Spectral Reordering of a Range-Dependent Weighted Random Graph

Desmond J. Higham[*]

July 6, 2004

Abstract

Grindrod [*Range-dependent random graphs and their application to modeling large small-world proteome datasets*, Physical Review E, 66, 2002] posed the problem of reordering a range-dependent random graph and showed that it is relevant to the analysis of datasets from bioinformatics. Reordering under a random graph hypothesis can be regarded as an extension of clustering and fits into the general area of datamining. Here, we consider a generalization of Grindrod's model and show how an existing spectral reordering algorithm that has arisen in a number of areas may be interpreted from a maximum likelihood range-dependent random graph viewpoint. Looked at this way, the spectral algorithm, which uses eigenvector information from the graph Laplacian, is found to be automatically tuned to an exponential edge density. The connection is precise for optimal reorderings, but is weaker when approximate reorderings are computed via relaxation. We illustrate the performance of the spectral algorithm in the weighted random graph context, and give experimental evidence that it can be successful for other edge densities. We conclude by applying the algorithm to a dataset from the biological literature that describes cortical connectivity in the cat brain.

keywords: bioinformatics, cortical connectivity, clustering, functional magnetic resonance imaging of the brain, genome datasets, Laplacian, maximum likelihood, small world networks, sparse matrix, two-sum.

AMS: 65F50, 68R10

[*] Department of Mathematics, University of Strathclyde, Glasgow G1 1XH, UK. Supported by a Research Fellowship from the Royal Society of Edinburgh/Scottish Executive Education and Lifelong Learning Department and by EPSRC grant GR/S62383/01.

1 Background

Grindrod [8, 9] recently defined a new class of *range-dependent random graphs* that generalize the Watts/Strogatz small world framework [14, 19] and have great potential as models for interaction networks in bioinformatics. Take nodes labelled $v_1, v_2, \ldots, v_N$ and suppose that an edge connecting v_i to v_j exists with independent probability $p_{ij} := g(|i-j|)$, for some suitable function g. Equivalently, the adjacency matrix $A \in \mathbb{R}^{N \times N}$ has $a_{ij} = 1$ with probability $g(|i-j|)$ and $a_{ij} = 0$ otherwise. In this framework there is an underlying ordering of the nodes, and the edge probabilities are functions of the "lengths" of the edges. In typical small world networks, short range connections ($|i-j|$ small) are much more frequent than long range connections ($|i-j|$ large) and hence range dependent random graph models where $g(k)$ decays rapidly as $k \to \infty$ are of interest. Grindrod suggested edge probabilities of the form $g(k) = \alpha\beta^{k-1}$, for $\alpha, \beta \in (0,1)$, and showed how various characteristics, such as the expected vertex degree, are easily derived. He also gave experimental evidence that these networks capture the connectivity structure seen in proteome interaction data.

In addition to providing a *model* for interaction data, Grindrod's work also suggested a fascinating *inverse problem* for extracting structural information. Given a network, that is, a set of nodes with all connections specified, how do we reorder the nodes so that any bias towards short range connections becomes apparent? In other words, how do we reorder a seemingly arbitrary network into one where the short-range/long-range connectivity patterns are revealed? Extracting this information is akin to, but more general than, clustering. In the bioinformatics context a suitable reordering reveals functional relationships and may drive further data collection. It gives answers to questions like:

- Which other genes/proteins behave like gene/protein X?
- What are the most likely false positives/false negatives in the data?
- Should a weakly observed interaction be classified as true or false?

A useful reordering algorithm must be robust with respect to erroneous data, including outliers—this rules out many methods from deterministic graph theory and sparse matrix computations.

In [10], we examined the potential of three sparse matrix reordering algorithms for unravelling random graphs. Here, we look at the most promising of these, spectral reordering, in the more flexible and general context of weighted edges.

The next section defines the general undirected range-dependent graph model in the case where the edge weights are drawn from continuous distributions. In section 2.1 we specify the inverse problem in maximum likelihood form, and reveal its connection to the minimum two-sum reordering problem. Section 2.2 deals with relaxed versions of these problems; here we find that the precise equivalence of the problems is lost. Section 3 shows how the theory carries through to the case where edge weights are drawn from discrete distributions, and also to the case of directed graphs. In section 4 we use a simulation approach to test the results on a range of graphs, and in section 5 we apply the reordering algorithm to a dataset arising from a biological application.

2 Continuous Weights

In this work we extend Grindrod's model to the case of an ordered graph with weighted edges. This allows for interaction data where, instead of "yes" or "no" quantification of similarity, there is a range of possible values, a generalization that is relevant to bioinformaticians, [1, 3, 5], and other bioscientists [11, 15]. For definiteness, we begin with undirected graphs and continuously weighted edges. We take the edge weights to be independent, non-negative, continuous random variables with range-dependent density. We remark that, in general, this leads to non-sparse networks. Our formal definition is as follows.

Definition 2.1 *Given a set of edge density functions* $\{f^{[k]}(x)\}_{k=1}^{N-1}$*, with* $f^{[k]}(x) = 0$ *for* $x < 0$*, the corresponding continuously weighted undirected range-dependent random graph has nodes ordered* $1, 2, 3, \ldots, N$ *with independent edge weights* $\{w_{ij}\}_{i<j}$*, such that* w_{ij} *has density* $f^{[j-i]}(x)$*.*

Grindrod has advocated the name *Renga*[1] for the range-dependent random graphs analysed in [8, 9]. Thus, following a referee's suggestion, we will use the phrase *C-Renga* in place of the more cumbersome "continuously weighted undirected range-dependent random graph" in Definition 2.1.

2.1 Reordering

We may now consider the reordering problem for a C-Renga. If, as would be typical in a bioinformatics application, the underlying range-dependent structure was believed to be present, but the nodes were ordered arbitrarily, can we find a reordering that reveals the structure? One way to formalize this question is to define an objective function that, given a particular ordering, measures how close the data is to C-Renga form. In Definition 2.2 we generalize the maximum likelihood approach of [8]; here $\mathcal{P}$ denotes the set of all permutation vectors, that is, all orderings of the integers $1, 2, \ldots, N$.

[1] A renga is a collection of haiku-like verses written by a network of poets. The word is both singular and plural.

Definition 2.2 *Given an instance of a C-Renga with edge densities* $\{f^{[k]}(x)\}_{k=1}^{N-1}$*, but with the nodes in arbitrary order, a* maximum likelihood reordering $p \in \mathcal{P}$ *solves*

$$\max_{p \in \mathcal{P}} \prod_{i<j} f^{[|p_i - p_j|]}(w_{ij}). \tag{1}$$

We note that the quantity to be maximized in (1) represents, for a given ordering $p \in \mathcal{P}$, the probability of that C-Renga arising.

Alternatively, given the general aim of having short-range edges with larger weights than long-range edges, we could reorder with respect to a two-sum objective function. This idea, which dates back to [4, 16], has been used as the basis for minimizing fill-in during sparse matrix computations.

Definition 2.3 *Given any undirected weighted graph with weights* $\{w_{ij}\}_{i<j}$*, a* minimum two-sum reordering $p \in \mathcal{P}$ *solves*

$$\min_{p \in \mathcal{P}} \sum_{i<j} (p_i - p_j)^2 w_{ij}. \tag{2}$$

Both maximum likelihood and two-sum reordering are hard, discrete optimization problems. However, as we mention in section 2.2, the two-sum version has a natural relaxation into a tractable eigenvalue computation. Definition 2.2 requires the C-Renga model, that is, the edge density functions $\{f^{[k]}(x)\}_{k=1}^{N-1}$, to be specified. The two-sum reordering in Definition 2.3, by contrast, makes no reference to a model. However, as Theorem 2.1 shows, it is possible to identify the two-sum problem as a special case of maximum likelihood.

Theorem 2.1 *A permutation* $p \in \mathcal{P}$ *solves the minimum two-sum reordering problem in Definition 2.3 if and only if it solves the maximum likelihood reordering problem in Definition 2.2 for the C-Renga model where* $f^{[k]}(x)$ *is the density function for the exponential distribution with parameter* k^2*; that is,* $f^{[k]}(x) = k^2 e^{-k^2 x}$*.*

Proof. Set $f^{[k]}(x) = k^2 e^{-k^2 x}$ in (1). Taking logs and negating converts the problem to

$$\min_{p \in \mathcal{P}} \left[\sum_{i<j} (p_i - p_j)^2 w_{ij} - 2 \sum_{i<j} \log |p_i - p_j| \right]. \tag{3}$$

Since the second summation inside the min is the same for all $p \in \mathcal{P}$, the problem is equivalent to finding a minimum two-sum reordering. □

Theorem 2.1 is of theoretical interest, as it relates an oft-quoted discrete optimization problem, two-sum reordering, to a new problem that is relevant in the context of random networks. However, as the next section shows, the connection is also of practical value, since, via relaxation, it provides an algorithm for approximate maximum likelihood reordering of a C-Renga.

2.2 Relaxed Reordering

For a large system, it is not feasible to examine all permutations $p \in \mathcal{P}$ in an attempt to optimize in (1) or (2). Grindrod [8] outlined a heuristic iterative algorithm that works directly with permutation vectors. Although the details of the algorithm were not given, impressive results were obtained on synthetic and real-life data.

Generalizing the approach in [10] we may investigate an alternative algorithm based on relaxing the problem to the continuous realm, where Lagrange multipliers can be exploited. This relaxation idea has proved successful for a number of applications in clustering, partitioning and datamining, [2, 4, 7, 17]. In order to define the relaxation process, we must also assume that the edge density expressions $\{f^{[k]}(x)\}_{k=1}^{N-1}$ continue to make sense when k is a positive real number. (We note that this is the case for the exponential distribution in Theorem 2.1.) In the following, $\mathbf{e}$ denotes the vector in $\mathbb{R}^N$ with all entries equal to one, and $\|\cdot\|_2$ denotes the Euclidean norm.

Definition 2.4 *Given an instance of a C-Renga with edge densities* $\{f^{[k]}(x)\}_{k=1}^{N-1}$*, but with the nodes in arbitrary order, a* relaxed maximum likelihood reordering $y \in \mathbb{R}^N$ *solves*

$$\max_{y\in\mathbb{R}^N,\, \|y\|_2=1,\, y^T\mathbf{e}=0} \prod_{i<j} f^{[|y_i-y_j|]}(w_{ij}). \tag{4}$$

Definition 2.5 *Given any undirected weighted graph with weights* $\{w_{ij}\}_{i<j}$*, a* relaxed minimum two-sum reordering $y \in \mathbb{R}^N$ *solves*

$$\min_{y\in\mathbb{R}^N,\, \|y\|_2=1,\, y^T\mathbf{e}=0} \sum_{i<j} (y_i - y_j)^2 w_{ij}. \tag{5}$$

Comparing Definitions 2.2 and 2.3 with Definitions 2.4 and 2.5, we see that relaxation involves weakening $p \in \mathcal{P}$ to $y \in \mathbb{R}^N$ and imposing the constraints $\|y\|_2 = 1$ and $y^T\mathbf{e} = 0$. We may recover a permutation $p \in \mathcal{P}$ from a componentwise ordering of y:

$$p_i \le p_j \Leftrightarrow y_i \le y_j. \tag{6}$$

The fact that such a p is invariant to scaling or shifting of y motivates the normalization $\|y\|_2 = 1$ and the orthogonality condition $y^T\mathbf{e} = 0$.

The relaxed two-sum problem has the advantage of being computationally tractable. It is easily shown that the objective function $\sum_{i<j}(y_i - y_j)^2 w_{ij}$ may be written $y^T(D-W)y$, where $W \in \mathbb{R}^{N\times N}$ has $(W)_{ij} = (W)_{ji} = w_{ij}$ and $D = \mathrm{diag}(\sum_i w_{ij})$. The matrix $D-W$ is known as the Laplacian of the graph (although this term is also used for a different, but related, matrix, [6]). By construction, $D-W$ is positive semi-definite and has a zero eigenvalue corresponding to the eigenvector $\mathbf{e}$. It follows from the Courant-Fischer Minimax Theorem, see for example, [12], that $y^T(D-W)y$, subject to $\|y\|_2 = 1$ and $y^T\mathbf{e} = 0$, is minimized by

finding the next smallest eigenvalue of $D-W$ and letting y be the corresponding normalized eigenvector. Computing a relaxed solution y and projecting back to a permutation p satisfying (6), the key idea of [4], is what we mean by spectral reordering. The vector y is often referred to as the *Fiedler vector* of W, and we note that special purpose software is available for its computation, [13].

Theorem 2.1 shows that the discrete formulations of maximum likelihood reordering (1) and minimum two-sum reordering (2) are equivalent, under $f^{[k]}(x) = k^2 e^{-k^2 x}$. We now give two simple examples to show that this equivalence is lost under relaxation. The effect arises because the second summation inside the min in (3) becomes significant.

Example 1 Consider the case

$$W = \begin{bmatrix} 0 & 1 & \alpha \\ 1 & 0 & 1 \\ \alpha & 1 & 0 \end{bmatrix}, \tag{7}$$

where $\alpha > 0$ is a parameter. The eigenvalue/eigenvector pairs of $D-W$ may be written $\{\lambda_i, v_i\}$, where

$$\lambda_1 = 0, \qquad \lambda_2 = 1 + 2\alpha, \qquad \lambda_3 = 3$$

and

$$v_1 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \qquad v_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}, \qquad v_3 = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}.$$

It follows that for $0 < \alpha < 1$, λ_2 is the second smallest eigenvalue, and the relaxed minimum two-sum reordering is found by taking $y = v_2$. Applying (6), this projects to the permutation $[1,2,3]^T$. It is clear that this is a true minimum two-sum reordering (2). For $\alpha > 1$, λ_3 is the second smallest eigenvalue of $D-W$, so $y = v_3$ gives the relaxed minimum two-sum reordering. Whichever way ties are broken in taking the projection (6), this corresponds to moving α in (7) away from the corner, which solves the discrete problem (2). However, because $y = v_3$ does not have distinct components, it cannot possibly be a relaxed maximum likelihood reordering, (4), under the exponential distribution in Theorem 2.1.

Example 2 To examine a less pathological case in practice, we perturb Example 1 to

$$W = \begin{bmatrix} 0 & 1.1 & 2 \\ 1.1 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix}. \tag{8}$$

MATLAB returned the permutation $[2,1,3]^T$ for the projected version of the relaxed minimum two-sum reordering. (This is a valid solution to the discrete problems (1) and (2).) To compute a relaxed maximum likelihood reordering, we minimized the negative log likelihood

$$\sum_{i<j}(y_i - y_j)^2 w_{ij} - 2\sum_{i<j}\log|y_i - y_j| \tag{9}$$

over $y \in \mathbb{R}^3$ of the form $y = \cos\theta\, v_2 + \sin\theta\, v_3$, where v_2 and v_3 are the eigenvectors corresponding to the middle and largest eigenvalues of $D - W$. Note that the relaxed minimum two-sum reordering occurs at $\theta = 0$. In Figure 1 we show how (9) varies for $\theta \in [0, \pi]$. The three local minima were found using MATLAB's `fminbnd` and the overall solution $\theta = 2.74$ produced a relaxed maximum likelihood reordering that projects to $[3, 1, 2]^T$ (which does not solve the discrete problems (1) and (2)). This example illustrates that *both the relaxed and the projected solutions may differ between two-sum and maximum likelihood formulations.*

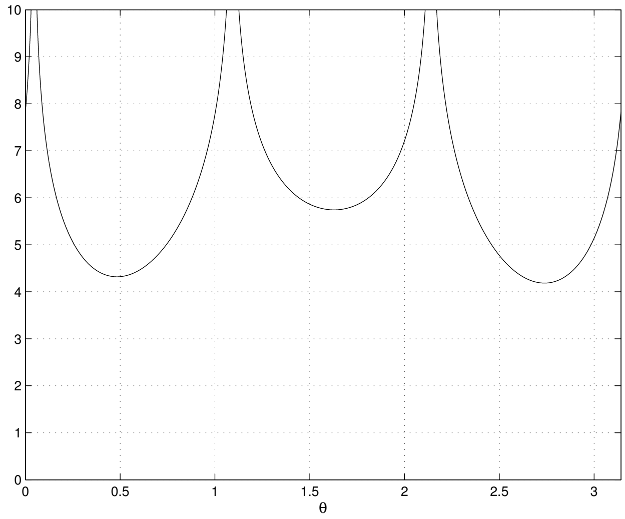

Figure 1: Negative log likelihood (9) for the example (8) using $y = \cos\theta\, v_2 + \sin\theta\, v_3$.

We also remark that the behaviour of relaxed minimum two-sum reordering in those examples is consistent with [2, Theorem 3.3], which shows that if it is possible to reorder so that entries along any row are non-increasing as we move away from the diagonal, then the Fiedler vector will find this reordering.

Examples 1 and 2 show that we cannot prove a direct analog of Theorem 2.1 for the relaxed problems. However, a weaker version does hold.

Theorem 2.2 *A vector $y \in \mathbb{R}^N$ solves the relaxed minimum two-sum reordering problem in Definition 2.5 if and only if it solves the relaxed maximum likelihood reordering problem in Definition 2.4 for a C-Renga model where*

$$f^{[k]}(x) = Ce^{-k^2 x}, \quad 0 \le x \le \max_{i<j} w_{ij},$$

for any constant C (independent of k).

Proof. The result can be proved in a similar manner to Theorem 2.1. □

We note that, given the data $\{w_{ij}\}$, a suitable density function for Theorem 2.2 can be constructed as

$$f^{[k]}(x) = \begin{cases} e^{-k^2 x}, & 0 \leq x \leq \max_{i<j} w_{ij}, \\ \epsilon, & \max_{i<j} w_{ij} < x \leq D_k, \\ 0, & D_k < x, \end{cases}$$

where ϵ and D_k are chosen to make $\int_0^\infty f^{[k]}(x)dx = 1$.

3 Discrete Weights and Directed Edges

An alternative to the C-Renga setting is to allow the range-dependent edge weights to be drawn from a discrete distribution. Below, we spell out how the material in section 2 converts to this case.

Definition 3.1 *Given a set of non-negative possible edge weights, $\{m_r\}_{r=1}^L$, and, for each range $1 \leq k \leq N-1$, a set of probabilities $\{f_r^{[k]}\}_{r=1}^L$ with $f_r^{[k]} \geq 0$ and $\sum_{r=1}^L f_r^{[k]} = 1$, the corresponding discretely weighted undirected range-dependent random graph has nodes ordered $1, 2, 3, \ldots, N$ with independent edge weights $\{w_{ij}\}_{i<j}$, such that $\mathbb{P}(w_{ij} = m_r) = f_r^{[j-i]}$.*

Following the remark after Definition 2.1, we will use the phrase *D-Renga* as an abbreviation for "discretely weighted undirected range-dependent random graphs" in Definition 3.1.

Definition 3.2 *Given an instance of a D-Renga where the possible edge weights $\{m_r\}_{r=1}^L$ have probabilities $\{f_r^{[k]}\}_{r=1}^L$, but with the nodes in arbitrary order, and writing $w_{ij} = m_{s_{ij}}$ for $i < j$, a* maximum likelihood reordering *$p \in \mathcal{P}$ solves*

$$\max_{p \in \mathcal{P}} \prod_{i<j} f_{s_{ij}}^{[|p_i - p_j|]}.$$

Theorem 3.1 *A permutation $p \in \mathcal{P}$ solves the minimum two-sum reordering problem in Definition 2.3 if and only if it solves the maximum likelihood reordering problem in Definition 3.2 for the D-Renga model where the edge weights w_{ij} are members of the set $\{m_r\}_{r=1}^L$ with probabilities*

$$f_r^{[k]} = \frac{e^{-k^2 m_r}}{\sum_{j=1}^L e^{-k^2 m_j}}.$$

Proof. The result can be proved in a similar manner to Theorem 2.1. □

Definition 3.3 *Given an instance of a D-Renga with possible edge weights $\{m_r\}_{r=1}^{L}$ and edge probabilities $\{f_r^{[k]}\}_{r=1}^{L}$, but with the nodes in arbitrary order, and letting $w_{ij} = s_{k_{ij}}$ for $i < j$, a* relaxed maximum likelihood reordering $y \in \mathbb{R}^N$ *solves*

$$\max_{y \in \mathbb{R}^N,\, \|y\|_2 = 1,\, y^T \mathbf{e} = 0} \prod_{i<j} f_{s_{ij}}^{[|y_i - y_j|]}.$$

Theorem 3.2 *A vector $y \in \mathbb{R}^N$ solves the relaxed minimum two-sum reordering problem in Definition 2.5 if and only if it solves the relaxed maximum likelihood reordering problem in Definition 3.3 for a D-Renga model where the edge weights w_{ij} are members of the set $\{m_r\}_{r=1}^{L}$ with probabilities*

$$f_r^{[k]} = C e^{-k^2 m_r},$$

where C is a constant (independent of k).

Proof. The result can be proved in a similar manner to Theorem 2.2. □

Given data w_{ij} from a set of edge values $\{m_r\}_{r=1}^{L}$, a suitable model for Theorem 3.2 can be constructed by setting $f_r^{[k]} = C e^{-k^2 m_r}$ for $1 \le r \le L$, where the constant C is chosen so that $\sum_{r=1}^{L} f_r^{[1]} < 1$, and then adding a dummy edge weight m_{L+1} and setting $f_{L+1}^{[k]} = 1 - \sum_{j=1}^{L} f_j^{[k]}$.

We also mention that this work extends readily to the case of directed graphs; that is, to unsymmetric W. The weights w_{ij} may be taken as i.i.d. with density $f^{[|j-i|]}(x)$ in Definition 2.1 and $\mathbb{P}\left(w_{ij} = m_r\right) = f_r^{[|j-i|]}$ in Definition 3.1. The corresponding maximum likelihood reordering and minimum two-sum reordering are given by (1) and (2), respectively, with $i < j$ replaced by $i \neq j$. All previous theorems remain valid. The relaxed problems are modified similarly, and a spectral reordering algorithm based on relaxation then uses the Laplacian of $\frac{1}{2}(W + W^T)$ instead of the Laplacian of W.

An unsymmetric W with discretely weighted entries is treated in section 5.

4 Computations on Weighted Random Graphs

We now present the results of some numerical tests on the spectral algorithm as a reordering tool for range-dependent random graphs. We focus on the case of continuously weighted edges—the next section deals with some real-life discretely weighted edge data. Our aims in this section are to test whether the relaxation technique produces good solutions to the underlying combinatoric problem, and to examine the robustness of the algorithm when a distribution other than the exponential distribution in Theorem 2.1 is used. We note that the type of tests performed in this section have also been found useful in measuring the performance of sparse matrix profile reduction algorithms, [13].

When computing with the algorithm and visualizing the results, it is natural to work with inverses of reordering permutations. Letting $p^{-1} \in \mathcal{P}$ denote the inverse of $p \in \mathcal{P}$, so that $p^{-1}_{p_k} = k$ and $p_{p^{-1}_k} = k$, we note the identity

$$\sum_{i,j} (p_i - p_j)^2 w_{ij} = \sum_{i,j} (i-j)^2 w_{p_i^{-1} p_j^{-1}}, \qquad \text{for any } p \in \mathcal{P}.$$

It follows that a reordering p in the sense of (2) may be regarded as replacing the matrix W by a new matrix whose i,jth entry is $w_{p_i^{-1}, p_j^{-1}}$. We also note that the MATLAB function `sort` directly computes the inverse, p^{-1}, of a p satisfying (6). For convenience, we use hats to denote permutations that represent inverses of reorderings, so $\widehat{p}$ corresponds to p^{-1}.

In Figure 2 we illustrate the spectral reordering algorithm on a C-Renga. We set $N = 20$ and generated weights a_{ij} for $i < j$ as samples from a pseudo-random number generator using an exponential distribution with parameter $(i-j)^2$. From Theorem 2.1, this is the natural range-dependency for the two-sum objective function. The upper left-hand picture shows the 20×20 symmetric weight matrix, M, using a grey-scale. The upper right-hand picture shows a shuffled version of this matrix. Here, a symmetric row and column permutation was applied—in other words, the nodes in the graph were swapped—using a permutation vector $\widehat{q} \in \mathcal{P}$. The resulting W, where $w_{ij} = a_{\widehat{q}_i, \widehat{q}_j}$, is the data matrix to which the algorithm was applied[2]. We computed the eigenvector y corresponding to the second smallest eigenvalue of the Laplacian, and used this to induce the inverse $\widehat{p}$ of a permutation $p \in \mathcal{P}$ satisfying (6). Applying this permutation to the shuffled matrix produced the matrix with entries $w_{\widehat{p}_i, \widehat{p}_j}$ shown in the lower left-hand picture. Comparing the two left-hand pictures, we see that the algorithm has successfully moved the large weights towards the diagonal (that is, associated them with short-range edges), although it has not recovered the original matrix precisely. In the lower right-hand picture we compare the arbitrary shuffling that we used to generate the data matrix with the unshuffling computed by the algorithm. To do this we plot the vector $\{\widehat{q}_{\widehat{p}_k}\}_{k=1}^{N}$. If the unshuffling were exact this would give points on a straight line of slope ±1. We see that the algorithm has done a reasonable job of reverse engineering the shuffle. Nodes that were neighbors in the original C-Renga are no more than two places apart in the computed solution. It is interesting to note that while p does a good job of unshuffling q, the converse is not true. We have

$$\begin{array}{rcl} \{\widehat{q}_{\widehat{p}_k}\} & = & \left[\begin{array}{cccccccccccccccccccc} 19 & 20 & 18 & 17 & 16 & 15 & 14 & 13 & 11 & 12 & 9 & 10 & 8 & 7 & 6 & 5 & 4 & 2 & 3 & 1 \end{array}\right], \\ \{\widehat{p}_{\widehat{q}_k}\} & = & \left[\begin{array}{cccccccccccccccccccc} 17 & 12 & 6 & 7 & 11 & 3 & 10 & 15 & 19 & 13 & 4 & 2 & 5 & 16 & 8 & 14 & 1 & 20 & 9 & 18 \end{array}\right]. \end{array}$$

[2] Because node reordering represents a similarity transformation of the graph Laplacian, the spectral algorithm would give the same solution (modulo tie-breaks) on the original and shuffled graphs. However, for the purpose of visualization we find it natural to display the shuffled version.

In our context it is the ordering of $\widehat{q}_{\widehat{p}_k}$ that matters—the task is to undo the original shuffle.

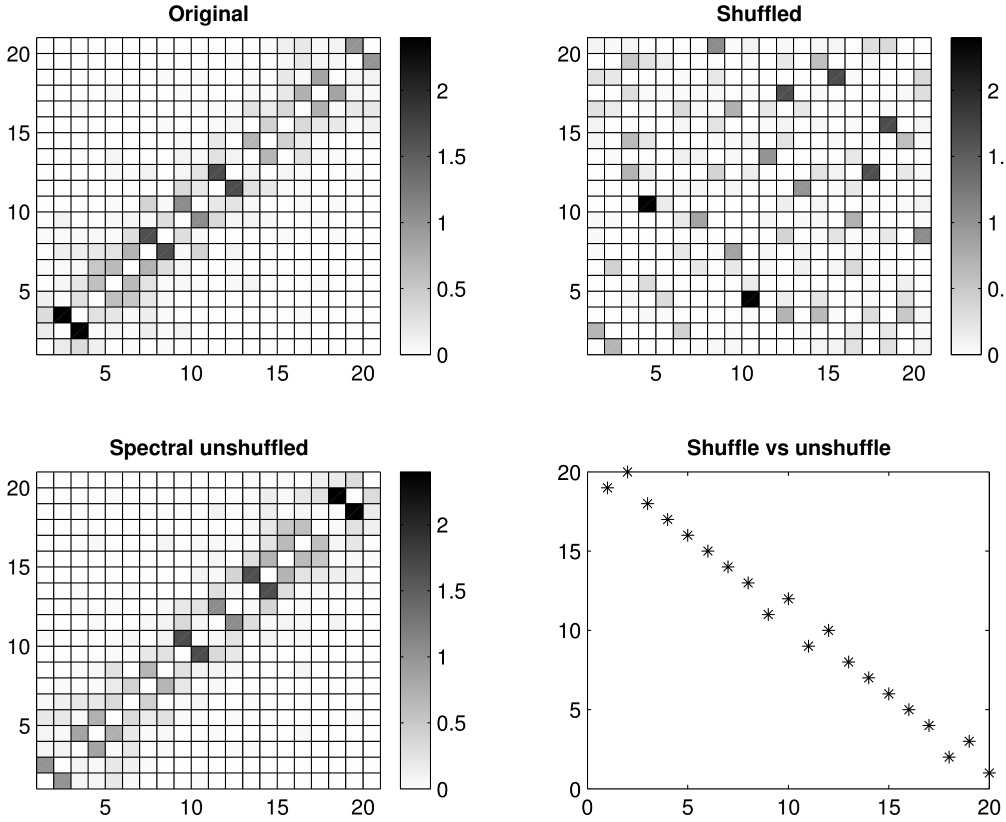

Figure 2: Upper left: C-Renga with exponentially distributed weights. Upper right: shuffled C-Renga. Lower left: unshuffled C-Renga from spectral algorithm, Lower right: comparison of shuffling and unshuffling.

Because the grey-scale pixel pictures become harder to interpret for bigger N, and because we want to report on a large batch of tests, we introduce two error measures. The first is

$$\text{perr} := \min\left(\max_{1\le k\le N}(|\widehat{q}_{\widehat{p}_k} - k|),\ \max_{1\le k\le N}(|\widehat{q}_{\widehat{p}_k} - (N-k+1)|)\right).$$

This is an indication of the accuracy of the unshuffle. The second is

$$\text{twosumerr} := \frac{\sum_{i<j}(i-j)^2\widehat{w}_{ij} - \sum_{i<j}(i-j)^2 w_{ij}}{\sum_{i<j}(i-j)^2 w_{ij}},$$

where $\widehat{W}_{ij} = w_{\widehat{p}_i,\widehat{p}_j}$ is the computed unshuffled matrix, illustrated by the lower left picture in Figure 2. Hence, twosumerr records the success of the algorithm in reducing the two-sum, relative to the two-sum of the original C-Renga. (Note that the two-sum also has a log-likelihood interpretation, as shown by Theorem 2.1.)

In the example of Figure 2 we had perr = 1 and twosumerr = −0.0177. The negative value for twosumerr indicates that the re-shuffled matrix has a smaller two-sum than the original.

Figure 3 illustrates the performance of the algorithm on 100 instances of a C-Renga of dimension $N = 1000$. As for Figure 2 we generated the weight matrices using an exponential distribution for w_{ij} with parameter $(i - j)^2$. The histogram in the upper picture shows how many of the 100 matrices were unshuffled with perr $= 0, 1, 2, 3, \ldots$. In the lower picture, for the same binning of matrices, we give the average over that bin of twosumerr. We see that all cases have a shuffle error of 1, 2 or 3. Experiments with other dimensions suggest that the performance generally improves as N is increased.

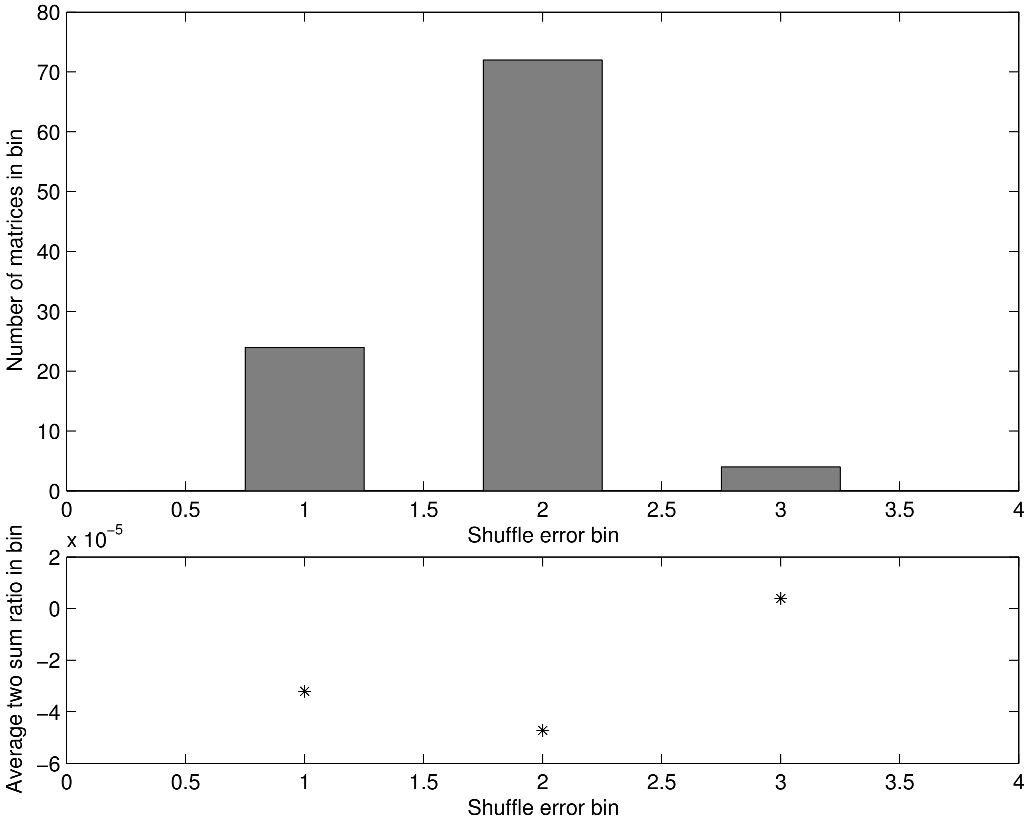

Figure 3: Performance of spectral algorithm on 100 instances of a dimension $N = 1000$ C-Renga with exponentially distributed weights. Upper plot: histogram of perr. Lower plot: average of twosumerr within each nonempty bin.

Next we consider weights that are fundamentally different from the ideal exponential distribution. First, we generate a_{ij} for $i < j$ as a sample from a uniform distribution over $(0, 1/(j-i))$. These edge densities no longer match the exponential densities in Theorem 2.1, and the mean edge weight at a distance k decays as $O(1/k)$ rather than $O(1/k^2)$. Figure 4 shows the perr and twosumerr values that arose. We see that the algorithm has continued to perform well. Figures 5 and 6 give the corresponding results for weights uniformly distributed over $(0, 1/(j-i)^\alpha)$, with $\alpha = 0.5$ and $\alpha = 0.25$, respectively. With α reduced to 0.5 the algorithm still does a reasonable job of recovering ordering information. As α is decreased the performance must, of course, degrade, and we see that with $\alpha = 0.25$ the reordered matrices differ significantly from the original data, although, on average, the two-sum is still well controlled.

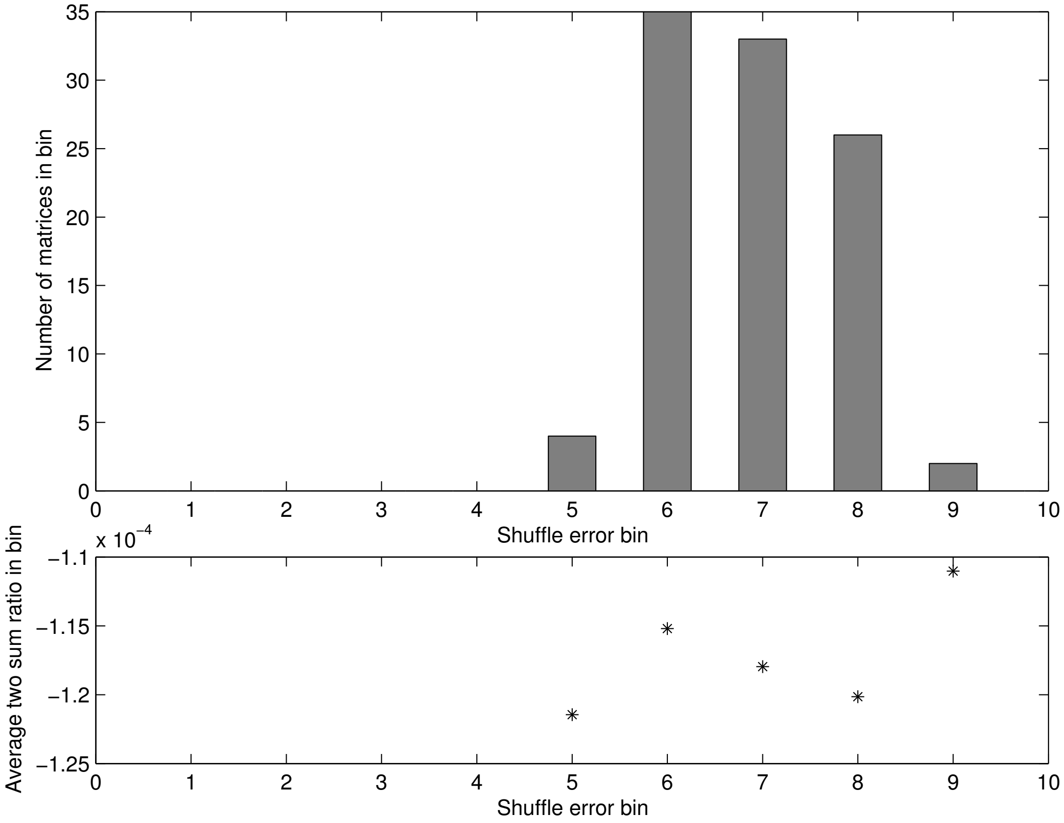

Figure 4: Similar to the computation in Figure 3, but with m_{ij} for $i < j$ drawn from a uniform $(0, 1/(j-i)^\alpha)$ distribution, with $\alpha = 1$. Dimension $N = 1000$.

5 Cat Brain Data

Figure 7 shows cortical connectivity data determined experimentally from the brain of a cat. This is taken from [11, Figure 1]. The entries have values 0, 1,

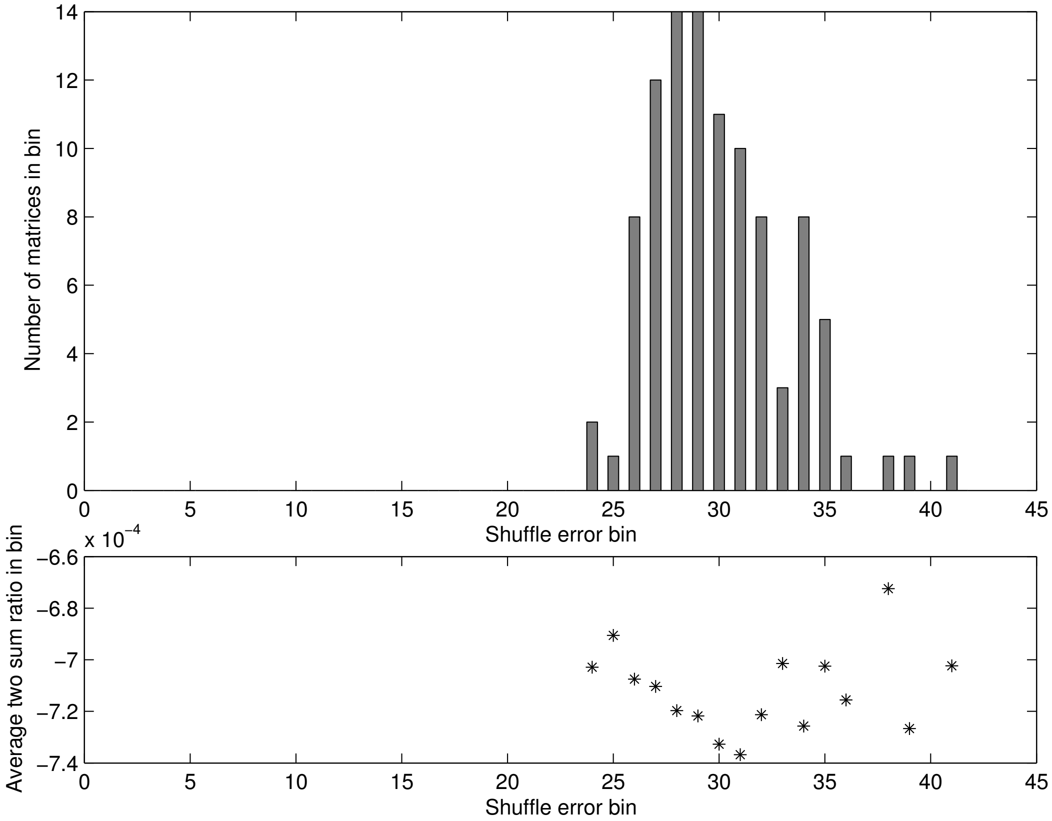

Figure 5: Similar to the computation in Figure 4, but with $\alpha = 0.5$.

2 or 3 and the matrix is unsymmetric. Here, a portion of the brain has been divided into regions, and the value of entry (i, j) indicates the strength of the connectivity between regions i and j. A key issue here is to pull together well-connected regions (which need not be geographically close); that is, to reorder the matrix so that large entries appear near the diagonal. This allows biologists to identify parts of the brain that are likely to be involved in related activities. The required information may be a clustering of the data [11] or, more generally, a reordering [15]. The authors in [11] defined a suitable objective function and used an evolutionary algorithm to search over the set of permutation vectors, an approach akin to that of Grindrod [8], and reveal clusters. Figure 7 shows their solution. Computation time for the algorithm on a DEC ALPHA 3000-600 UNIX workstation was reported to range “from several hours to a few days”.

In Figure 8 we show a corresponding spectral reordering. The data was symmetrized as described in section 3. As it involved only a single eigenvector computation, the spectral algorithm took a fraction of a second to run on a modern workstation. To obtain this solution, we experimented with various monotonic transformations of the $\{0, 1, 2, 3\}$ data, and settled on replacing each entry by its

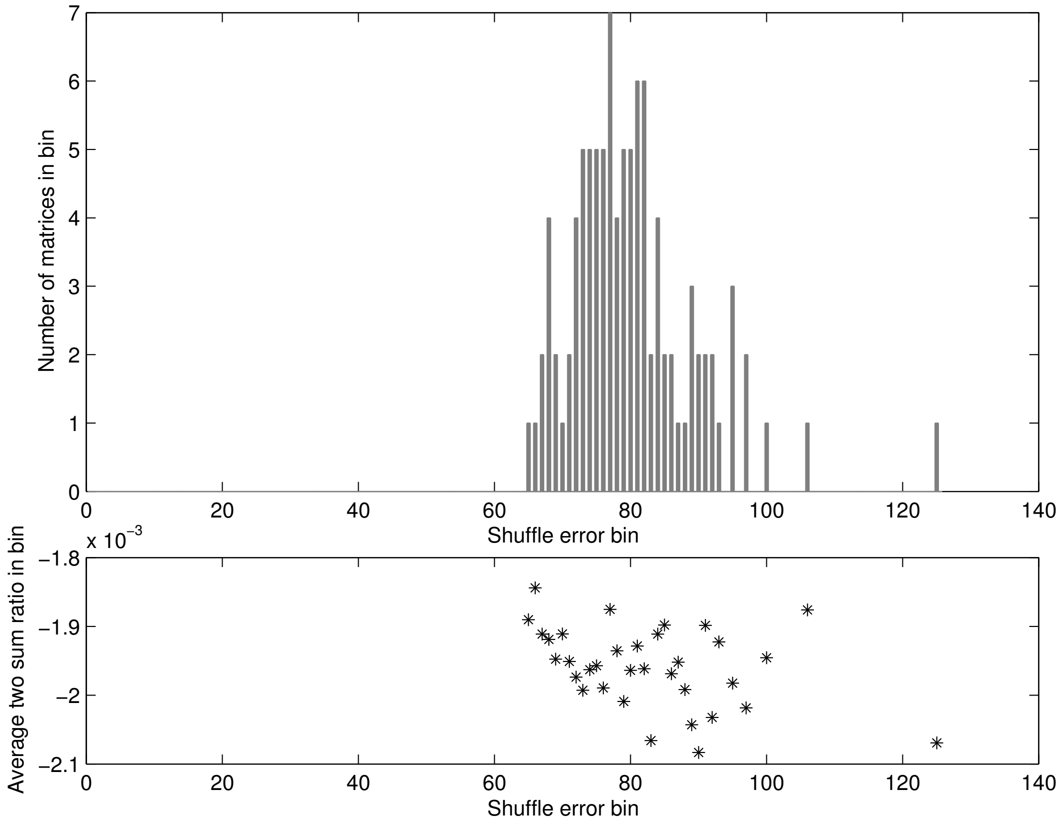

Figure 6: Similar to the computation in Figure 4, but with $\alpha = 0.25$.

3.4th power. Of course, the resulting reordering was applied to the original data. The question of whether there is an optimal way to preprocess a matrix before applying the spectral algorithm is an interesting area for future research.

Figure 9 compares the two solutions by plotting the inverse of the permutation that maps Figure 7 into Figure 8. We see that the broad features of the two solutions are similar—both suggest four highly inter-connected blocks consisting of similar components, although the precise orderings are quite distinct. The two-sums for the matrices in Figures 7 and 8 are 3.5×10^5 and 2.0×10^5, respectively. Of course, aside from using quantitative objective functions, such as the two-sum, the issue of "how good is a solution" would best be judged in terms of whether it extracts meaningful information for a biological scientist.

Overall, the spectral algorithm has the reassuring feature of reproducing biologically relevant information from [11], whilst having a low complexity that makes feasible the analysis of larger datasets, coming from more highly refined subdivisions. This observation is further strengthened by recent work reported in [15]. Here the spectral algorithm was successfully applied to 150×150 matrices that arise when diffusion imaging of the human brain is used to define connectivity networks relating to the visual system.

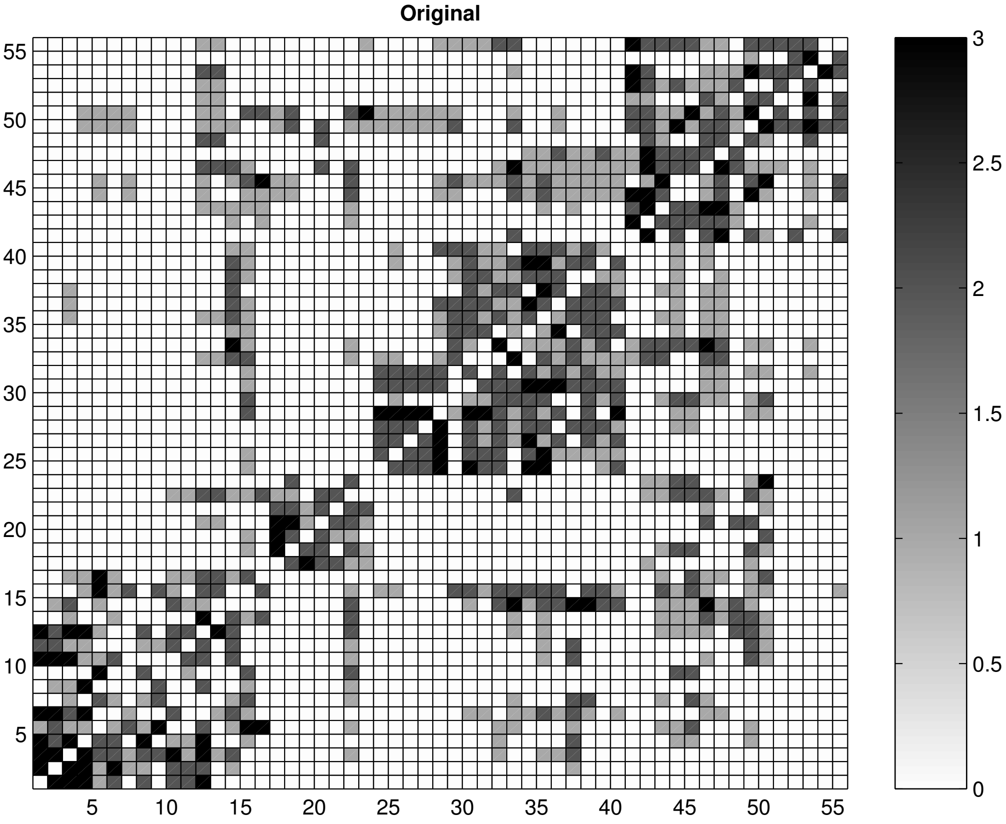

Figure 7: Original, ordered cat brain data from [11].

6 Summary and Conclusions

The two main thrusts of this work were

- to extend Grindrod's range-dependent random graph model to the practically relevant cases of weighted and undirected edges, and
- to show via analysis and numerical experiment that a spectral algorithm gives an effective means to tackle the corresponding reordering problem.

Given the current data deluge from bioinformatics [1, 9] and network biology [3], and, more generally, the high profile of complex networks across a range of applications, [18], the spectral algorithm's low complexity makes it an extremely promising tool for extracting meaning from large systems.

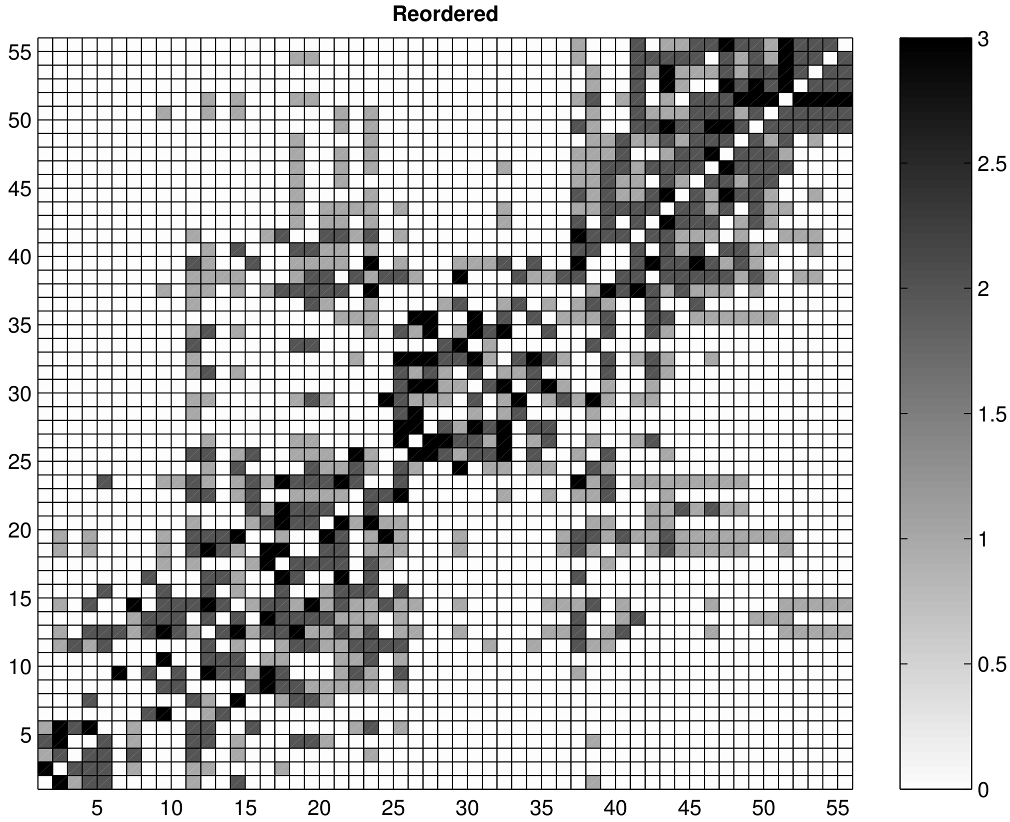

Figure 8: Spectrally reordered cat brain data.

Acknowledgements Much of this work was done while I visited the Centre for Advanced Study, Oslo, during the Special Year in Geometric Integration. I am grateful to Heidi Johansen-Berg and Tim Behrens of the Oxford Centre for Functional Magnetic Resonance Imaging of the Brain for pointing me towards the dataset used in section 5.

References

[1] R. B. Altman and S. Raychaudhuri, *Whole-genome expression analysis: challenges beyond clustering*, Current Opinion in Structural Biology, 11 (2001), pp. 340–347.

[2] J. E. Atkins, E. G. Boman, and B. Hendrickson, *A spectral algorithm for seriation and the consecutive ones problem*, SIAM J. Comput., 28 (1998), pp. 297–310.

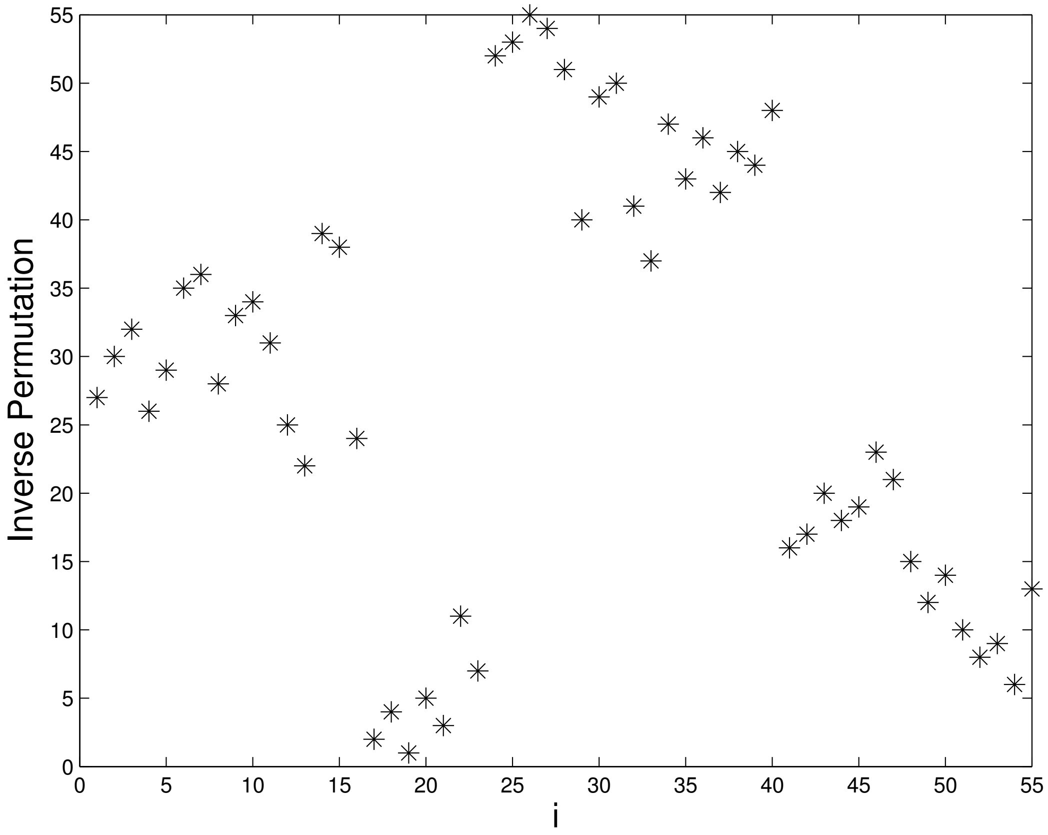

Figure 9: Inverse permutation from spectral reordering of cat brain data.

[3] A.-L. Barabási and Z. Oltvai, *Network biology: Understanding the cell's functional organization*, Nature Reviews Genetics, 5 (2004), pp. 101–114.

[4] S. T. Barnard, A. Pothen, and H. D. Simon, *A spectral algorithm for envelope reduction of sparse matrices*, Numerical Linear Algebra with Applications, 2 (1995), pp. 317–334.

[5] J. S. Chuang and D. Roth, *Gene recognition based on DAG shortest paths*, Bioinformatics, 1 (2001), pp. 1–9.

[6] F. Chung, *Spectral Graph Theory*, American Mathematical Society, Providence, RI, 1997.

[7] C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, *A min-max cut algorithm for graph partitioning and data clustering*, in Proceedings of the 1st IEEE Conference on Data Mining, 2001, pp. 107–114.

[8] P. Grindrod, *Range-dependent random graphs and their application to modeling large small-world proteome datasets*, Physical Review E, 66 (2002), pp. 066702–1 to 7.

[9] ——, *Modeling proteome networks with range-dependent graphs*, American Journal of PharmacoGenomics, 3 (2003), pp. 1–4.

[10] D. J. Higham, *Unravelling small world networks*, J. Comp. Appl. Maths, 158 (2003), pp. 61–74.

[11] C. C. Hilgetag, G. A. Burns, M. A. O'Neill, J. W. Scannell, and M. P. Young, *Anatomical connectivity defines the organization of clusters of cortical areas in the macaque monkey and the cat*, Phil. Trans. R. Soc. Lond. B, Biol. Sci., 355 (2000), pp. 91–110.

[12] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.

[13] Y. Hu and J. A. Scott, HSL_MC73*: A fast multilevel Fiedler and profile reduction code*, RAL-TR-2003-36, Numerical Analysis Group, Computational Science and Engineering Department, Rutherford Appleton Laboratory, 2003.

[14] H. Jeong, B. Tombor, R. Albert, Z. Oltval, and A.-L. Barabási, *The large scale organisation of metabolic networks*, Nature, 407 (2000), pp. 651–654.

[15] H. Johansen-Berg, T. E. J. Behrens, I. Drobnjak, S. M. Smith, D. J. Higham, and P. M. Matthews, *Connectivity based anatomical parcellation of cortical grey matter*, 10th International Conference for Functional Mapping of the Human Brain, June 2004, Budapest, Hungary. Available on CD-Rom in NeuroImage, (2004).

[16] M. Juvan and B. Mohar, *Optimal linear labelings and eigenvalues of graphs*, Disc. Appl. Math., 36 (1992), pp. 153–168.

[17] J. Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22 (2000), pp. 888–905.

[18] S. H. Strogatz, *Exploring complex networks*, Nature, 410 (2001), pp. 268–276.

[19] D. J. Watts and S. H. Strogatz, *Collective dynamics of 'small-world' networks*, Nature, 393 (1998), pp. 440–442.