: MIT Press, 2011.
2. Cox TF, Cox MAA. *Multidimensional Scaling*. London: Chapman and Hall, 1994.
3. Puppe T. *Spectral Graph Drawing: A Survey*. Saarbrücken, Germany: VDM Verlag, 2008.
4. Shi J, Malik J. Normalized cuts and image segmentation. *IEEE Trans Pattern Anal Machine Intelligence* 2000;**22**:888–905.
5. Wall ME, Rechtsteiner A, Rocha LM. Singular value decomposition and principal component analysis. In: Berrar DP, Dubitzky W, Granzow M (eds). *A Practical Approach to Microarray Data Analysis*. Vol. LANL LA-UR-02-4001. Kluwer, 2003, 91–109.
6. Rifkin R, Mukherjee S, Tamayo P, *et al*. An analytical method for multiclass molecular cancer classification. *SIAM Rev* 2003;**45**:706–23.
7. Lee CH, Alpert P, Benjamin O, *et al*. GSVD comparison of patient-matched normal and tumor aCGH profiles reveals global copy-number alterations predicting glioblastoma multiforme survival. *PLoS ONE* 2012;**7**:e30098.
8. Ponnapalli SP, Saunders MA, Van Loan CF, *et al*. A higher-order generalized singular value decomposition for comparison of global mRNA expression from multiple organisms. *PLoS ONE* 2011;**6**:e28072.
9. Schreiber A, Shirley NJ, Burton RA, *et al*. Combining transcriptional datasets using the generalized singular value decomposition. *BMC Bioinformatics* 2008;**9**:335.
10. Xiao X, Dawson N, Higham DJ, *et al*. Exploring metabolic pathway disruption in the subchronic phencyclidine model of schizophrenia with the generalized singular value decomposition. *BMC Systems Biol* 2011;**72**:5.
11. Eisen MB, Spellman PT, Brown PO, *et al*. Cluster analysis and display of genome-wide expression patterns. *Genetics* 1998;**95**(25):14863–8.
12. Emilsson V, Thorleifsson G, Zhang B, *et al*. Genetics of gene expression and its effect on disease. *Nature* 2008;**452**:423–8.
13. Vass JK, Higham DJ, Mudaliar MAV, *et al*. Discretization provides a conceptually simple tool to build expression networks. *PLoS ONE* 2011;**6**:e18634.
14. Kluger Y, Basri R, Chang JT, *et al*. Spectral biclustering of microarray cancer data: co-clustering genes and conditions. *Genome Res* 2003;**13**:703–16.
15. Higham DJ, Kalna G, Vass JK. Spectral analysis of two-signed microarray expression data. *IMA Mathematical Med Biol* 2007;**24**:131–48.
16. Lacroix V, Cottret L, Thebault P, *et al*. An introduction to metabolic networks and their structural analysis. *Comput Biol Bioinformatics, IEEE/ACM Trans* 2008;**5**:594–617.
17. Huang DW, Sherman BT, Lempicki RA. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature Protocols* 2009;**4**(1):44–57.
18. Huang DW, Sherman BT, Lempicki RA. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res* 2009;**37**:1–13.
19. Higham DJ, Kalna G, Kibble M. Spectral clustering and its use in bioinformatics. *J Comput Appl Math* 2007;**204**:25–37.
20. Chung F. *Spectral Graph Theory*. Providence, RI: American Mathematical Society, 1997.
21. Speilman DA. Spectral graph theory. In: Naumann U, Schenk O (eds). *Combinatorial Scientific Computing*. London: Chapman & Hall/CRC Computational Science, 2012: 495–524.
22. Horn RA, Johnson CR. *Matrix Analysis*. Cambridge: Cambridge University Press, 1985.
23. Grindrod P, Higham DJ, Kalna G, *et al*. DNA meets the SVD. *Mathematics Today* 2008;**44**:80–85.
24. Grindrod P. Range-dependent random graphs and their application to modeling large small-world proteome datasets. *Phys Rev E* 2002;**66**:066702–1 to 7.
25. Kalna G, Vass JK, Higham DJ. Multidimensional partitioning and bi-partitioning: analysis and application to gene expression datasets. *Int J Comp Math* 2008;**85**:475–485.
26. *Using MATLAB*. Natick, MA: The MathWorks, Inc. Online version.
27. Grindrod P, Higham DJ, Kalna G. Perodic reordering. The Institute of Mathematics and Its Applications (IMA). *J Numer Anal* 2010;**30**:195–207.
28. Higham DJ. Unravelling small world networks. *J Comp Appl Math* 2003;**158**:61–74.
29. Dhillon IS. Co-clustering documents and words using bipartite spectral graph partitioning. In: *Proceedings of the Seventh ACM SIGKDD Conference,* 2001.
30. Van Driessche R, Roose D. An improved spectral bisection algorithm and its application to dynamic load balancing. *Parallel Comput* 1995;**21**:29–48.
31. Fiedler M. Algebraic connectivity of graphs. *Czechoslovak Math J* 1973;**23**:298–305.

32. Golub GH, Van Loan CF. *Matrix Computations*. Baltimore, MD: Johns Hopkins University Press, 1996.
33. Buettner C, Muse ED, Cheng A, *et al*. Leptin controls adipose tissue lipogenesis via central, STAT3-independent mechanisms. *Nat Med* 2008;**14**:667–75.
34. Estrada E. *The Structure of Complex Networks*. Oxford: Oxford University Press, 2011.
35. Newman M. *Networks: An Introduction*. New York: Oxford University Press, 2010.
36. Newman MEJ. Mixing patterns in networks. *Phys Rev E* 2003;**67**:26–126.
37. Newman MEJ, Girvan M. Finding and evaluation community structure in networks. *Phys Rev E* 2004;**69**:26–113.
38. Diggle CP, Shires M, Leitch D, *et al*. Ketohexokinase: expression and localization of the principal fructose-metabolizing enzyme. *J Histochem Cytochem* 2009;**57**:763.
39. Basciano H, Federico L, Adeli K. Fructose, insulin resistance, and metabolic dyslipidemia. *Nutrition Metab* 2005; **2**:5.
40. Estrada E, Higham D, Hatano N. Communicability and multipartite structures in complex networks at negative absolute temperatures. *Phys Rev E* 2008;**78**(2):026102.
41. Crofts JJ, Higham DJ. Googling the brain: Discovering hierarchical and asymmetric network structures, with applications in neuroscience. *Internet Math* 2011; **7**(4):233–254.