BIOINFORMATICS ORIGINAL PAPER Vol. 27 no. 23 2011, pages 3293–3299
doi:10.1093/bioinformatics/btr558

Gene expression

Advance Access publication October 7, 2011

Computational network analysis of the anatomical and genetic organizations in the mouse brain

Shuiwang Ji

Department of Computer Science, Old Dominion University, Norfolk, VA 23529, USA

Associate Editor: John Quackenbush

ABSTRACT

Motivation: The mammalian central nervous system (CNS) generates high-level behavior and cognitive functions. Elucidating the anatomical and genetic organizations in the CNS is a key step toward understanding the functional brain circuitry. The CNS contains an enormous number of cell types, each with unique gene expression patterns. Therefore, it is of central importance to capture the spatial expression patterns in the brain. Currently, genome-wide atlas of spatial expression patterns in the mouse brain has been made available, and the data are in the form of aligned 3D data arrays. The sheer volume and complexity of these data pose significant challenges for efficient computational analysis.

Results: We employ data reduction and network modeling techniques to explore the anatomical and genetic organizations in the mouse brain. First, to reduce the volume of data, we propose to apply tensor factorization techniques to reduce the data volumes. This tensor formulation treats the stack of 3D volumes as a 4D data array, thereby preserving the mouse brain geometry. We then model the anatomical and genetic organizations as graphical models. To improve the robustness and efficiency of network modeling, we employ stable model selection and efficient sparsity-regularized formulation. Results on network modeling show that our efforts recover known interactions and predicts novel putative correlations.

Availability: The complete results are available at the project website: http://compbio.cs.odu.edu/mouse/

Contact: sji@cs.odu.edu

Supplementary Information: Supplementary data are available at Bioinformatics online.

Received on May 2, 2011; revised on October 1, 2011; accepted on October 3, 2011

1 INTRODUCTION

The mammalian central nervous system (CNS) generates high-level control functions, and knowledge on the anatomical and genetic organizations in this system can elucidate the functional brain circuitry. The enormous complexity of this system is reflected in the large number of cell types, each with unique gene expression patterns. Therefore, it is of central importance to capture the anatomical localization of gene expressions in the brain. Recent advances in bioimaging technologies, such as the high-throughput *in situ* hybridization (ISH) technique, have made it possible to capture the spatial expression patterns in the adult mouse brain (Lein *et al.*, 2007). Consequently, genomic-scale expression atlases in the form of digital images have been produced at increasing speed and resolution. The marriage of image processing tools and advanced computational methods opens the door for unraveling the functional brain circuitry and the generation of high-level cognitive functions on top of it.

The Allen Brain Atlas (ABA) (Lein *et al.*, 2007) contains 3D atlas of gene expression in the adult mouse brain and is one of the most comprehensive datasets for spatial expression patterns in the mammalian CNS. It provides cellular resolution 3D expression patterns in the male, 56-day-old C57BL mouse brain. In this atlas, genome-wide coverage is available in sagitally oriented sections. In addition, coronal sections at a more refined scale are available for a set of about 4000 genes showing restricted expression patterns. The image data are generated by *in situ* hybridization using gene-specific probes, followed by slide scanning, 3D image registration to the Allen Reference Atlas (ARA) (Dong, 2009) and expression segmentation (Lein *et al.*, 2007; Ng *et al.*, 2007). This results in a set of spatially aligned 3D volumes of size $67 \times 41 \times 58$, one for each gene, that document the spatial expression patterns of genes in the mouse brain. Efficient and effective analysis of these high-throughput data can shed light on the global function of mammalian CNS (Jones *et al.*, 2009). On the other hand, the sheer volume and complexity of these data pose significant challenges for efficient computational analysis. Hence, computational understanding of these data is limited to unsupervised techniques, which cluster the brain regions into co-expressed groups (Bohland *et al.*, 2010).

In this article, we employ advanced computational techniques to model the anatomical and genetic organizations in the mouse brain as networks. First, to reduce the size of data and accelerate efficient analysis and storage, we propose to apply tensor factorization techniques to reduce the data volumes (Kolda and Bader, 2009; Wrede, 1972). This tensor formulation treats the stack of 3D volumes as a 4D data array, thereby preserving the mouse brain geometry. Based on the reduced data, we model the anatomical and genetic organizations as graphical models in which each vertex represents a spatial location or a gene, and the edges between vertices encode the correlations between locations and genes (Dempster, 1972; Edwards, 2000). To improve the efficiency of network modeling, we employ an approximate formulation for Gaussian graphical modeling, which involves a series of sparsity regularized regression problems (Meinshausen and Bühlmann, 2006). The efficiency of this approximate formulation enables us to employ a robust estimation technique known as stability selection (Meinshausen and Bühlmann, 2010), which estimate and combine multiple models based on resampling.

We apply the data reduction and network modeling techniques to learn the anatomical and genetic networks underlying the mouse

brain using the ABA expression volume data. Results show that the expression patterns of spatially adjacent voxels tend to correlate. We also observe that the expression patterns of certain brain structures are correlated to the patterns of a large number of other regions, some of which are spatially distant. In-depth analysis reveals that such correlation patterns recover existing knowledge on the brain functionality. Our efforts on genetic network modeling identify functionally related genes that act in a concerted manner in the mouse brain.

2 HIGH-ORDER FEATURE EXTRACTION VIA TENSOR FACTORIZATION

In ABA, the ISH image series of each gene are aligned to the ARA. To faithfully capture the mouse brain geometry, a 3D grid is employed to divide the 3D ARA space into quadrats, and expression information within each quadrat is summarized. Specifically, an expression segmentation algorithm is employed to identify expressed cells, and then an expression energy value is computed from each voxel as a function of the intensity and density of expression within that voxel. These image processing steps convert each expression pattern into a 3D volume. To enable the application of matrix computation techniques such as the singular value decomposition (SVD), these volumes are usually converted to vectors and stacked into a data matrix (Bohland *et al.*, 2010). However, such conversion fails to retain the spatial locality and other high-order information in the expression volumes. To overcome this limitation, we propose to treat the 3D volumes as 3D tensors and stack them together to form a 4D tensor. We then employ tensor factorization techniques to reduce the dimensionality of this 4D tensor along each mode, resulting in significant data compression.

A key advantage of this tensor representation is that the associated tensor computation techniques, such as high-order SVD and low-rank tensor approximation, can be employed to compress the data without flattening the internal structure of the high-order data array. These techniques approximate the original tensor by a core tensor multiplied by a basis matrix along each mode. Hence, the core tensor and the set of basis matrices give a compact representation of the original tensor, and the core tensor captures the major information in the original tensor.

2.1 Background on tensors

Tensors, also known as multidimensional matrices (Kolda and Bader, 2009; Wrede, 1972), are higher order generalizations of vectors (first-order tensors) and matrices (second-order tensors). The order of a tensor is the number of indices, also known as modes or ways. In this article, tensors are denoted by boldface Euler script letters, e.g. $\mathscr{X} \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_N}$, and its elements are denoted as $x_{j_1, j_2, \ldots, j_N}$, where $1 \leq j_n \leq J_n$ for $n = 1, \ldots, N$. As a generalization of matrix multiplication, the n-mode tensor-matrix product defines the multiplication of a tensor by a matrix in mode n (Lathauwer *et al.*, 2000a). The n-mode product of a tensor $\mathscr{X} \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_N}$ with a matrix $A \in \mathbb{R}^{I \times J_n} = (a_{ij_n})$ is denoted by $\mathscr{X} \times_n A$. The result is a tensor of size $J_1 \times \ldots \times J_{n-1} \times I \times J_{n+1} \times \ldots \times J_N$ defined elementwise as

$$(\mathscr{X} \times_n A)_{j_1 \ldots j_{n-1} i j_{n+1} \ldots j_N} = \sum_{j_n=1}^{J_n} x_{j_1 \ldots j_{n-1} j_n j_{n+1} \ldots j_N} a_{i j_n}.$$

Let $\mathscr{Y} \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_N}$ be another tensor of the same size as $\mathscr{X}$. The scalar product of these two tensors is defined as:

$$< \mathscr{X}, \mathscr{Y} > = \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \cdots \sum_{j_N=1}^{J_N} x_{j_1, j_2, \ldots, j_N} y_{j_1, j_2, \ldots, j_N}. \quad (1)$$

Based on the scalar product, the Frobenius norm of a tensor $\mathscr{X}$ can be defined as

$$\| \mathscr{X} \| = \sqrt{< \mathscr{X}, \mathscr{X} >}. \quad (2)$$

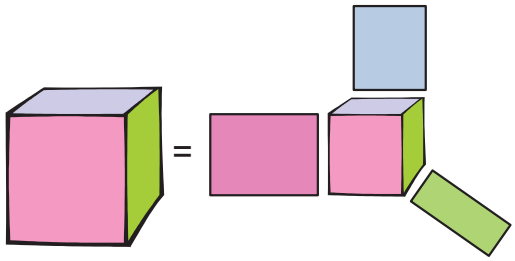

Fig. 1. Illustration of tensor factorization. The three-way tensor on the left is factorized into the products of a core tensor and three basis matrices on the right.

The mode-n vectors of $\mathscr{X}$ are the J_n-dimensional vectors obtained from $\mathscr{X}$ by varying index j_n while keeping all other indices fixed. Tensors can be converted into matrices via a process known as unfolding (Kolda and Bader, 2009). Specifically, the mode-n unfolding of $\mathscr{X}$ yields a matrix $X_{(n)} \in \mathbb{R}^{J_n \times (J_1 J_2 \ldots J_{n-1} J_{n+1} \ldots J_N)}$ whose columns consist of the mode-n vectors of $\mathscr{X}$. The mode-n rank of $\mathscr{X}$, denoted as $\text{rank}_n(\mathscr{X})$, is defined as the rank of the matrix obtained from mode-n unfolding of $\mathscr{X}$: $\text{rank}_n(\mathscr{X}) = \text{rank}(X_{(n)})$. Tensors have been used in a wide range of domains including microarray data analysis (Omberg *et al.*, 2007) and natural image modeling (Vasilescu and Terzopoulos, 2004; Wang *et al.*, 2005).

2.2 Tensor factorization

High-order singular value decomposition (HOSVD) (Lathauwer *et al.*, 2000a) is a generalization of the SVD for matrices. Given a tensor $\mathscr{X} \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_N}$, its HOSVD can be expressed as

$$\mathscr{X} = \mathscr{S} \times_1 U^{(1)} \times_2 U^{(2)} \times \cdots \times_N U^{(N)}, \quad (3)$$

where $\mathscr{S} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$, and $U^{(n)} \in \mathbb{R}^{J_n \times J_n}$, for $n = 1, \ldots, N$, are orthogonal matrices. In HOSVD, the basis matrices $\{U^{(n)}\}_{n=1}^N$ are computed as the left singular matrices of the mode-n unfolding of $\mathscr{X}$, and the core tensor can then be computed as

$$\mathscr{S} = \mathscr{X} \times_1 (U^{(1)})^T \times \cdots \times_N (U^{(N)})^T. \quad (4)$$

Given a tensor $\mathscr{X} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$, a rank-$(R_1, \ldots, R_N)$ factorization of $\mathscr{X}$ (Lathauwer *et al.*, 2000b) is formulated as finding a tensor $\hat{\mathscr{X}}$ with $\text{rank}_n(\hat{\mathscr{X}}) = R_n \leq \text{rank}_n(\mathscr{X})$ for $1 \leq n \leq N$ such that the following cost function is minimized:

$$\hat{\mathscr{X}} = \arg\min_{\hat{\mathscr{X}}} \| \mathscr{X} - \hat{\mathscr{X}} \|. \quad (5)$$

It follows from this definition that $\hat{\mathscr{X}}$ can be expressed as

$$\hat{\mathscr{X}} = \mathscr{C} \times_1 V^{(1)} \times_2 V^{(2)} \times \cdots \times_N V^{(N)}, \quad (6)$$

where $\mathscr{C} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ is called the core tensor and $V^{(n)} \in \mathbb{R}^{J_n \times R_n}$ $(1 \leq n \leq N)$ has orthonormal columns. When the basis matrices $\{V^{(n)}\}_{n=1}^N$ are given, the core tensor $\mathscr{C}$ can be readily computed as Lathauwer *et al.* (2000a)

$$\mathscr{C} = \mathscr{X} \times_1 (V^{(1)})^T \times_2 (V^{(2)})^T \times \cdots \times_N (V^{(N)})^T. \quad (7)$$

Hence, the key to the low-rank tensor factorization problem is to compute the basis matrices. The factorization of a 3D tensor is illustrated in Figure 1.

One of the commonly used algorithms to compute the basis matrices is the alternating least squares (ALS) method (Lathauwer *et al.*, 2000b). In each iteration of this method, one of the basis matrices is optimized while all others are fixed. Specifically, when $V^{(1)}, \ldots, V^{(n-1)}, V^{(n+1)}, \ldots, V^{(N)}$ are fixed, we first compute $\mathscr{X}_n = \mathscr{X} \times_1 (V^{(1)})^T \times \cdots \times_{n-1} (V^{(n-1)})^T \times_{n+1} (V^{(n+1)})^T \times \cdots \times_N (V^{(N)})^T$. Then the columns of $V^{(n)}$ can be obtained as the first R_n columns of the left singular matrix of $(\mathscr{X}_n)_{(n)}$, which is the mode-n unfolding of $\mathscr{X}_n$. In ALS, the basis matrices are usually initialized as the truncated basis

matrices from HOSVD (Lathauwer *et al.*, 2000b). That is, $V^{(n)}$ is initialized as the first R_n columns of $U^{(n)}$, for $n=1,\ldots,N$. When the size of the tensor is very large and cannot fit into memory, an out-of-core algorithm can be applied by partitioning the tensor into blocks (Wang *et al.*, 2005).

The advantages of tensor-based methods in comparison to matrix-based approaches have been addressed in the literature (Omberg *et al.*, 2007; Vasilescu and Terzopoulos, 2004; Wang *et al.*, 2005). In summary, tensor-based methods have the following two major advantages: (i) tensor-based methods can be applied to large datasets for which matrix-based methods are too expensive to apply. For example, the size of the data array for genetic network modeling in this article is $3012\times67\times41\times58$. While the tensor-based method requires the SVD of three matrices of sizes 67×67, 41×41, and 58×58, respectively, the matrix-based method requires the SVD of a matrix of size $3012\times159,326$. (ii) Although matrix-based methods give the lowest reconstruction error due to the best low-rank approximation property of matrix SVD, tensor-based methods preserve the geometry of the high-order data array. In the literature, tensor-based and matrix-based methods have been compared in classification tasks (Ye, 2005). Specifically, it has been shown that, though tensor-based methods give larger reconstruction error, they usually yield higher classification accuracy.

3 NETWORK CONSTRUCTION VIA SPARSE MODELING

The 4D tensor of gene expression obtained from the ABA is factorized as described above. The core tensor retains most of the information in the original tensor while its size is significantly reduced. This data reduction step is critical for the subsequent efficient analysis. Based on the reduced data, we employ sparse graphical modeling approaches to construct the anatomical and genetic networks underlying the mouse brain.

3.1 A sparsity regularization formulation

Gaussian graphical models are a class of methods for modeling the relationships among a set of variables (Edwards, 2000; Whittaker, 1990). In this formulation, the d-dimensional variable $\mathbf{x}=[x_1,x_2,\ldots,x_d]^T$ is assume to follow a multivariate Gaussian distribution $\mathbf{x}\sim N(\mu,\Sigma)$, where $\mu\in\mathbb{R}^d$ and $\Sigma\in\mathbb{R}^{d\times d}$ are the mean and covariance, respectively. The conditional dependency between pairs of variables can be encoded into a graphical model in which vertices represent variables and edges characterize the conditional dependency between variables. In particular, there is an edge between nodes corresponding to x_i and x_j if and only if these two variables are conditionally dependent given all other variables. This is equivalent to the saying that there exists an edge between nodes corresponding to x_i and x_j if and only if the (i,j)-th entry of the inverse covariance matrix (also known as concentration matrix) $\Omega=\Sigma^{-1}$ is non-zero (Dempster, 1972; Edwards, 2000). This correspondence is illustrated in Figure 2.

Given a set of n observations $\mathbf{y_1},\mathbf{y_2},\ldots,\mathbf{y_n}$, the concentration matrix can be estimated by maximizing the penalized log likelihood as follows (Banerjee *et al.*, 2008; Friedman *et al.*, 2008; Yuan and Lin, 2007):

$$\hat{\Omega}=\arg\max_{\Omega\succ0}\log\det\Omega-\mathrm{trace}(S\Omega)-\lambda\|\Omega\|_1, \tag{8}$$

where $\det\Omega$ is the determinant of Ω, $\Omega\succ0$ represents that Ω is positive definite, S denotes the empirical covariance matrix computed from data, and $\|\Omega\|_1$ is the 1-norm of Ω, which is the sum of the absolute values of the entries of Ω. The first two terms in Equation (8) are the log likelihood, and the last term is used to enforce that many entries of Ω are set to zero, yielding a sparsely connected graph. This formulation has been used to model the gene networks in *Arabidopsis thaliana* (Wille *et al.*, 2004). The optimization problem in Equation (8) is convex and can be solved by several algorithms such as the interior point method (Banerjee *et al.*, 2008) and the graphical lasso algorithm (Friedman *et al.*, 2008). However, all these algorithms are computationally expensive and can only be applied to small-scale problems. For the modeling of mouse brain networks, we have thousands of genes and tens of thousands of voxels; hence, this formulation is not applicable.

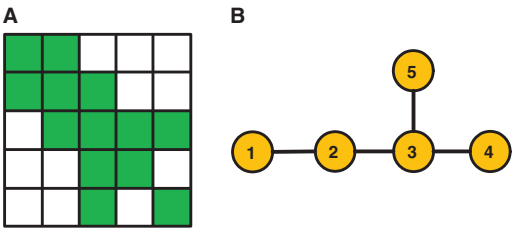

Fig. 2. Illustration of the concentration matrix (**A**) and the corresponding graphical model (**B**). The zero entries in the concentration matrix are unfilled while the non-zero entries are filled with green. In this example, x_1 and x_5 are conditionally independent given all other variables.

In Meinshausen and Bühlmann (2006), an approximate formulation is proposed to learn Gaussian graphical models by solving a series of sparse regression problems. Specifically, the conditional dependencies between x_i and all other variables are learned by solving the following 1-norm penalized regression problem known as lasso (Tibshirani, 1996):

$$\hat{\mathbf{w}}=\arg\min_{\mathbf{w}\in\mathbb{R}^{d-1}}\|\mathbf{y_i}-Y^{-i}\mathbf{w}\|^2+\lambda\|\mathbf{w}\|_1, \tag{9}$$

where $Y^{-i}=[\mathbf{y_1},\ldots,\mathbf{y_{i-1}},\mathbf{y_{i+1}},\ldots,\mathbf{y_n}]\in\mathbb{R}^{d\times(n-1)}$ is the data matrix obtained by removing the i-th data item. The conditional dependencies between x_i and all other variables are obtained from the corresponding components in the weight vector $\mathbf{w}$. Note that the regression of x_i onto x_j and that of x_j onto x_i may not give the same result. Hence, two simple schemes, based on logic operations *or* and *and*, are proposed to interpret the results. In the first scheme, two variables are considered to be conditionally dependent if *either* of them yields non-zero weight (Meinshausen and Bühlmann, 2006). In the second scheme, they are considered as conditionally dependent if *both* of them give non-zero weights. The first scheme is employed in this work (Meinshausen and Bühlmann, 2006). The pairwise relationships between all pairs of variables can be obtained by running the sparse regression problem in Equation (9) for each variable. A critical observation that leads to the efficiency of the formulation in Equation (9) is that it involves solving d independent lasso problems, one for each variable. The lasso problem can be solved very efficiently by many algorithms such as the accelerated gradient method (Liu *et al.*, 2009). It has been shown that this sparse regression formulation of Gaussian graphical modeling maximizes the pseudo likelihood (Friedman *et al.*, 2010) and is an approximation to the maximum likelihood scheme in Equation (8) (Banerjee *et al.*, 2008; Friedman *et al.*, 2008). In particular, the exact maximization of log likelihood involves solving the lasso problems iteratively as in the graphical lasso algorithm (Friedman *et al.*, 2008), and the formulation in Equation (9) can be considered as a one-step approximation to the maximum likelihood scheme. We employ this approximate formulation to learn the mouse brain networks due to its efficiency.

3.2 Robust estimation via stability selection

The regularization parameter λ in Equation (9) controls the trade-off between the sparsity of solution and data fit. Specifically, when λ is set to a very large value, most of the entries of $\mathbf{w}$ are set to zero. Hence, a challenge in practice is how to select the value for λ. Stability selection (Meinshausen and Bühlmann, 2010) addresses this problem by ideas similar to the ensemble learning methods widely used in machine learning (Bühlmann, 2004). In stability selection, we choose a set of λ values denoted by Λ, instead of a single λ value. For each $\lambda\in\Lambda$, we compute the selection probability for each variable, which is defined as the probability of each variable been selected when randomly resampling from the data. Formally, let I be a random subsample of $\mathbf{y_1},\mathbf{y_2},\ldots,\mathbf{y_n}$ of size $\lfloor n/2\rfloor$ drawn without replacement. The selection probability for variable x_i is defined as

$$\hat{\Pi}^{\lambda}_{x_i}=P\{x_i\subseteq A^{\lambda}(I)\}, \tag{10}$$

where $A^{\lambda}(I)$ denotes the set of variables that have been selected when I is used as the sample and the regularization parameter is set to λ. Note that this definition of $A^{\lambda}(I)$ is independent of the specific method used for variable selection. The probability in Equation (10) is with respect to both the random sampling and other sources of randomness such as that induced by the algorithm as we discuss below. For every variable x_i, the stability path is given by the selection probabilities $\hat{\Pi}^{\lambda}_{x_i}, \lambda \in \Lambda$. It has been shown in Meinshausen and Bühlmann (2010) that 100 random resampling is sufficient to obtain accurate estimates.

Based on the selection probabilities, stable variables can be defined. For a cutoff π_{thr} with $0 < \pi_{\text{thr}} < 1$ and a set of parameters Λ, the set of stable variables are defined as

$$\hat{S}^{\text{stable}} = \{x_i : \max_{\lambda \in \Lambda}(\hat{\Pi}^{\lambda}_{x_i}) \geq \pi_{\text{thr}}\}. \tag{11}$$

By choosing the set of stable variables under the control of the cutoff π_{thr}, we keep variables with a high selection probability and discard those with low selection probabilities. It has been show that the results of stability selection vary little for sensible choices of the cutoff π_{thr} and the parameter set Λ.

It has also been shown that performance can be further improved if additional randomness is introduced into the lasso problem in Equation (9). In particular, we can randomize the amount of regularization for each variable by solving the following problem:

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w} \in \mathbb{R}^{d-1}} \|\mathbf{y_i} - Y^{-i}\mathbf{w}\|^2 + \lambda \sum_{k \in D_d^{-i}} \frac{|w_k|}{c_k}, \tag{12}$$

where $D_d^{-i} = \{1, \ldots, i-1, i+1, \ldots, d\}$, c_i are IID random variables in $[\alpha, 1]$ and $\alpha \in (0, 1]$ is a user-specified weakness factor.

4 RESULTS AND DISCUSSION

4.1 Experimental setup

In this article, we use a set of expression volumes for 3012 genes documented in the coronal sections as in Bohland *et al.* (2010). This set of genes exhibit restricted expression patterns and thus are of high neurobiological interest. For anatomical network modeling, we only use the left hemisphere voxels, since only this part of the brain is annotated in ARA. This gives rise to a 4D tensor of size $3012 \times 67 \times 41 \times 33$ in which the first index corresponds to genes, and the other three indices represent the rostral–caudal, dorsal–ventral and left–right spatial directions, respectively. In tensor factorization, we keep the dimensionality of the last three modes while reduce the dimensionality of the first mode, since we are interested in modeling the relationships among brain voxels. For genetic network modeling, we use the full volumes, and the size of our 4D tensor is $3012 \times 67 \times 41 \times 58$. In this case, we keep the dimensionality of the first mode while reducing the dimensionality of the other three modes.

The computational experiments were performed on a cluster consisting of 256 cores and 512 GB RAM. The lasso formulation was solved using the SLEP package (Liu *et al.*, 2009). We can determine the λ value that enforces $\mathbf{w}$ to be a zero vector in Equation (9) (Liu *et al.*, 2009), and this λ value is denoted as $\lambda_{\max}$. Then we set $\Lambda = \{0.1\lambda_{\max}, 0.2\lambda_{\max}, \ldots, 0.9\lambda_{\max}\}$. The selection probabilities were estimated on 100 random resampling, and the weakness factor α was set to 0.8. The sizes of reduced data were set to retain 90 and 80% of the original information for anatomical and genetic network modeling, respectively, based on the computational resource requirements. Specifically, the size of the reduced tensor is $179 \times 67 \times 41 \times 33$ in anatomical network modeling and is $3012 \times 22 \times 13 \times 19$ in genetic network modeling.

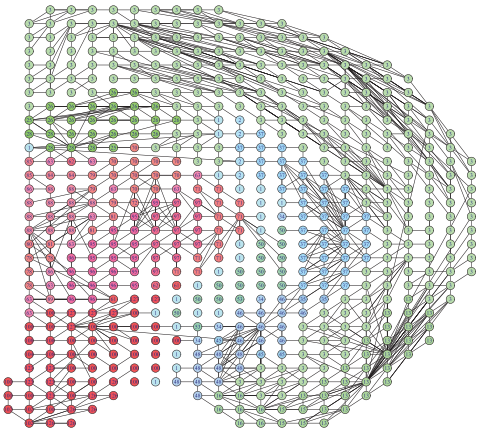

Fig. 3. Sample correlation patterns from the coronal view when the cutoff $\pi_{\text{thr}} = 0.3$. The vertices are color-coded according to the ARA annotations. Each vertex is labeled with the ARA informatics ID of the brain structure, and the corresponding structure name is given in Supplementary Table S2.

4.2 Results on anatomical network modeling

Computational modeling of the anatomical organization in the mouse brain yields a graph in 3D space in which the vertices represent brain regions, and the edges characterize the expression correlations between regions. The correlation patterns can be visualized by showing slices of the 3D brain network on 2D planes. Figure 3 shows one slice of the brain network along the coronal section. We can observe that most of the edges connect adjacent regions, showing that spatially adjacent regions tend to exhibit correlated expression patterns. Note that these correlation patterns are learned without knowing the spatial locations of voxels.

Although most of the edges connect spatially adjacent regions, there are apparent exceptions. A slice-by-slice examination of the entire anatomical networks at multiple cutoffs reveal that the voxels annotated as dentate gyrus (DG) in the ARA are highly correlated to many voxels in distant regions as shown in Figure 4. According to classical neuroanatomy, the DG plays an important role in learning and memory by processing and representing spatial information, and it has always been a topic of intense interest (Scharfman, 2007). It has been shown that the DG receives multiple sensory inputs including vestibular, olfactory, visual, auditory and somatosensory from its upstream perirhinal cortex and entorhinal cortex. It plays the role of a gate or filter, blocking or filtering excitatory activity from the inputs and controlling the amount of excitation that is propagated to the downstream hippocampus (Scharfman, 2007). A close examination of Figure 4 shows that the correlation patterns are largely consistent with those classical results. A more quantitative analysis of the results show that the correlation patterns obtained solely based on gene expressions match well with the known functions of DG. In particular, the expression patterns of the DG is highly correlated to those of the cerebral cortex and the main olfactory bulb, which provide sensory inputs to DG. In addition, DG is highly correlated to the hippocampal region and the retrohippocampal region, propagating the filtered signals to its downstream regions. We also observe that the intra-DG correlations dominate, demonstrating again that most of the edges connect spatially adjacent regions. Besides the correlations with known

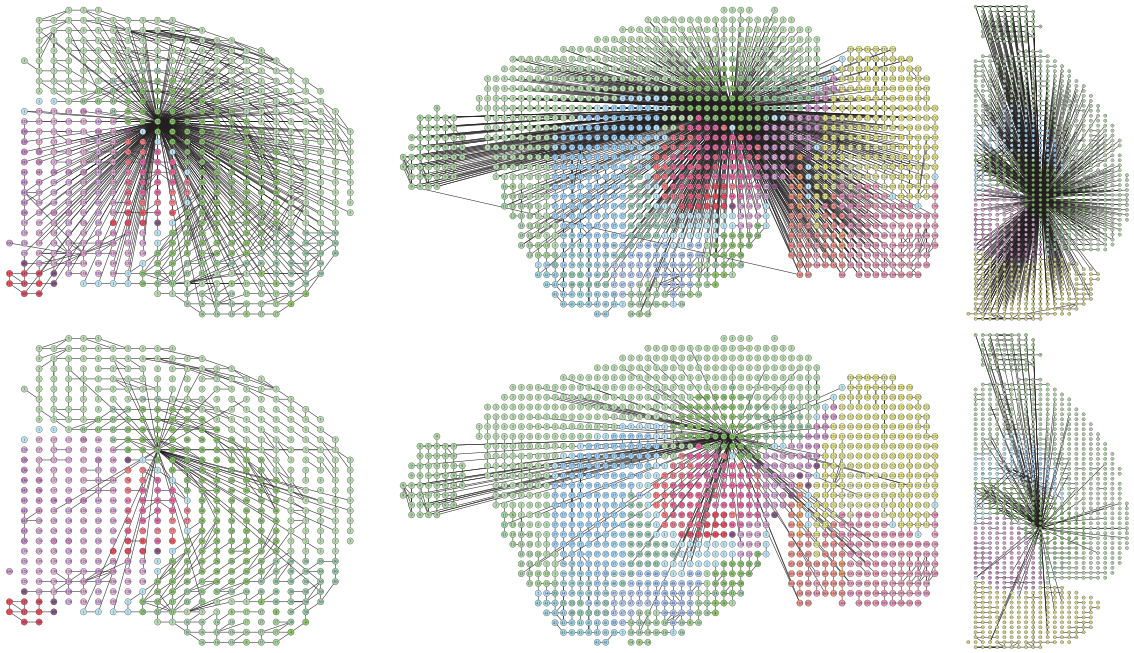

Fig. 4. Slices of the correlation patterns in the coronal (left), sagittal (middle) and horizontal (right) views when $\pi_{\text{thr}} = 0.4$ (top) and 0.5 (bottom). Each vertex is labeled with the ARA informatics ID of the brain structure, and the corresponding structure name is given in Supplementary Table S2. The region with the largest number of connections corresponds to the brain structure *dentate gyrus*.

functions, our modeling of the anatomical networks also identifies many new relationships with DG that are not known from classical anatomical studies.

Based on the obtained networks in 3D space, a variety of network analysis and visualization techniques can be employed to analyze the anatomical organization in the mouse CNS. In Bohland *et al.* (2010), the K-means algorithm is used to cluster the brain voxels into groups based on dimensionality reduced expression data, and a metric known as the S index was employed to quantitatively characterize the correspondence of the clustering results with the classical anatomy reflected in the ARA annotations. Specifically, let $R = \{r_1, \ldots, r_N\}$ be a partition of the set of brain voxels in which each r_i comprises the set of indices of the voxels that map to that cluster (or anatomical label). The spatial overlap between a region from the ARA and the clustering result is defined as: $P_{ij} = |r_i \cap r_j|/|r_j|$. From the P_{ij} values that are computed over all pairs of ARA regions and cluster result, we can then derive a global scalar index of similarity between the two partitions. Since $P_{ij} \neq P_{ji}$, X_{ij} is defined as $X_{ij} = \max\{P_{ij}, P_{ji}\}$ along with $W_{ij} = U_{ij}/\sum U_{ij}$, where $U_{ij} = \min\{|r_i|, |r_j|\}$ if $X_{ij} > 0$ and 0 otherwise. Finally, the S index is defined as $S = 1 - 4\sum_{ij} W_{ij} X_{ij}(1 - X_{ij})$.

To compare our network modeling method with the K-means clustering, we apply the leading eigenvector community detection algorithm proposed by Newman (2006) and treat each detected community as a cluster. Since different cutoff values π_{thr} in the stability selection yield different graphs, we vary π_{thr} from 0.5 to 0.85 and detect communities from each of the resulting graphs. We then run K-means with K equal to the number of communities so that the results are comparable. Since the results of K-means depend on the initialization, we run this algorithm 10 times and choose the one with the best result. We compute the S index for each case and report the results in Figure 5. We can observe that the community detection results consistently give higher S index values, indicating that the structures of our anatomical networks are in higher accordance with the classical anatomy. We also plot the number of detected communities as the cutoff changes in Figure 5. We can see that the number of communities lies approximately between 100 and 250, which is largely in correspondence with the number of structures in classical anatomy. Detailed results on community identification are provided in the Supplementary Material.

The classical anatomy was created mainly based on brain functions. Since functions are mainly determined by gene expression, the expression patterns within anatomical structures should be more correlated than those across structures. To validate this hypothesis, we show the distribution of the edges within and across the anatomical structures when $\pi_{\text{thr}} = 0.5$ in Figure 6. We also show the number of edges within and across structures when the cutoff varies from 0.2 to 0.9. We can observe that the edges within structures dominate in all cases, indicating that the expression patterns within classical anatomy are highly correlated. We can also observe from Figure 6 that the proportion of edges within anatomical structures increases as the cutoff increases. This indicates that most of the cross-structure edges have relatively small selection probabilities, and they are removed as the cutoff increases. The ranked lists of regions in terms of the number of connections are provided in the Supplementary Material.

4.3 Results on genetic network modeling

Modeling of the gene interactions using the techniques described in Section 3 yields a network consisting of 3012 vertices in which vertices represent genes, and edges characterize the correlations

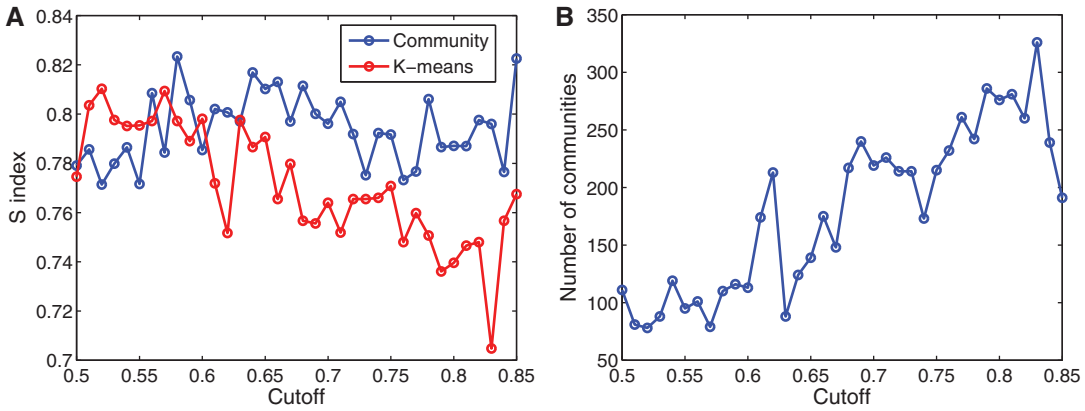

Fig. 5. Comparison of the communities detected in the anatomical networks and the K-means clustering results. (**A**) Shows the S index comparison between the anatomical structures in ARA and the results of community detection and K-means. (**B**) Shows the number of communities as the cutoff changes.

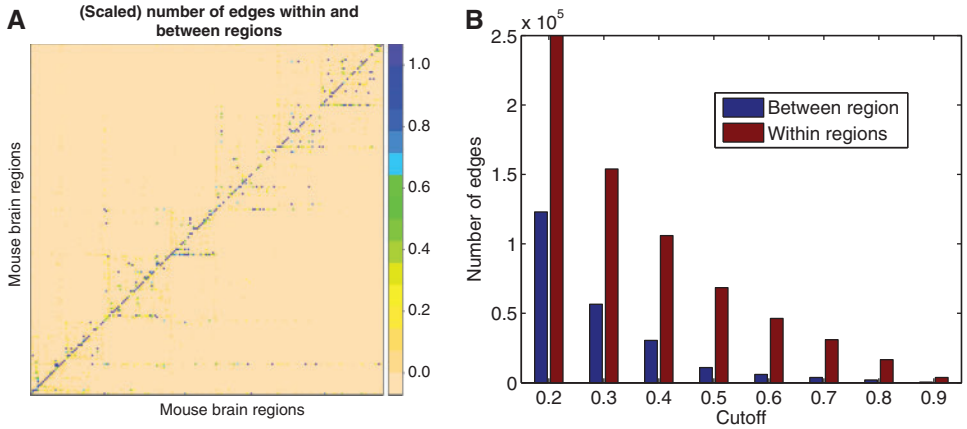

Fig. 6. Visualization of the edge distribution within and between different brain regions. (**A**) Shows the distribution of the edges within and across the anatomical structures when $\pi_{\text{thr}} = 0.5$. The rows and columns correspond to the structures annotated in the ARA. This matrix is normalized to the interval $[0, 1]$ row by row, and hence it is not symmetric. Each row indicates the proportion of a particular structure's edges that connect to other structures. In particular, each entry (i, j) in the matrix represents the proportion of structure i's edges that connect to structure j. (**B**) Shows the number of edges within and between anatomical structures in ARA as the cutoff changes.

between genes. Since genes involved in the same pathway usually exhibit similar expression patterns, correlated expression patterns may imply similar biological functions. We hence use Gene Ontology (GO) (Ashburner *et al.*, 2000) to evaluate the functional relationships among tightly connected genes in the network. In particular, we consider a gene and its direct neighbors as a group (Gustafsson *et al.*, 2005) and evaluate the functional enrichment of each group using the hypergeometric distribution (Boyle *et al.*, 2004). We apply Bonferroni correction for multiple hypothesis testing and consider GO terms with corrected $P < 0.05$ as statistically significant (Boyle *et al.*, 2004). We vary the cutoff π_{thr} and observe that most of the groups are annotated with at least one statistically significant GO term. In particular, when $\pi_{\text{thr}} = 0.5$, there are 2702 groups annotated with at least one statistically significant GO term, and the average number of terms per group is 15. This indicates that most of the groups are associated with multiple enriched terms.

It has been previously observed that the degrees of many biological networks follows a power-law distribution (Barabási and Oltvai, 2004). This indicates that there exists a small number of highly connected genes known as hubs. We vary the cutoff and observe that the set of highly connected genes are largely consistent (details provided in the Supplementary Material). We report the top 10 genes with the largest number of connections in Table 1 when $\pi_{\text{thr}} = 0.8$ and show slices of their expression patterns in the Supplementary Material. We can observe that all these groups are highly enriched with the biological function *binding* or *protein binding*, implicating that they are likely to encode transcription factors. Among these 10 genes, the APP encodes an integral membrane protein expressed in many tissues and concentrated in the synapses of neurons. Homologous proteins have been identified in other organisms such as *Drosophila*, *C.elegans* and all mammals. APP is best known for its association with the Alzheimer's disease, and mutations in critical regions of APP cause familial susceptibility to Alzheimer's disease. It would be interesting to investigate how the 'hubness' of APP is related to CNS disease.

Table 1. Top 10 genes with the largest number of connections when $\pi_{thr} = 0.8$

Gene	No. of neighbors	GO molecular function	Corrected *P*-value
App	274	Binding	8.00e-31
Nsf	231	Binding	2.50e-21
Acsl5	219	Binding	6.00e-20
Nrgn	155	Protein binding	5.66e-24
Acadvl	95	Binding	1.68e-05
Syt1	94	Protein binding	4.36e-14
Chn2	82	Protein binding	1.15e-07
Btg1	69	Binding	5.74e-08
Eef1a1	59	Binding	1.79e-06
Apoe	54	Binding	6.76e-06

The molecular function and corrected *P*-values are also shown.

5 CONCLUSIONS

We model the anatomical and genetic organizations in the mouse brain as networks. To enable robust and efficient network construction, we employ tensor factorization techniques to reduce the data volumes. The resulting networks recover known relations and predict novel correlations not known from the literature. The employed network modeling formulation is an approximate scheme. It would be interesting to compare this approximate formulation with the exact one on small datasets, where exact optimization can be applied. The proposed methods can be applied to model other biological systems, such as the *Drosophila* transcriptional networks. We will explore the network modeling of other biological systems in the future.

Funding: This work was supported by Old Dominion University.

Conflict of Interest: none declared.

REFERENCES

Ashburner,M. *et al.* (2000) Gene Ontology: tool for the unification of biology. *Nat. Genet.*, **25**, 25–29.

Banerjee,O. *et al.* (2008) Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *J. Mach. Learn. Res.*, **9**, 485–516.

Barabási,A.-L. and Oltvai,Z.N. (2004) Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.*, **5**, 101–113.

Bohland,J.W. *et al.* (2010) Clustering of spatial gene expression patterns in the mouse brain and comparison with classical neuroanatomy. *Methods*, **50**, 105–112.

Boyle,E.I. *et al.* (2004) GO::TermFinder–open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics*, **20**, 3710–3715.

Bühlmann,P. (2004) Bagging, boosting and ensemble methods. In Gentle,J. *et al.* (eds) *Handbook of Computational Statistics: Concepts and Methods*. Springer, Berlin, Heidelberg, Germany, pp. 877–907.

Dempster,A.P. (1972) Covariance selection. *Biometrics*, **28**, 157–175.

Dong,H.W. (2009) *The Allen Reference Atlas: A Digital Color Brain Atlas of the C57BL/6J Male Mouse*. John Wiley & Sons, Inc., Hoboken, New Jersey.

Edwards,D. (2000) *Introduction to Graphical Modelling*, 2nd edn. Springer, New York, Inc.

Friedman,J. *et al.* (2008) Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, **9**, 432–441.

Friedman,J. *et al.* (2010) Applications of the lasso and grouped lasso to the estimation of sparse graphical models. *Technical Report*. Department of Statistics, Stanford University, Stanford, CA.

Gustafsson,M. *et al.* (2005) Constructing and analyzing a large-scale gene-to-gene regulatory network-lasso-constrained inference and biological validation. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **2**, 254–261.

Jones,A.R. *et al.* (2009) The Allen Brain Atlas: 5 years and beyond. *Nat. Rev. Neurosci.*, **10**, 821–828.

Kolda,T.G. and Bader,B.W. (2009) Tensor decompositions and applications. *SIAM Rev.*, **51**, 455–500.

Lathauwer,L.D. *et al.* (2000a) A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, **21**, 1253–1278.

Lathauwer,L.D. *et al.* (2000b) On the best rank-1 and rank-(R_1,R_2,$\cdots$,R_N) approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.*, **21**, 1324–1342.

Lein,E.S. (2007) Genome-wide atlas of gene expression in the adult mouse brain. *Nature*, **445**, 168–176.

Liu,J. *et al.* (2009) *SLEP: Sparse Learning with Efficient Projections*. Arizona State University, Tempe, Arizona, USA.

Meinshausen,N. and Bühlmann,P. (2006) High-dimensional graphs and variable selection with the lasso. *Ann. Stat.*, **34**, 1436–1462.

Meinshausen,N. and Bühlmann,P. (2010) Stability selection. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **72**, 417–473.

Newman,M.E.J. (2006) Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, **74**, 036104.

Ng,L. *et al.* (2007) Neuroinformatics for genome-wide 3-D gene expression mapping in the mouse brain. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **4**, 382–393.

Omberg,L. *et al.* (2007) A tensor higher-order singular value decomposition for integrative analysis of DNA microarray data from different studies. *Proc. Natl Acad. Sci. USA*, **104**, 18371–18376.

Scharfman,H.E. (2007) *The Dentate Gyrus: A Comprehensive Guide to Structure, Function, and Clinical Implications*. Elsevier Science, Amsterdam, The Netherlands.

Tibshirani,R. (1996) Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B*, **58**, 267–288.

Vasilescu,M.A.O. and Terzopoulos,D. (2004) TensorTextures: multilinear image-based rendering. *ACM Trans. Graph.*, **23**, 336–342.

Wang,H. *et al.* (2005) Out-of-core tensor approximation of multi-dimensional matrices of visual data. *ACM Trans. Graph.*, **24**, 527–535.

Whittaker,J. (1990) *Graphical Models in Applied Multivariate Statistics*. Wiley, New York, NY.

Wille,A. *et al.* (2004) Sparse graphical Gaussian modeling of the isoprenoid gene network in *Arabidopsis thaliana*. *Genome Biol.*, **5**, R92.

Wrede,R.C. (1972) *Introduction to Vector and Tensor Analysis*. Dover Publications, Mineola, NY.

Ye,J. (2005) Generalized low rank approximations of matrices. *Mach. Learn.*, **61**, 167–191.

Yuan,M. and Lin,Y. (2007) Model selection and estimation in the Gaussian graphical model. *Biometrika*, **94**, 19–35.