Summer 2016

A Computational Framework for Learning from Complex Data: Formulations, Algorithms, and Applications

Wenlu Zhang
Old Dominion University

A COMPUTATIONAL FRAMEWORK FOR LEARNING FROM COMPLEX DATA: FORMULATIONS, ALGORITHMS, AND APPLICATIONS

by

Wenlu Zhang
M.S. June 2010, City College of New York

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
August 2016

Approved by:

Shuiwang Ji (Director)

Andrey Chernikov (Member)

Nikos Chrisochoides (Member)

Christopher Osgood (Member)

ABSTRACT

A COMPUTATIONAL FRAMEWORK FOR LEARNING FROM COMPLEX DATA: FORMULATIONS, ALGORITHMS, AND APPLICATIONS

Wenlu Zhang
Old Dominion University, 2016
Director: Dr. Shuiwang Ji

Many real-world processes are dynamically changing over time. As a consequence, the observed complex data generated by these processes also evolve smoothly. For example, in computational biology, the expression data matrices are evolving, since gene expression controls are deployed sequentially during development in many biological processes. Investigations into the spatial and temporal gene expression dynamics are essential for understanding the regulatory biology governing development. In this dissertation, I mainly focus on two types of complex data: genome-wide spatial gene expression patterns in the model organism fruit fly *Drosophila melanogaster* and Allen Brain Atlas mouse brain data. I provide a framework to explore spatiotemporal regulation of gene expression during development. I develop evolutionary co-clustering formulation to identify co-expressed domains and the associated genes simultaneously over different temporal stages using a mesh-generation pipeline. I also propose to employ the deep convolutional neural networks as a multi-layer feature extractor to generate generic representations for gene expression pattern *in situ* hybridization (ISH) images. Furthermore, I employ the multi-task learning method to fine-tune the pre-trained models with labeled ISH images. My proposed computational methods are evaluated using synthetic data sets and real biological data sets including the

gene expression data from the fruit fly BDGP data sets and Allen Developing Mouse Brain Atlas in comparison with baseline existing methods. Experimental results indicate that the proposed representations, formulations, and methods are efficient and effective in annotating and analyzing the large-scale biological data sets.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Prof. Shuiwang Ji, for his guidance, encouragement, and support during my dissertation research. He is an incredible advisor, an enthusiastic researcher, and an easygoing friend. The experience with him are my lifelong assets. I would like to thank my defense committee members, Prof. Andrey Chernikov, Prof. Nikos Chrischoides and Prof. Christopher Osgood, for their valuable interactions and feedbacks.

I have been very fortunate to work on the fruit fly and mouse brain projects, which involve a group of incredible people to whom I deeply indebted. I would like to thank my collaborators, Rongjian Li, Tao Zeng and Ahmed Fakhry for their insightful discussions and valuable interactions.

Last but not least, I want to thank my family for their support and understanding.

My dissertation work is supported in part by the National Science Foundation grant DBI-1147134, DBI-1350258.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The complex data generated by many real-world processes are dynamically changing over time. For example, in literature mining, the author-conference co-occurrence matrix evolves dynamically over time, since authors may shift their research interests smoothly. Temporal data mining aims at discovering knowledge from time-varying data and is now receiving increasing attention in many domains, including graph and network analysis [1–3], information retrieval [4, 5], text mining [6], clustering analysis [7–10], and matrix factorization [11]. Since the complex data are evolving smoothly over time, the patterns embedded into the data are also expected to change smoothly. Therefore, one of the key challenges in temporal data mining is how to incorporate temporal smoothness into the patterns identified from adjacent time points.

In this dissertation, I focus on a fundamental challenge in biological complex data, which is to elucidate the gene expression controls that generate the complex body plans during development. Currently, gene expression controls are deployed sequentially in many biological processes. This generates the expression data matrices that are evolving over time. Advances in sequencing and gene-prediction technologies have led to the discovery of virtually complete sets of protein-coding sequences in many model systems. In contrast, how these coding sequences are controlled

by the regulatory sequences to transform a single cell, through cell division and differentiation, into a complex multicellular organism remains largely unknown. In multicellular organisms, one of the primary purposes of gene control is execution of the genomic regulatory code to generate complex body plans during development [12, 13]. This process critically depends on the right gene being activated in the right cell (spatially) at the right time (temporally). Thus, analysis of spatiotemporal gene expression patterns provides a promising way for investigating the gene regulatory networks governing development. Recently, genome-wide spatial gene expression patterns in the model organism fruit fly *Drosophila melanogaster* have been generated using high-throughput RNA *in situ* hybridization [14,15]. These data provide useful information to study the temporal and spatial gene expression patterns and the underlying developmental regulatory networks [16–19].

In this dissertation, I use the *Drosophila* ISH gene expression pattern images provided by the FlyExpress database [20,21], which contains genome-wide, standardized images from multiple sources, including the Berkeley *Drosophila* Genome Project (BDGP). For each *Drosophila* embryo, a set of high-resolution, two-dimensional image series was taken from different views (lateral, dorsal, and lateral-dorsal and other intermediate views). These images were then subsequently standardized semi-manually.

In this dissertation, I focus on the lateral-view images only, since most of images in FlyExpress are in lateral view. In the FlyExpress database, the embryogenesis of *Drosophila* has been divided into six discrete stage ranges (stages 1-3, 4-6, 7-8, 9-10,

11-12, and 13-17). I use those images in the later 5 stage ranges in the controlled vocabulary (CV) term annotation, since only a very small number of keywords are used in the first stage range. This wealth of data creates opportunities for studying the developmental regulatory networks. However, the sheer volume and complexity of these data preclude the traditional practice of manual analysis and make automated methods essential [17–20, 22–26].

The mammalian brain controls cognition, emotion, and perception and is one of the most complex yet least understood biological systems [27]. It is known that there are at least several hundreds of distinct types of cells in the mammalian brain. These cell types are arranged into complex circuits, which ultimately are responsible for generating brain function. The phenotypic properties of cells of different types are largely the consequences of unique combinations of expressed gene products; therefore, analysis of gene expression patterns provides an informative modality to study developmental gene regulation and cellular diversity. To date, the Allen Brain Atlas (ABA) [28] contains one of the most comprehensive collections of genome-scale, cellular-resolution, three-dimensional (3D) gene expression patterns in the brain of a mouse, a core model for mammalian brain development and behavioral genetics. Analysis of this data set would shed light on the anatomic and genetic organizations of the mammalian brain. Currently, the Allen Brain Atlas provides gene expression data for the developing and adult mouse and human brains [28–30]. Building upon the foundation established by the Allen Adult Mouse Brain Atlas [28], the Allen Developing Mouse Brain Atlas provides spatiotemporal *in situ* hybridization (ISH) gene

expression data across multiple stages of mouse brain development [30], yielding effectively a four-dimensional brain atlas. It provides a framework to explore temporal and spatial regulation of gene expression during development. To establish a common coordinate framework for analyzing the ISH data, the ISH image series are aligned to the Allen Developing Mouse Brain Reference Atlas (the Reference Atlas). The Reference Atlas was created based on the "prosomeric model" [31], which proposes that the neural tube is divided into a grid-like pattern of longitudinal and transverse regions. These divisions form the primary histogenetic domains upon which further elaboration of expression are developed independently [32]. Therefore it is fundamentally important to study the gene regulations that lead to the formation of these domains across multiple stages of mouse brain development.

1.1 CONTRIBUTIONS OF THIS DISSERTATION

1. I demonstrate the mining of the hidden block structures from data matrices that evolve dynamically over time. I develop a probabilistic model for evolutionary co-clustering complex data. The proposed probabilistic model assumes that the observed data matrices are generated via a two-step process that depends on the historic co-clusters, thereby capturing the temporal smoothness in a probabilistically principled manner. To enable maximum likelihood parameter estimation, I develop an EM algorithm for probabilistic model. An appealing feature of the proposed probabilistic model is that it leads to soft co-clustering assignments naturally.

2. I perform a systematic application study on the analysis of *Drosophila* gene expression pattern images. In this application, I use a geometric domain tessellation

pipeline to convert gene expression pattern images to an algebraic representation, which is a data matrix for each of the developmental time points. I then apply my evolutionary co-clustering algorithm to cluster the genes and the mesh elements simultaneously across multiple time points.

3. I employ a co-clustering model to cluster the genes and the voxel simultaneously, thereby elucidating the genetic and anatomic interactions governing mouse brain development. I represent the data set as a bipartite graph and propose to approximate the bipartite graph using a tripartite graph, leading to a graph approximation formulation for co-clustering. I show that this formulation can be mathematically expressed in the framework of probabilistic latent semantic analysis (PLSA). I give probabilistic random walk interpretation of PLSA in the context of co-clustering. This allows me to use the expectation maximization algorithm for PLSA to estimate the co-clustering parameters.

4. I explore whether the transfer learning property of convolutional neural networks (CNNs) can be generalized to compute features for biological images. I propose to transfer knowledge from natural images by training CNNs on the ImageNet data set. To take this idea one step further, I propose to fine-tune the trained model with labeled ISH images, and resume training from already learned weights using multi-task learning schemes. The two models are then both used as feature extractors to compute image features from *Drosophila* gene expression pattern images. The resulting features are subsequently used to train and validate my machine learning method for annotating gene expression patterns.

1.2 SUMMARY OF REMAINING CHAPTERS

Chapter 2: Evolutionary Soft Co-clustering. In this chapter, I consider the mining of hidden block structures from time-varying data using evolutionary co-clustering. Existing methods are based on the spectral learning framework, thus lacking a probabilistic interpretation. To overcome this limitation, I develop a probabilistic model for evolutionary co-clustering. The proposed model assumes that the observed data are generated via a two-step process that depends on the historic co-clusters, thereby capturing the temporal smoothness in a probabilistically principled manner. I develop an EM algorithm to perform maximum likelihood parameter estimation. An appealing feature of the proposed probabilistic model is that it leads to soft co-clustering assignments naturally. I evaluate the proposed method on both synthetic and real data sets. Experimental results show that my method consistently outperforms prior approaches based on spectral method.

Chapter 3: *Drosophila* Gene Expression Pattern Image Analysis. In this chapter, I develop a set of computational methods and open source tools for identifying co-expressed embryonic domains and the associated genes simultaneously across multiple developmental stages. To map the expression patterns of many genes into the same coordinate space and account for the embryonic shape variations, I develop a mesh generation method to deform a meshed generic ellipse to each individual embryo. I then apply my evolutionary co-clustering formulation to cluster the genes and the mesh elements, thereby identifying co-expressed embryonic domains and the associated genes simultaneously. Experimental results indicate that the gene and

mesh co-clusters can be correlated to key developmental events during the stages of embryogenesis I study. The open source software tool has been made available at https://github.com/DIVE-WSU/MeshClustering.

Chapter 4: A Probabilistic Latent Semantic Analysis Model for Co-Clustering the Mouse Brain Atlas. In this chapter, I employ a graph approximation formulation to co-cluster the genes and the brain voxels simultaneously for each time point. I show that this formulation can be expressed as a probabilistic latent semantic analysis (PLSA) model, thereby allowing me to use the expectation- maximization algorithm for PLSA to estimate the co-clustering parameters. To provide a quantitative comparison with prior methods, I evaluate the co-clustering method on a set of standard synthetic data sets. Results indicate that my method consistently outperforms prior methods. I apply my method to co-cluster the Allen Developing Mouse Brain Atlas data. Results indicate that my clustering of voxels is more consistent with classical neuroanatomy than those of prior methods. My analysis also yields sets of genes that are co-expressed in a subset of the brain voxels.

Chapter 5: Deep Model Based Transfer and Multi-Task Learning for Biological Image Analysis. In this chapter, I develop problem-independent feature extraction methods to generate hierarchical representations for ISH images. My approach is based on the deep CNNs that can act on image pixels directly. To make the extracted features generic, the models are trained using a natural image set with millions of labeled examples. These models are transferred to the ISH image domain and used directly as feature extractors to compute image representations. Furthermore,

I employ the multi-task learning method to fine-tune the pre-trained models with labeled ISH images, and also extract features from the fine-tuned models. Experimental results show that feature representations computed by deep models based on transfer and multi-task learning significantly outperform other methods for annotating gene expression patterns at different stage ranges. I also demonstrate that the intermediate layers of deep models produce the best gene expression pattern representations.

Chapter 6: Conclusion and Outlook. In this chapter, I provide a summary of my contributions and discuss future research directions.

1.3 NOTATIONS

I use $\mathrm{Tr}(W)$ to represent the trace of matrix W where $\mathrm{Tr}(W) = \sum_{i=1}^{n} w_{ii}$ for any matrix $W \in \mathbb{R}^{n\times n}$. The squared Frobenius norm of a matrix W is defined as $\|W\|_F^2 = \sum_{i,j} w_{i,j}^2 = \mathrm{Tr}(W^T W)$. I use $A \in \mathbb{R}^{m\times n}$ to denote the data matrix for a problem with k co-clusters, the co-clustering results can be encoded into a co-cluster indicator matrix $R \in \mathbb{R}^{(m+n)\times k}$. Let $R^T = [R_1^T, R_2^T]$, where $R_1 \in \mathbb{R}^{m\times k}$ and $R_2 \in \mathbb{R}^{n\times k}$. The indicator matrix R is defined as follows: $(R_1)_{ij} = 1$ if the ith row belongs to the jth co-cluster, and zero otherwise; $(R_2)_{ij} = 1$ if the ith column belongs to the jth co-cluster, and zero otherwise. I further define $\tilde{R} \in \mathbb{R}^{(m+n)\times k}$, where each column of $\tilde{R}$ is the corresponding column in R divided by the square root of the number of ones in that column.

CHAPTER 2

EVOLUTIONARY SOFT CO-CLUSTERING

I consider the mining of hidden block structures from time-varying data using evolutionary co-clustering. Existing methods are based on the spectral learning framework, thus lacking a probabilistic interpretation. To overcome this limitation, I develop a probabilistic model for evolutionary co-clustering in this paper. The proposed model assumes that the observed data are generated via a two-step process that depends on the historic co-clusters, thereby capturing the temporal smoothness in a probabilistically principled manner. I develop an EM algorithm to perform maximum likelihood parameter estimation. An appealing feature of the proposed probabilistic model is that it leads to soft co-clustering assignments naturally. To the best of my knowledge, my work represents the first attempt to perform evolutionary soft co-clustering. My evaluate the proposed method on both synthetic and real data sets. Experimental results show that my method consistently outperforms prior approaches based on spectral method.

2.1 BACKGROUND

Cluster analysis aims at grouping a set of data points into clusters so that the data points in the same cluster are similar, while those in different clusters are dissimilar. Given a data matrix $A = [a_1, a_2, \cdots, a_n] \in \mathbb{R}^{m \times n}$ consisting of n data points $\{a_i\}_{i=1}^{n} \in \mathbb{R}^m$. Let $\Pi = \{\pi_j\}_{j=1}^{k}$ denote a partition of the data into k clusters; that is, $\pi_j =$

$\{v|a_v \text{ in cluster } j\}$ and $\pi_i \bigcap \pi_j = \emptyset$ for $i \neq j$. The partition can also be encoded equivalently into an $n \times k$ cluster indicator matrix $Y = [y_1, y_2, \cdots, y_k]$, where $Y_{pq} = 1$ if the pth data point belongs to the qth cluster, and 0 otherwise. I further define a normalized cluster indicator matrix $\tilde{Y} = [\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_k]$, where $\tilde{y}_i = y_i/\sqrt{|\pi_i|}$ and $|\pi_i|$ denotes the number of data points in the ith cluster. It can be verified that the columns of $\tilde{y}$ are orthonormal, i.e., $\tilde{y}^T\tilde{y} = I_k$.

2.1.1 SPECTRAL CLUSTERING

In spectral clustering [33–36], the data set is represented by a weighted graph $G = (V, E)$ in which the vertices in V correspond to data points, and the edges in E characterize the similarities between data points. The weights of the edges are usually encoded into the adjacency matrix W. Several constructions of similarity graph are regularly used, such as the ϵ-neighborhood graph and the k-nearest neighbor graph [34].

Spectral clustering is based on the idea of graph cuts, and different graph cut measures have been defined. Two popular approaches are to maximize the average association and to minimize the normalized cut [33]. For two subsets, π_p, $\pi_q \in \Pi$, the cut between π_p and π_q is defined as $cut(\pi_p, \pi_q) = \sum_{i\in\pi_p, j\in\pi_q} W(i, j)$. Then the k-way average association (AA) and the k-way normalized cut (NC) can be written as

$$\text{AA} = \sum_{l=1}^{k} \frac{cut(\pi_l, \pi_l)}{|\pi_l|}, \quad \text{NC} = \sum_{l=1}^{k} \frac{cut(\pi_l, \Pi \setminus \pi_l)}{cut(\pi_l, \Pi)}, \tag{1}$$

where $\setminus$ denotes the set minus operation. In [9], the negated average association

is defined as $\mathrm{NA} = \mathrm{Tr}(W) - AA$. Note that the average association characterizes the within cluster association, while the normalized cut captures the between cluster separation. Furthermore, maximizing the average association is equivalent to minimizing the negated average association. Hence, the negated average association will be used throughout this dissertation.

It has been shown [33] that exact minimization of common graph cut measures, such as the normalized cut and the negated average association, is intractable. Hence, a two-step procedure is commonly employed in spectral clustering. In the first step, the graph cut problems are relaxed to a trace optimization problem, whose solution typically can be obtained by computing the eigen-decomposition of the graph Laplacian matrices [34, 37]. Then in the second step, the final clustering results are generated by clustering the solution of the relaxed problem.

Note that I focus on how to incorporate smoothness constraints into the first step in this dissertation, so the second step will not be discussed further in the rest of this dissertation.

2.1.2 SPECTRAL CO-CLUSTERING

In [38, 39], the spectral clustering formalism is extended to solve co-clustering problems. Given a data matrix $A \in \mathbb{R}^{m \times n}$, such as the word-by-document matrix, a bipartite graph is constructed, where the two sets of vertices correspond to the rows and the columns, respectively.

Then the co-clustering problem is reduced to perform graph cuts on this bipartite

graph. Formally, the similarity matrix of the bipartite graph can be written as

$$W = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}. \tag{2}$$

A variety of graph cut criteria can then be applied to partition the bipartite graph. For example, when the normalized cut is used, the Laplacian matrix and the degree matrix for this bipartite graph can be written as

$$L = \begin{bmatrix} D_1 & -A \\ -A^T & D_2 \end{bmatrix}, \quad D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}, \tag{3}$$

where D_1 and D_2 are diagonal matrices whose diagonal elements are defined as

$$D_1(ii) = \sum_j A_{ij}, \;\; D_2(jj) = \sum_i A_{ij}.$$

Then the normalized cut criterion can be relaxed, and the solution for the relaxed problem can be obtained by solving the following eigenvalue problem:

$$\begin{bmatrix} D_1 & -A \\ -A^T & D_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \tag{4}$$

where $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$ are the relaxed row and column cluster indicator matrices, respectively.

2.1.3 EVOLUTIONARY CLUSTERING

When the data matrices evolve along the temporal dimension, it is desirable to capture the temporal smoothness in clustering analysis. Recently, several evolutionary clustering methods have been developed to cluster time-varying data by incorporating temporal smoothness constraints directly into the clustering framework [8–10].

In [9], two main frameworks, known as preserving cluster quality (PCQ) and p-reserving cluster membership (PCM), are proposed to incorporate temporal smoothness. In these two formulations, the cost functions contain two terms, known as the snapshot cost (CS) and the temporal cost (CT) as $\text{Cost} = \alpha \cdot \text{CS} + (1-\alpha)\text{CT}$, where $0 \leq \alpha \leq 1$ is a tunable parameter. In this formulation, the snapshot cost captures the clustering quality on the current data matrix, while the temporal cost encourages the temporal smoothness with respect to either historic data or historic clustering results. The main difference between PCQ and PCM lies in the definitions of the temporal costs. Specifically, the temporal cost in PCQ is devised to encode the consistency between current clustering results with historic data, while that in PCM is used to encourage temporal smoothness between current and historic clustering results.

Let Y_t denote the cluster indicator matrix for time t, then the objective function for PCQ can be expressed as $\text{Cost}_{\text{PCQ}} = \alpha \cdot \text{Cost}_t|_{Y_t} + (1-\alpha) \cdot \text{Cost}_{t-1}|_{Y_t}$, where $\text{Cost}_t|_{Y_t}$ and $\text{Cost}_{t-1}|_{Y_t}$ denote the costs of applying the clustering results in Y_t to the data at time points t and $t-1$, respectively. In contrast, the temporal cost in PCM is expressed as the difference between the current and the historic clustering results, leading to the following overall objective function $\text{Cost}_{\text{PCM}} = \alpha \cdot \text{Cost}_t|_{Y_t} + (1-\alpha) \cdot \text{dist}(Y_t, Y_{t-1})$, where $\text{dist}(\cdot,\cdot)$ denotes certain distance measure.

Following the soft clustering framework proposed in [40], an evolutionary clustering method based on nonnegative matrix factorization (NMF) has been developed in [10]. Let W_t be the similarity matrix for time point t, the objective function for

evolutionary clustering in [10] can be expressed as

$$\text{Cost}_{\text{NMF}} = \alpha \cdot D(W_t \| X_t \Lambda_t X_t^T) + (1 - \alpha) \cdot D(X_{t-1} \Lambda_{t-1} \| X_t \Lambda_t),$$

where $D(\cdot\|\cdot)$ is the KL-divergence, X_t is the soft clustering indicator matrix, and Λ_t is a diagonal matrix. An iterative procedure is devised to compute the solution. It is also shown in [10] that the proposed method can be interpreted from the perspective of probabilistic generative models.

2.2 EVOLUTIONARY SOFT CO-CLUSTERING

Although both co-clustering and evolutionary clustering have been intensively studied, the field of evolutionary co-clustering remains largely unexplored [41]. In addition, prior method (discussed in Section 2.3) employs singular value decomposition (SVD) in computing the solutions of relaxed problems. In many applications, such as image and text analysis, the original data matrices are nonnegative. A factorization such as SVD produces factors containing negative entries. This leads to complex cancelations between positive and negative numbers, and the results are usually difficult to interpret [42]. To address this challenge, I propose a probabilistic model for evolutionary co-clustering in this section. This model results in nonnegative factors, thereby overcoming the limitation of spectral methods. In addition, the probabilities can be interpreted to produce soft co-clusters.

2.2.1 THE PROPOSED MODEL

In the proposed model, I assume that the similarity matrix W_t of the bipartite

graph can be factorized as

$$W_t = H_t \tilde{H}_t, \tag{5}$$

where

$$W_t = \begin{bmatrix} 0 & A_t \\ A_t{}^T & 0 \end{bmatrix}, \tag{6}$$

$A_t \in \mathbb{R}^{m \times n}$ is the data matrix,

$$H_t = \begin{bmatrix} H_{1,t} & 0 \\ 0 & H_{2,t} \end{bmatrix}, \quad \tilde{H}_t = \begin{bmatrix} 0 & H_{2,t}^T \\ H_{1,t}^T & 0 \end{bmatrix}, \tag{7}$$

where $H_t \in \mathbb{R}^{(m+n)\times(2k)}$, $\tilde{H}_t \in \mathbb{R}^{(2k)\times(m+n)}$, $H_{1,t} \in \mathbb{R}^{m\times k}$ denotes the row cluster indicator matrix, and $H_{2,t} \in \mathbb{R}^{n\times k}$ denotes the column cluster indicator matrix. It follows that

$$H_t \tilde{H}_t = \begin{bmatrix} 0 & H_{1,t} H_{2,t}^T \\ \left(H_{1,t} H_{2,t}^T\right)^T & 0 \end{bmatrix}, \tag{8}$$

which matches the structure of W_t in Eq. (6).

In the proposed probabilistic model, the similarity matrix W_t is generated via a two-step process. In the first step, $H_t \tilde{H}_t$ is generated based on the co-clustering results $H_{t-1}\tilde{H}_{t-1}$ at time point $t-1$ using $P(H_t\tilde{H}_t | H_{t-1}\tilde{H}_{t-1})$. In the second step, the observed similarity matrix W_t is generated based on $H_t\tilde{H}_t$ using $P(W_t|H_t\tilde{H}_t)$. Following [10], I employ the Dirichlet and multinomial distributions in the first and second steps, respectively. This gives rise to the following log likelihood function of observing the current weight matrix W_t:

$$\begin{aligned} L &= \log P(W_t|H_t\tilde{H}_t) + \nu \log P(H_t\tilde{H}_t|H_{t-1}\tilde{H}_{t-1}) \\ &= 2\sum_{ij} (A_t)_{ij} \log (H_{1,t}H_{2,t}^T)_{ij} + 2\nu \sum_{ij} (H_{1,t-1}H_{2,t-1}^T)_{ij} \log (H_{1,t}H_{2,t}^T)_{ij}, \end{aligned}$$

where parameter ν controls the temporal smoothness.

2.2.2 AN EM ALGORITHM

To maximize the log likelihood in Eq. (9), I derive an EM algorithm in the following. To simplify notation, I omit the subscript t when the time information is clear from context. I use variables with hat (e.g., $\hat{h}_{1;ik}$ and $\hat{H}_1$) to denote the values obtained from the previous iteration.

In the E-step, I compute the expectation as

$$\phi_{ijk} = \hat{h}_{1;ik}\hat{h}_{2;jk}/(\hat{H}_1\hat{H}_2^T)_{ij}, \tag{9}$$

where $\sum_k \phi_{ijk} = 1$, $\hat{h}_{1;ik}$ and $\hat{h}_{2;jk}$ denote the ikth and the jkth entries, respectively, of H_1 and H_2 computed from the previous iteration.

In the M-step, I maximize the expectation of log likelihood with respect to $\Phi = (\Phi)_{ijk}$

$$\begin{aligned} E_\Phi[L] &= 2 \times \sum_{ijk} \phi_{ijk} a_{ij}^t \log(h_{1;ik}^t h_{2;jk}^t) \\ &+ 2 \times \nu \sum_{ijk} h_{1;ik}^{t-1} h_{2;jk}^{t-1} \log(h_{1;ik}^t h_{2;jk}^t), \end{aligned} \tag{10}$$

where the superscripts t and $t-1$ are used to denote variables at the corresponding time points. To facilitate a probabilistic interpretation of the co-clustering results, I impose the following normalization constraints:

$$\sum_i h_{1;ik}^t = 1, \quad \sum_j h_{2;jk}^t = 1.$$

By using Lagrange multipliers for these constraints, it can be shown that the following update rules will monotonically increase the expected log likelihood defined

in Eq. (10), thereby leading to convergence to an locally optimal solution [40]:

$$h_{1;ik} \leftarrow 2 \times \sum_j \frac{\hat{h}_{1;ik}\hat{h}_{2;jk}a^t_{ij}}{(\hat{H}_1\hat{H}_2^T)_{ij}} + 2 \times \nu \sum_j (h^{t-1}_{1;ik}h^{t-1}_{2;jk}),$$

$$h_{2;jk} \leftarrow 2 \times \sum_i \frac{\hat{h}_{1;ik}\hat{h}_{2;jk}a^t_{ij}}{(\hat{H}_1\hat{H}_2^T)_{ij}} + 2 \times \nu \sum_i (h^{t-1}_{1;ik}h^{t-1}_{2;jk}).$$

The results are then normalized such that $\sum_i h^t_{1;ik} = 1$ and $\sum_j h^t_{2;jk} = 1$, $\forall k$.

The E-step and and M-step are repeated until a locally optimal solution is obtained. Then the matrices $H_{1,t}$ and $H_{2,t}$ can be used as row and column co-cluster indicator matrices, respectively, to obtain soft co-clustering results. My experimental results show that this probabilistic model achieves superior performance on both synthetic and real data sets.

2.2.3 CO-CLUSTER EVOLUTION

An unique property of the proposed probabilistic model is that the identified co-clusters can be related across time points, giving rise to co-cluster evolution. Figure 1 shows how co-clusters evolve for a 5×4 example data matrix, where r_1 to r_5 correspond to the five rows, c_1 to c_4 correspond to the four columns, and R_1 to R_4 denote the co-clusters. In panel (a), the matrix is co-clustered into 3 co-clusters as indicated by the dashed ovals. At time t in panel (b), the data is clustered into 4 co-clusters. The row and column co-clusters across time points can be related naturally by considering the sharing of rows and columns between co-clusters. This is illustrated in panels (c) and (d), which depict how the row and column co-clusters, respectively, evolves from time points $t-1$ to t. Note that the co-cluster evolution is a direct product of the soft co-cluster assignment proposed in this dissertation.

This demonstrates that the soft co-cluster assignment formalism captures additional temporal dynamics, which have been ignored by prior methods. More importantly, I show in Section 2.4 that my evolutionary soft co-clustering formulation outperforms prior methods consistently.

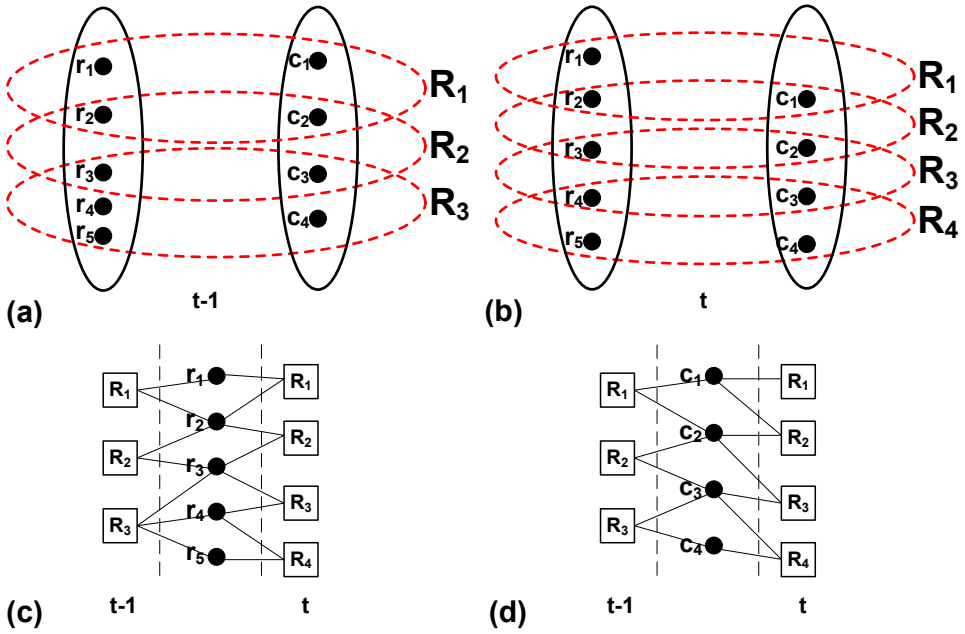


FIG. 1: Illustration of co-cluster evolution. Panels (a) and (b) show the co-clustering results at time points $t-1$ and t, respectively. Panels (c) and (d) show the row and column co-cluster evolution, respectively, between time points $t-1$ and t. See text for detailed explanations.

2.3 RELATED WORK AND EXTENSIONS

Following the evolutionary spectral clustering framework in [9], two spectral methods for evolutionary co-clustering have been proposed in [41]. In this section, I systematically extend the spectral methods in [41] using two different graph cut criteria,

leading to four different methods for capturing the temporal smoothness. My experimental results in Section 2.4 show that the probabilistic model proposed in this dissertation consistently outperforms the spectral methods.

2.3.1 PRESERVING CO-CLUSTER QUALITY

In preserving co-cluster quality (PCCQ), the temporal cost measures the quality of current co-clustering results when applied to historic data. In the following, I describe the PCCQ formalism using both the negated average association and the normalized cut criteria.

Negated Average Association

Given a data matrix $A \in \mathbb{R}^{m \times n}$, the negated average association objective function in co-clustering can be written as

$$NA = \mathrm{Tr}(W) - \mathrm{Tr}(\tilde{R}^T W \tilde{R}), \tag{11}$$

where $\tilde{R} \in \mathbb{R}^{(m+n) \times k}$ is the normalized co-cluster indicator matrix, W is defined in Eq. (2) and denotes the similarity matrix associated with the bipartite graph. Writing $\tilde{R} = [P^T, Q^T]^T$, where $P \in \mathbb{R}^{m \times k}$ and $Q \in \mathbb{R}^{n \times k}$ are the row and column cluster indicator matrices, respectively, and substituting W into Eq. (11), I obtain

$$NA = -\mathrm{Tr}(P^T A^T Q + P^T A Q) = -2\mathrm{Tr}(P^T A Q). \tag{12}$$

I propose to employ the following cost function for the PCCQ evolutionary co-clustering formalism based on negated average association:

$$\mathrm{NA}_{\mathrm{PCCQ}} = \alpha \cdot NA_t|_{\tilde{R}_t} + (1-\alpha) \cdot NA_{t-1}|_{\tilde{R}_t} = -\mathrm{Tr}\left(P_t^T \left(\alpha A_t + (1-\alpha) A_{t-1}\right) Q_t\right),$$

where A_t, P_t, and Q_t denote the corresponding matrices for time point t. Since solving the above problem exactly is intractable, I propose to relax the constraints on the entries in P_t and Q_t while keeping the orthonormality constraints. It follows from the spectral co-clustering formalism [38] that columns of the optimal P_t^* and Q_t^* that minimize the relaxed problem are given by the k principal left and right, respectively, singular vectors of the matrix $\alpha A_t + (1-\alpha)A_{t-1}$.

Normalized Cut

It follows from Proposition 1 in [43] that the normalized cut criterion can be expressed equivalently as

$$NC = k - \mathrm{Tr}\left(S^T(D^{-\frac{1}{2}}WD^{-\frac{1}{2}})S\right), \tag{13}$$

where

$$D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}, \quad W = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}, \tag{14}$$

and $S \in \mathbb{R}^{(m+n)\times k}$ satisfies two conditions: (a) the columns of $D^{-1/2}S$ are piecewise constant with respect to R, and (b) $S^TS = I$. Let $S = [E^T, F^T]^T$, where $E \in \mathbb{R}^{m\times k}$ and $F \in \mathbb{R}^{n\times k}$, then the normalized cut criterion in Eq. (13) can be written as $NC = k - 2\mathrm{Tr}\left(E^T(D_1^{-1/2}AD_2^{-1/2})F\right)$.

I propose to employ the following cost function in PCCQ under the normalized cut criterion:

$$\begin{aligned} \mathrm{NC}_{\mathrm{PCCQ}} &= \alpha \cdot NC_t|_{S_t} + (1-\alpha)\cdot NC_{t-1}|_{S_t} \\ &= k - 2\mathrm{Tr}\left(E_t^T(\alpha D_{1,t}^{-1/2}A_tD_{2,t}^{-1/2} + (1-\alpha)D_{1,t-1}^{-1/2}A_{t-1}D_{2,t-1}^{-1/2})F_t\right), \end{aligned}$$

where $D_{1,t}$ and $D_{2,t}$ are the diagonal matrices at time t. Similar to the case of negated average association, I relax the constraints on the entries of E_t and F_t while keep the orthonormality constraints. It can be verified that columns of the optimal E_t^* and F_t^* that minimize the relaxed problem consist of the principal left and right, respectively, singular vectors of $\alpha D_{1,t}^{-1/2} A_t D_{2,t}^{-1/2} + (1-\alpha) D_{1,t-1}^{-1/2} A_{t-1} D_{2,t-1}^{-1/2}$. Then the rows of the matrix $\left[\; (D_{1,t}^{-1/2} E_t^*)^T, (D_{2,t}^{-1/2} F_t^*)^T \; \right]^T$ are clustered to identify co-clusters.

2.3.2 PRESERVING CO-CLUSTER MEMBERSHIP

In preserving co-cluster membership (PCCM), the temporal cost measures the consistency between temporally adjacent co-clustering results. Let U_t and V_t denote the solutions of the relaxed problems at time point t as described in Section 2.3.1. Note that columns of U_t and V_t are the left and right singular vectors, respectively, of certain matrix. Since the singular vectors of a matrix may not be unique [44], I cannot require U_t and U_{t-1} to be similar and V_t and V_{t-1} to be similar. however, it is known that $U_t V_t^T$ is unique in all cases. I propose to employ the following temporal cost in PCCM:

$$\mathrm{CT}_{\mathrm{PCCM}} = \|U_t V_t^T - U_{t-1} V_{t-1}^T\|_F^2. \tag{15}$$

Negated Average Association

By using the temporal cost in Eq. (15) to quantify the smoothness, I propose the following overall cost function for PCCM under the negated average association criterion: $\mathrm{NA}_{\mathrm{PCCM}} = \alpha \cdot \mathrm{CS}_{\mathrm{NA}} + (1-\alpha) \cdot \mathrm{CT}_{\mathrm{PCCM}} = 2(1-\alpha)k - 2\mathrm{Tr}\left(U_t^T \left(\alpha A_t + (1-\alpha) U_{t-1} V_{t-1}^T\right) V_t\right)$. Maximizing

$\text{Tr}\left(U_t^T\left(\alpha A_t + (1-\alpha)U_{t-1}V_{t-1}^T\right)V_t\right)$ is equivalent to minimizing NA_{PCCM}. Hence, columns of the the optimal U_t^* and V_t^* consist of the principal left and right singular vectors, respectively, of the matrix $\alpha A_t + (1-\alpha)U_{t-1}V_{t-1}^T$.

Normalized Cut

When the temporal cost in Eq. (15) is used along with the normalized cut criterion, I obtain the following problem:

$$\begin{aligned}\text{NC}_{\text{PCCM}} &= (2-\alpha)k \\ -2\text{Tr}&\left(U_t^T\left(\alpha D_{1,t}^{-1/2}A_t D_{2,t}^{-1/2} + (1-\alpha)U_{t-1}V_{t-1}^T\right)V_t\right).\end{aligned}$$

Minimizing NC_{PCCM} is equivalent to maximizing

$$\text{Tr}\left(U_t^T\left(\alpha D_{1,t}^{-1/2}A_t D_{2,t}^{-1/2} + (1-\alpha)U_{t-1}V_{t-1}^T\right)V_t\right).$$

Hence, columns of the the optimal U_t^* and V_t^* consist of the principal left and right singular vectors, respectively, of the matrix $\alpha D_{1,t}^{-1/2}A_t D_{2,t}^{-1/2} + (1-\alpha)U_{t-1}V_{t-1}^T$. The final co-clusters are obtained by clustering the rows of the matrix $\begin{bmatrix} D_{1,t}^{-1/2}U_t^* \\ D_{2,t}^{-1/2}V_t^* \end{bmatrix}$.

2.4 EXPERIMENTAL EVALUATION

2.4.1 SYNTHETIC DATA # 1

I generate a synthetic data set with 7 time-steps and 5 co-clusters, each containing 200 instances and 10 features. At $t = 0$, the entries corresponding to rows and columns in the same co-cluster are set to nonzero with a high probability p while

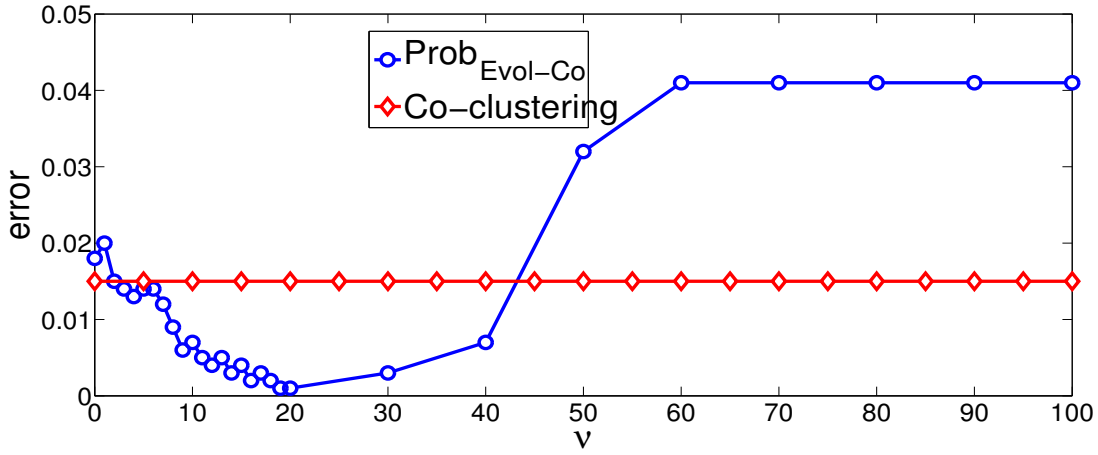


FIG. 2: Performance comparison between the proposed probabilistic model ($\text{Prob}_{\text{Evol-Co}}$) with that of the co-clustering method when ν varies from 0 to 100.

other entries are set to nonzero with a low probability q which satisfies $p = 4q$ and $p + 4q = 1$. The data at $t = 1$ are generated by adding a Gaussian noise to each entry of the data at $t = 0$. To simulate the evolving nature of the data, 20% of the instances in co-cluster I are set to be weakly correlated to features in co-cluster III at $t = 2$. The level of correlation by the same set of instances is increased at $t = 3$ so that they are equally correlated to features in co-clusters I and III. At $t = 4$, this set of instances are no longer correlated to features in co-cluster I, and their correlations with features in co-cluster III are further increased. At $t = 5$, a sudden change occurs and the data matrix at $t = 1$ is restored. At $t = 6$, the size of the data matrix is changed by adding some extra instances to co-cluster I.

To demonstrate the effectiveness of the temporal cost, I compare my formulation with co-clustering method without the temporal cost. I use error rate as the performance measure, since the co-cluster memberships are known for synthetic data. The performance of the proposed model along with that of the co-clustering method

(equivalent to $\nu = 0$) is reported in Figure 2. It can be observed that when ν is increased from 0 to 20, the error rate drops gradually. When ν is increased beyond 20, the error rate increases gradually. When ν lies in the interval $[5, 40]$, the proposed method outperforms the co-clustering method significantly. This shows that the evolutionary co-clustering formulation yields improved performance for a large range of ν.

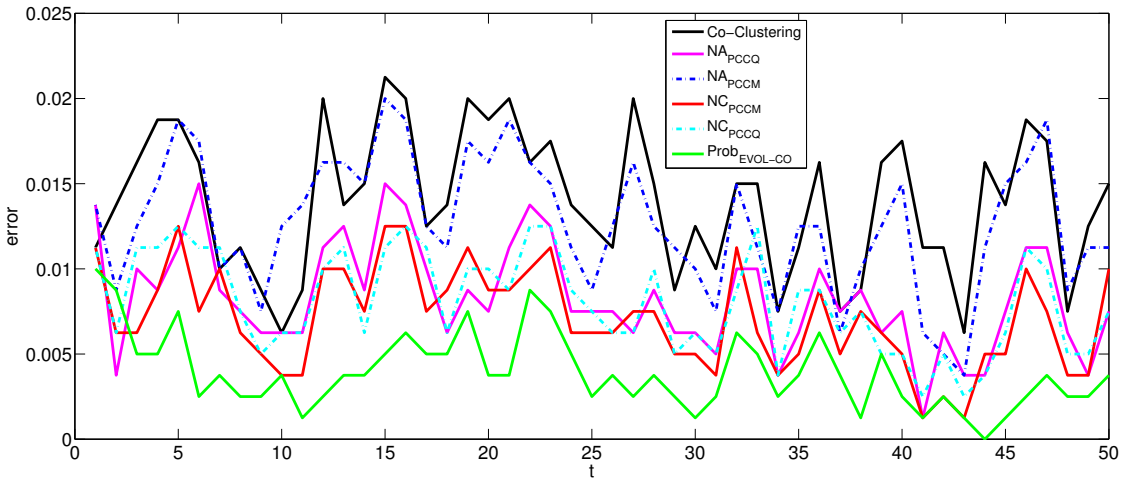
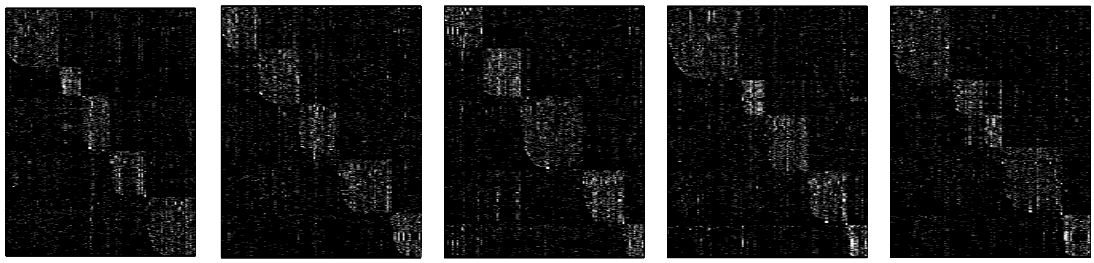


FIG. 3: Performance of the probabilistic model with four methods based on spectral learning and the co-clustering method on synthetic data # 2.



1998 – 1999 2000 – 2001 2002 – 2003 2004 – 2005 2006 – 2007

FIG. 4: The block structures identified by the proposed probabilistic model on the DBLP data.

2.4.2 SYNTHETIC DATA # 2

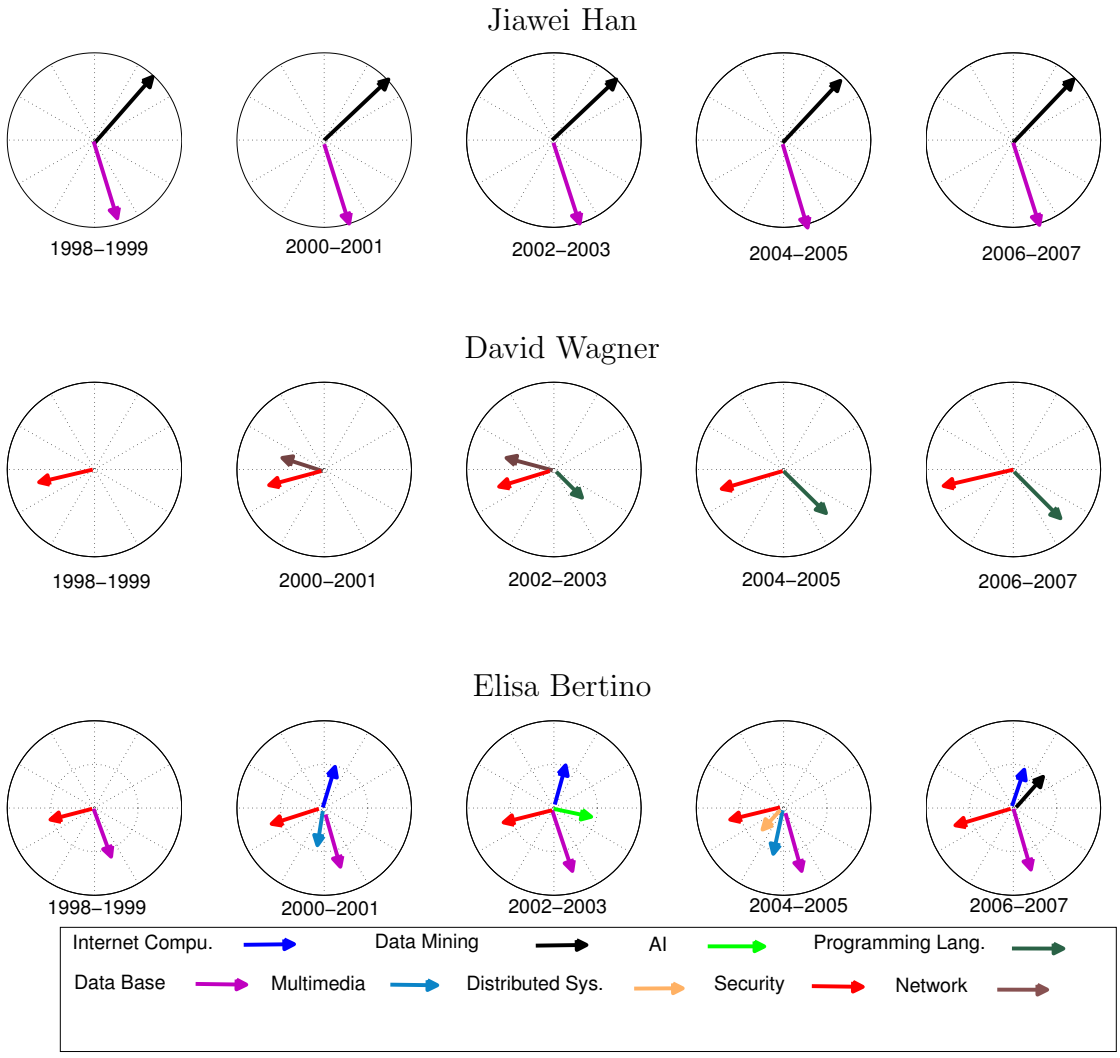


FIG. 5: The evolution patterns of three authors identified by the proposed probabilistic model.

The second synthetic data set is generated to evaluate the performance of the proposed model in comparison to prior methods based on spectral learning. This data set contains 50 time-steps, each with 4 co-clusters, and each co-cluster contains 100 instances and 10 features. At $t = 0$, the data set is generated by following the

same strategy as the first synthetic data set when $t = 0$. In each of the 0 to 49 time-steps, I add Gaussian noise to the data from previous time-step. I optimize the α and ν values on the synthetic data separately. This set of experiments, including data generation, are repeated 40 times and the average results are reported in Figure 3 for all time-steps. I can observe from Figure 3 that the proposed probabilistic model ($\text{Prob}_{\text{EVOL-CO}}$) consistently outperforms prior methods (i.e., NA_{PCCQ}, NC_{PCCQ}, NA_{PCCM}, and NC_{PCCM}). This demonstrates that the proposed model is very effective in improving performance by requiring the factors to be nonnegative. Similar to the observation in Section 2.4.1, all evolutionary co-clustering approaches outperform co-clustering method consistently across most time-steps. This demonstrates that the temporal cost is effective in improving performance.

2.4.3 DBLP DATA

I conduct experiments on the DBLP data to evaluate the proposed methods. The DBLP data [4, 11] contain the author-conference information for 418,236 authors and 3,571 conferences during 1959-2007. For each year, the author-conference matrix captures how many papers are published by an author in a conference. The author-conference data matrices are very sparse, and I sample 252 conferences spanning 12 main research areas (Internet Computing, Data Mining, Machine Learning, AI, Programming Language, Data Base, Multimedia, Distributed System, Security, Network, Social Network, Operating System) in my experiments. I also remove authors with too few papers, resulting in 4147 authors from the 252 conferences. I choose the data for ten years (1998-2007) and add the data for two consecutive years,

leading to data of five time points.

I apply the probabilistic model to the DBLP data in order to discover the author-conference co-occurrence relationship and their temporal evolution. I set the number of co-clusters to be 12 in the experiments, and this results in 5 major co-clusters and 7 minor co-clusters as shown in Figure 4. The 5 major co-clusters can be easily identified from my co-clustering results, and their evolutions are temporally smooth. A close examination of the results shows that related conferences are clustered into the same co-cluster consistently across all time points. For example, the co-cluster for Data Mining always contains KDD, ICDM, SDM etc., and the co-cluster for Data Base always contains SIGMOD, ICDE, VLDB, etc.

I also investigate how the authors' research interests change dynamically over time. In Figure 5, I plot the results for three authors: Jiawei Han, David Wagner, and Elisa Bertino. For each author and each time point, I distribute the 12 conference categories evenly around a circle, and each category occupies a sector. I then use an arrow pointing to a particular sector to indicate the author's participation in the conferences in this category, where the level of participation is indicated by the length of the arrow.

It can be observed from Figure 5 that Jiawei Han was actively participating Data Mining and Data Base conferences across all five time points, and this pattern remains very stable across years. On the other hand, David Wagner showed some change of research interests. He is actively participating Security conferences across all years. During 2000-2001, he developed interests in Network, and this is maintained through

2002-2003 before he smoothly switched to Programming Language. Elisa Bertino showed very dynamic change of research interests during this ten-year period. She is actively participating Data Base and Security conferences across all years. During some period of time, she also participated Internet Computing, Distributed Systems, AI, and Data Mining conferences. These results demonstrate that the proposed methods can identify smooth evolution of author's research interests over years.

CHAPTER 3

DROSOPHILA GENE EXPRESSION PATTERN IMAGE ANALYSIS

To fully exploit the real-world impact of my methods, I perform a systematic application study on the analysis of *Drosophila* gene expression pattern images.

3.1 BACKGROUND

Genes are fundamental elements for regulating many biological activities from cell division to protein composition. The continuous progress of the gene identification from DNA sequences has required continuous improvements in both the experimental techniques and computational algorithms. however, how these sequences are transformed from a single cell during the development, into a functionality organism remains largely unknown. Discovering gene expression in temporal and spatial patterns is essential for understanding the regulatory biology. In sequencing and gene-prediction technologies, advances have led to broad research areas of protein-coding sequences in many model systems. Recently, during the development of *Drosophila* melanogaster, systematic analysis on annotated gene expression already focuses on the high-throughput RNA in situ hybridization to generate a database of gene expression patterns [14–16, 18]. This database provides useful information to discover the temporal and spatial gene expression patterns in the regulatory networks and development [17–19].

In this chapter, I develop a set of ISH image computing and machine learning methods for the automated analysis of *Drosophila* gene expression pattern images. Specifically, I develop a mesh generation pipeline for mapping the expression patterns of many genes into the same geometric space [18]. This enables accurate comparative analysis of the spatial expression patterns of multiple genes and accounts for the differences in embryo morphology. I fit an ellipsoid to the boundary of each embryo using the least squares criterion. I then average the fitted ellipsoids for all images in the same stage range to obtain a generic ellipsoid. I automatically interpolate the boundary of this generic ellipsoid and use a Delaunay mesh method [45–48] to generate triangulated mesh on this ellipsoid.

I accurately capture the morphology of each embryo by employing a systematic procedure to deform the generic, meshed ellipsoid to each individual embryo. I first establish correspondences between vertices on the generic ellipsoid and those on the fitted ellipsoids. Then the vertices on the fitted ellipsoids are deformed to the embryo boundary using the minimum distance criterion. Finally, the coordinates of all the other vertices are computed by solving an elastic finite element problem.

The mesh generation scheme allows me to organize the expression pattern images of many genes into a data matrix in which one dimension corresponds to genes and the other dimension corresponds to mesh elements as in the Genomewide-Expression-Maps (GEMs) [20, 49]. To identify co-expressed embryonic domains and the associated genes, I apply my proposed evolutionary co-clustering formulation to cluster the mesh elements and the genes simultaneously.

I apply the mesh generation and co-clustering methods to a set of gene expression pattern images in the FlyExpress database [20]. My results show that my methods generate co-expressed domains that overlap with many embryonic structures. In addition, these results show that the proposed methods yield gene clusters that are functionally more related than those discovered in prior studies. More importantly, I show that the mesh and gene co-clusters correlate strongly with key developmental events during the stage of embryogenesis under investigation.

3.2 MESH GENERATION

3.2.1 REQUIREMENTS

Let $I_1, \ldots, I_m$ be a list of embryo images. The goal of this module of the pipeline is to overlay each of the embryo images with a triangular mesh, such that all meshes have the same number of triangles and connectivity. For a given image, all triangles I create are of approximately the same size, in terms of their area. Let a stand for an upper bound on triangle area. Then all triangles in a single mesh which I construct have area slightly less than a. Let $M_j(a)$ be the mesh that I construct for image I_j that depends on area bound a. For simplicity I will omit the parameter a below.

More precisely, let $M_j = (V_j, T_j)$, where V_j is the list of vertices and T_j is the list of triangles. Each vertex is defined by its two-dimensional coordinate, and each triangle is defined by a triple of vertex indices (p_1, p_2, p_3), $1 \leq p_1, p_2, p_3 \leq |V_j|$. These meshes are expected to satisfy the following requirements:

- All of the T_j contain the same number of triangles, *i.e.*, $|T_j| = |T_i|$ for $i, j =$

$1, \ldots, m$.

- All of the T_j contain the same triples of vertex indices in the corresponding positions. As a result, I can omit the subscript and use T for all meshes M_j, $j = 1, \ldots, m$.

- All of the V_j contain the same number of vertices: $|V_j| = |V_i|$ for $i, j = 1, \ldots, m$.

- All vertices on the boundary of mesh M_j lie on the boundary of the embryo of image I_j.

- Each triangle in $M_j = M_j(a)$ has area approximately equal to a.

- All vertices in V_j are geometrically close to the vertices in the corresponding positions in V_i for all $i, j = 1, \ldots, m$, with respect to their location within an embryo.

3.2.2 CONSTRUCTION AND MESHING OF THE AVERAGE ELLIPSE

For each image I_j, $j = 1, \ldots, m$, I compute the parameters of the equation of the ellipsoid E_j that realizes the best fit to the boundary of the embryo in this image. I compute the best fitted ellipsoid using the least squares criterion to the set of the embryo's boundary pixels. Then I average the parameters of all ellipsoids to obtain the average ellipsoid E'.

Given a value of a, I construct a mesh of E'. First, I use linear interpolation to approximate the boundary of E', and then use a Delaunay mesh generator, Triangle [45], to mesh the interior of E'. Delaunay refinement is my meshing method

of choice since it is backed by proven theoretical guarantees [46–48] that make it a push-button technology: its being able to guarantee termination with angle and area bounds allow for a guaranteed quality automatic pipeline.

I interpolate the boundary of E' by performing the following steps. First, I calculate the side length ℓ of an equilateral triangle with area a. Then I use an iterative subdivision of the boundary of E' with a set of vertices $v_1, \ldots, v_s = v_0$ until all segment lengths $|v_{i-1}v_i|$, $i = 1, \ldots, s$ are approximately equal to ℓ. In other words, this is a uniform distribution of vertices with respect to the lengths of segments. The union of all these segments is a piecewise linear interpolation of the boundary of E'.

To tessellate the interior of E', I use Triangle with the following parameters:

- A planar straight line graph (PSLG) composed of the segments and the points interpolating the boundary of E' plus one point in the center of E'. I instruct Triangle to preserve this PSLG and not to split the boundary segments, so that the discretization of the PSLG appears as a subgraph of the final mesh.

- The area bound a instructing Triangle to produce all triangles with areas bounded from above by a. Triangle starts with a coarse mesh and iteratively splits triangles until their areas fall below a, and therefore this is an approximate target area.

- An angle bound of 25° which instructs Triangle to enforce all angles in the final mesh to be 25° or above. Theoretically, Triangle guarantees only a minimum angle bound of 20.7° or below, however I find that in practice it can mesh an ellipsoid with a 25° angle bound, since it is a simple shape.

Let the mesh of the average ellipsoid be denoted as M', and the list of radial angles corresponding to the subdivision vertices as $\theta'_1, \ldots, \theta'_s$.

3.2.3 DEFORMATION OF THE MESH OF THE AVERAGE ELLIPSE

For each ellipsoid E_j, I use the angles $\theta'_1, \ldots, \theta'_s$ to find the vertices that discretize the boundary of E_j. Then I project these vertices onto the closest points from the boundary of the embryo in image I_j. I define closeness in terms of the Euclidean distance, and use the Matlab's Euclidean distance transform function to find the nearest boundary pixels simultaneously for all pixels in the image. Using the result of this function, I determine the required projections.

For each image I_j, I deform the mesh M', such that the boundary vertices of M' assume the coordinates of the corresponding vertices (with respect to their radial ordering) on the boundary of the embryo in I_j. The target coordinates of all the other vertices in V' are computed by solving an elastic finite element problem [50]. As a result, the triangles of the generic mesh are deformed minimally and proportionally to their distance to the projected vertices on the boundary of the embryo in I_j and to the amount of the displacement at these boundary vertices.

3.3 RELATED WORK

My work on mesh generation is motivated by the prior work in [18]. however, there are some substantial differences between my approach and the prior method. Besides the expanded analysis based on meshes with a range of triangle sizes, for a given triangle size a my methodology also offers a number of significant improvements in

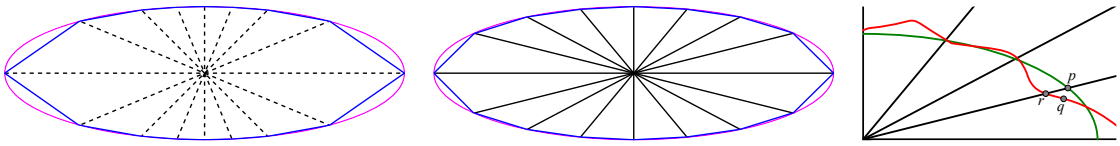


FIG. 6: **Left:** Subdivision of an ellipse (pink) based on equal radial angles (dashed black lines) leads to inaccurate boundary interpolation (blue). **Center:** A more accurate subdivision (solid black lines) based on equal lengths of interpolating segments. **Right:** Euclidean projection (q) from a point (p) on the ellipse (green) onto the boundary of the embryo (red) is more accurate than a projection along a radial line (r).

the accuracy of capturing embryo shapes. Frise *et al.* [18] define E' as a predetermined ellipsoid of axial ratio 4 : 2, while I compute E' from the actual embryo shapes. As a result, I make sure that E' is close to the particular set of shapes, since different sets of shapes can have different average ellipsoids. Frise *et al.* [18] discretize the boundary of E' based on approximately equal radial angles, while my discretization is based on approximately equal edge lengths. See Figure 6 (left and center) for an illustration. Frise *et al.* [18] project the discretization vertices from E_j onto the actual boundary of the embryo along the radial lines emanating from the center of E_j, while I choose the closest points based on Euclidean distance. See Figure 6 (right) for an illustration.

My work is related to the seminal work in [51], where the Gaussian mixture models (GMM) were applied to generate co-expression domains for the purpose of image comparison. My work is different from [51] in both its objectives and approaches.

In [51], image pixels were considered directly as the basic elements of modeling while I use triangulated mesh to warp and discretize the embryos in order to account for the shape and morphological variations. It has been shown in prior work [18] that the use of mesh leads to biologically significant results. In addition, GMM was used to cluster the pixels in [51], while I use a co-clustering method to co-cluster the mesh elements and the genes simultaneously. Since each domain is expected to be defined by only a subset of genes in the genome, co-clustering aims at identifying the domains and the associated genes simultaneously. As shown by my experimental results, co-clustering leads to more significant results.

3.4 EXPERIMENTAL EVALUATION

3.4.1 CLUSTERING OF MESH ELEMENTS

The mesh elements represent localized spatial areas of the embryo, and can be used to discover distinct domains of developmental gene expression. I apply my mesh method to the data set of 553 stage 4-6 lateral embryos to gain insight into major developmental co-expression domains during this time. Co-clustering with different numbers of co-clusters is applied to the data matrix. Results are then mapped to the average ellipsoid and color-coded (Figure 7 and 8). To ensure that cluster boundaries are not the result of data processing artifacts, data is randomized at multiple points of the pipeline.

Figure 7 and 8 reveal the resulting clusters resemble the fate map of the developing embryo [52]. The clusters represent domains of high co-expression. They

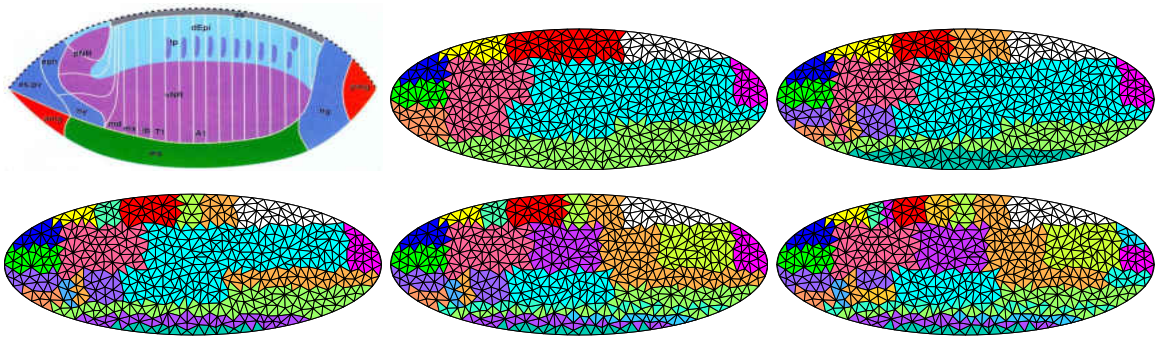


FIG. 7: Clusters of mesh elements when the number of clusters is varied from 10 to 30 with a step size of 5 (left to right, top to bottom) on stage 4-6 expression patterns. The first figure at the first row shows the fate map of the blastoderm [52].

invariably form spatially contiguous regions, and are composed of rectangular shapes. Further, the cluster boundaries are largely parallel to the anterior/posterior (A/P) and dorsal/ventral (D/V) axes of the embryo. As the number of co-clusters is increased (Figure 7), the rectangular cluster shape is often retained, with larger clusters subdivided into smaller ones. In my data set, this subdivision of clusters often occurred at the far A/P and D/V regions of the embryo. These increased subdivisions correlate with major developmental events during stages 4-6 of *Drosophila* embryogenesis [52, 53]. Signals along the A/P and D/V axes drive this pattern formation [54]. During Stage 6 gastrulation begins, and the ventral and cephalic furrows form. Looking back at the clusters, I see a greater proportion of subdivisions along where these furrows form in the developing embryo. The general clustering patterns remain the same while the cluster boundaries become smoother as the number of mesh elements increases (Figure 8).

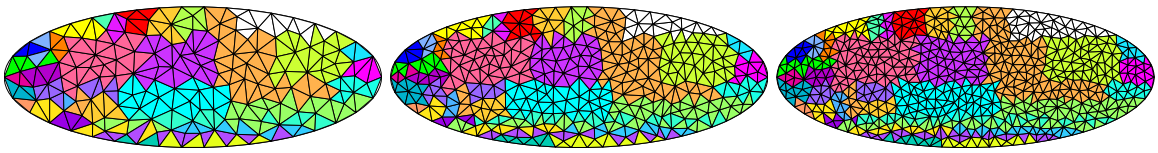


FIG. 8: Clusters of mesh elements when the number of co-clusters is set to 39 as in [18], and the number of mesh elements are set to 300, 600, and 1000 (left to right). In these figures, colors are used to visualize clusters so that mesh elements in the same cluster are in the same color, and those in different clusters are in different colors.

3.4.2 CLUSTERING OF GENES

Co-clustering of the data matrix leads to clusters of genes. I use gene ontology (GO) [55] to evaluate the gene clusters and compare the results with those reported in [18]. My gene clusters are the combined results of the mesh generation and co-clustering methods. Hence, I evaluate the effects of these two methods separately.

First, I compare my mesh generation method with the approach described in [18]. I apply both methods to the set of 553 images, yielding two data matrices. I then apply the co-clustering method with different numbers of co-clusters to these two data matrices. Since the same co-clustering method is used for both data matrices, the differences in the results should be contributed by differences in the mesh generation methods. I use the hypergeometric distribution to compute enriched GO terms [56] in order to evaluate the gene clusters generated from these two data matrices. The numbers of terms with p-values less than 0.001 are reported in Table 1. I can see that

TABLE 1: The numbers of enriched gene ontology terms generated by the original (Original) and the proposed (New) mesh generation methods. The number of co-clusters is varied from 30 to 40. In each case, the total number of enriched terms from all clusters are reported.

# of clusters	Biological process		Cellular component		Molecular function	
	New	Original	New	Original	New	Original
30	168	169	36	36	43	43
31	168	169	36	36	43	43
32	155	156	35	35	38	38
33	174	175	30	30	40	40
34	174	175	30	30	40	40
35	169	170	30	30	38	38
36	189	176	30	29	38	38
37	189	176	30	29	38	38
38	189	176	30	29	38	38
39	192	177	32	31	38	38
40	192	177	32	31	38	38

these two methods give similar numbers of biological process terms when the number of clusters is relatively small (30-35). however, as the number of cluster increases, my new mesh generation method yields larger numbers of enriched terms. This result shows that the new mesh generation approach and pipeline tools I developed are more accurate and can produce statistically more significant results when the number of clusters is large. I also observe that these two methods give similar numbers of cellular

component and molecular function terms in all cases. Since the numbers of enriched terms in these two categories are relatively small, the differences in mesh generation methods might not be significant enough to be reflected in these two categories.

I also compare my co-clustering approach with the affinity propagation method used in [18]. Namely, I compare my EM-based co-clustering method with the affinity propagation clustering by applying these two methods to the data matrix generated by my mesh using 553 images. The affinity propagation method automatically determines the number of clusters and yields 39 clusters on this data set [18]. I also apply my co-clustering method on this data set to generate 39 clusters. I then compute the number of enriched GO terms for each cluster, and the results are depicted in Figure 9. I can see that my co-clustering method is able to generate gene clusters that are functionally more related than those by the affinity propagation approach.

The significantly different results might be due to the fundamentally different approaches taken by the two studies. Specifically, Frise *et al.* [18] used clustering method to group the genes into clusters based on all the mesh elements. In another word, clustering method measures the expression patterns of genes across the whole embryo. That is, for two genes to be in the same cluster, they need to have similar expression patterns over the entire embryo. In comparison, I propose to use a co-clustering method, which identifies gene and mesh co-clusters simultaneously. In my approach, two genes can be grouped into the same cluster if they share similar local expression patterns. Note that co-clustering was mainly motivated from gene expression studies [57], and my results show that co-clustering method yields statistically

more significant results.

3.4.3 EVOLUTIONARY CLUSTERING OF MESH ELEMENTS

I apply my methods to all of the *Drosophila* gene expression pattern images from stage 4-6 to stage 13-16 to gain insight on the developmental gene co-expression dynamics. Evolutionary co-clustering with different numbers of co-clusters is applied to the five data matrices simultaneously. The results are mapped to the average ellipsoid and color-coded to visualize the co-clusters. In order to make sure that the generated clusters are not the results of data processing artifacts, I randomize the data sets at multiple points of the pipeline. My results show that the co-expressed domains established via my evolutionary co-clustering algorithm are consistent with many actual embryonic structures. Moreover, I show that the co-clusters of mesh elements and genes have strong correlation with the key events of *Drosophila* embryogenesis.

In Figure 10, I show the co-clustering results of mesh elements when the number of clusters is varied from 20 to 40 on stage 4-6 data. A number of existing co-clustering techniques also aim to identifying the block structures. In particular, I compare my evolutionary co-clustering method with a variant of the minimum sum-squared residue co-clustering (MSSRCC) method [58]; namely NBIN+RI+MSSRCC+LS, which denotes MSSRCC with random initialization, local search, and data binormalization [59], since different variants of MSSRCC generate similar results. I can observe that the co-clustering boundaries of the proposed method are mostly parallel to the anterior/posterior (A/P) and dorsal/ventral (D/V) axes of the embryo. This is

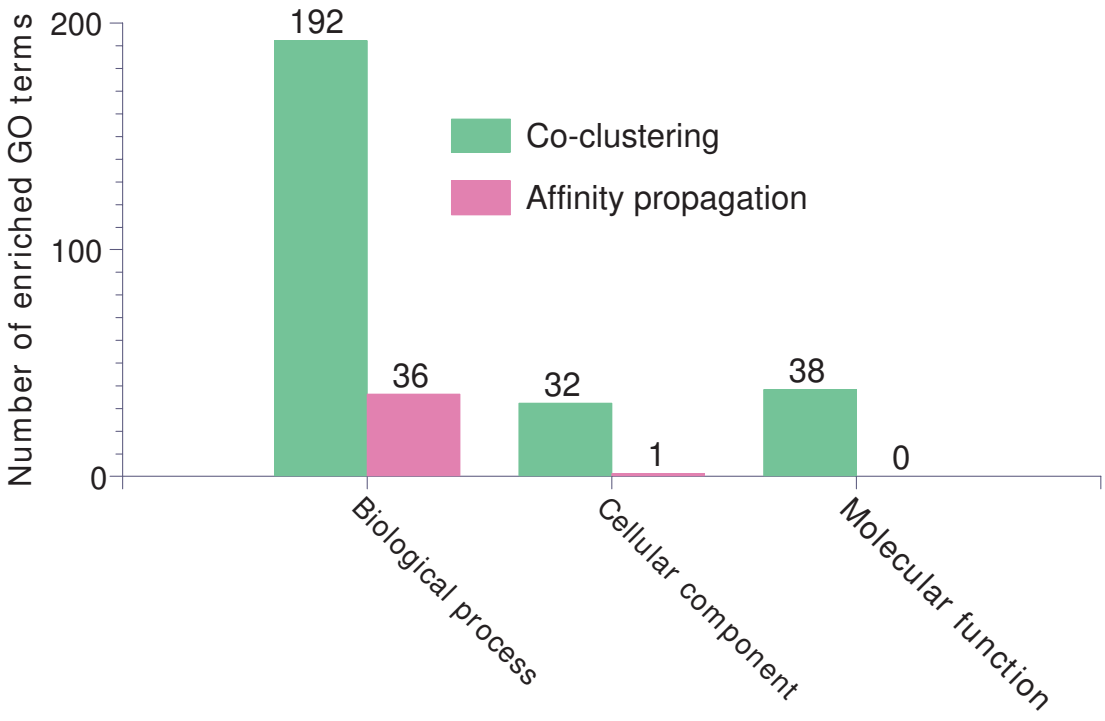


FIG. 9: Comparison of the total numbers of enriched gene ontology terms obtained from my co-clustering method and the affinity propagation method used in [18]. The reported numbers here are the total number of terms in each cluster.

consistent with the underlying biology of *Drosophila* embryonic patterning, which is achieved by two sets of systems along the horizontal and vertical axis independently ([52] and Figure 12). Furthermore, as the number of co-clusters is increased, the shape of rectangular cluster generated by my method is continuously preserved (the left column of Figure 10); namely, new clusters are generated by subdividing existing clusters, and all other clusters are preserved. In comparison, the cluster boundaries generated by MSSRCC do not align with the horizontal or vertical axes. Additionally, the cluster boundaries generated by MSSRCC are mostly not preserved when the number of clusters varies.

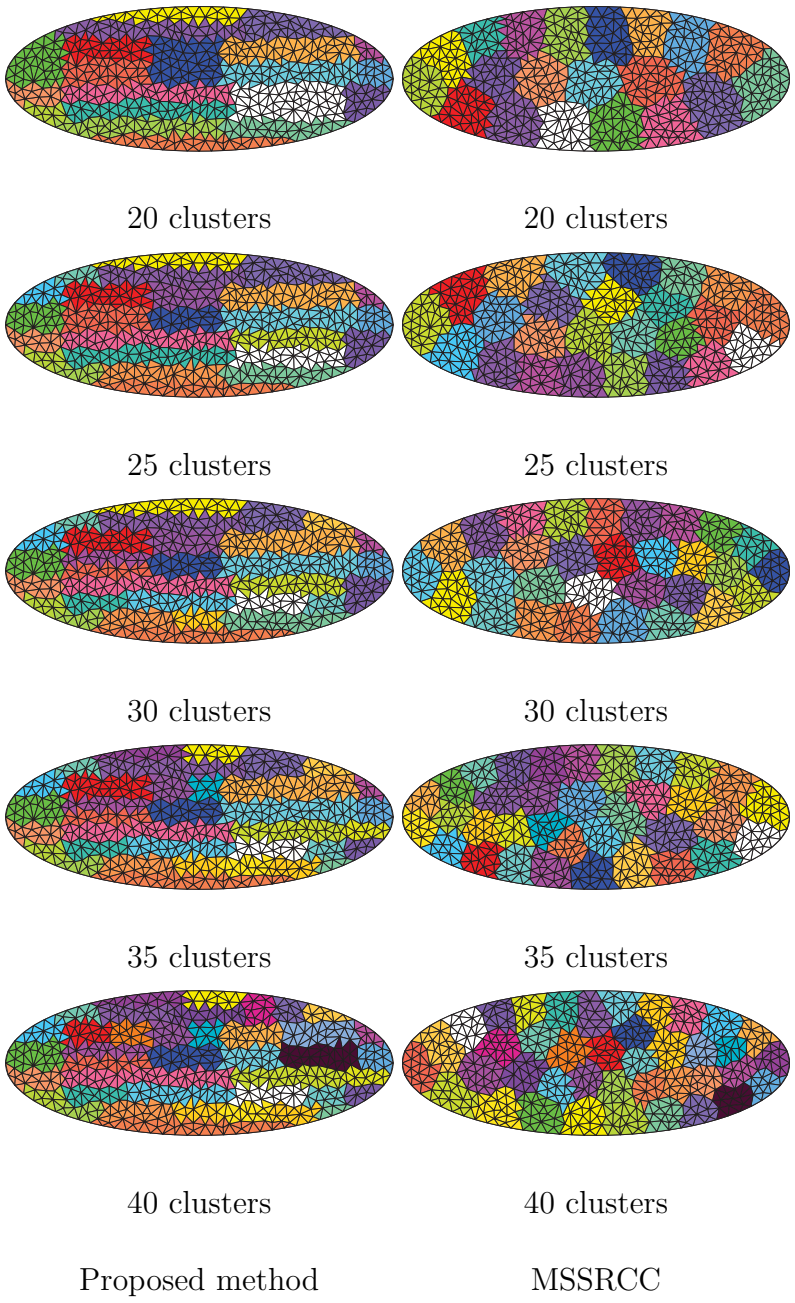


FIG. 10: Clusters of mesh elements when the number of clusters is varied from 20 to 40 with a step size of 5 (top to bottom) on stage 4-6 expression patterns. The left column shows the results of the proposed method and the right column shows the results of NBIN+RI+MSSRCC+LS.

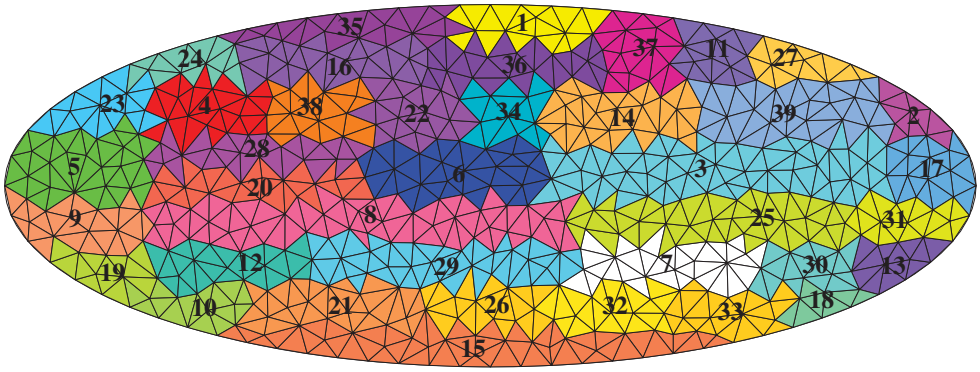


FIG. 11: Mesh clusters when the number of clusters is set to 39. Each mesh cluster element is labeled with the cluster number.

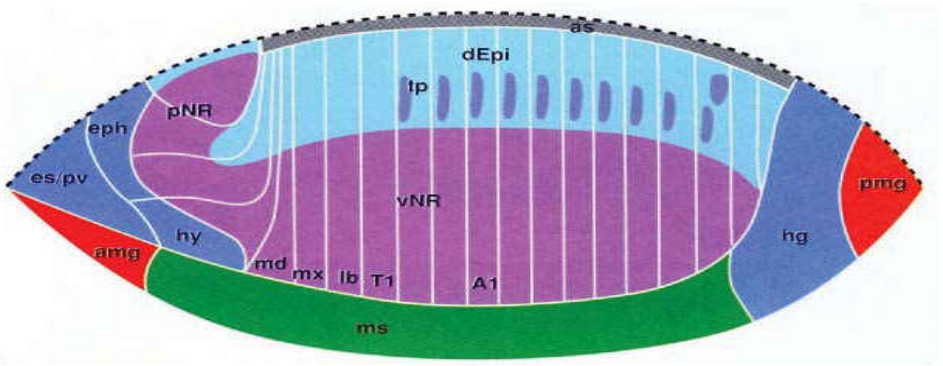


FIG. 12: The fate map of *Drosophila* blastoderm [52].

In Figure 14, I show the clustering results generated by my evolutionary co-clustering method and by NBIN+RI+MSSRCC+LS for the five stage range data (i.e., stage 4-6 to stage 13-16) when the number of clusters is fixed to 35. I can again observe that the clusters generated by my method usually have rectangular shapes whose sides are approximately aligned with the horizontal or vertical axes. In comparison, the results generated by MSSRCC do not have a rectangular shape.

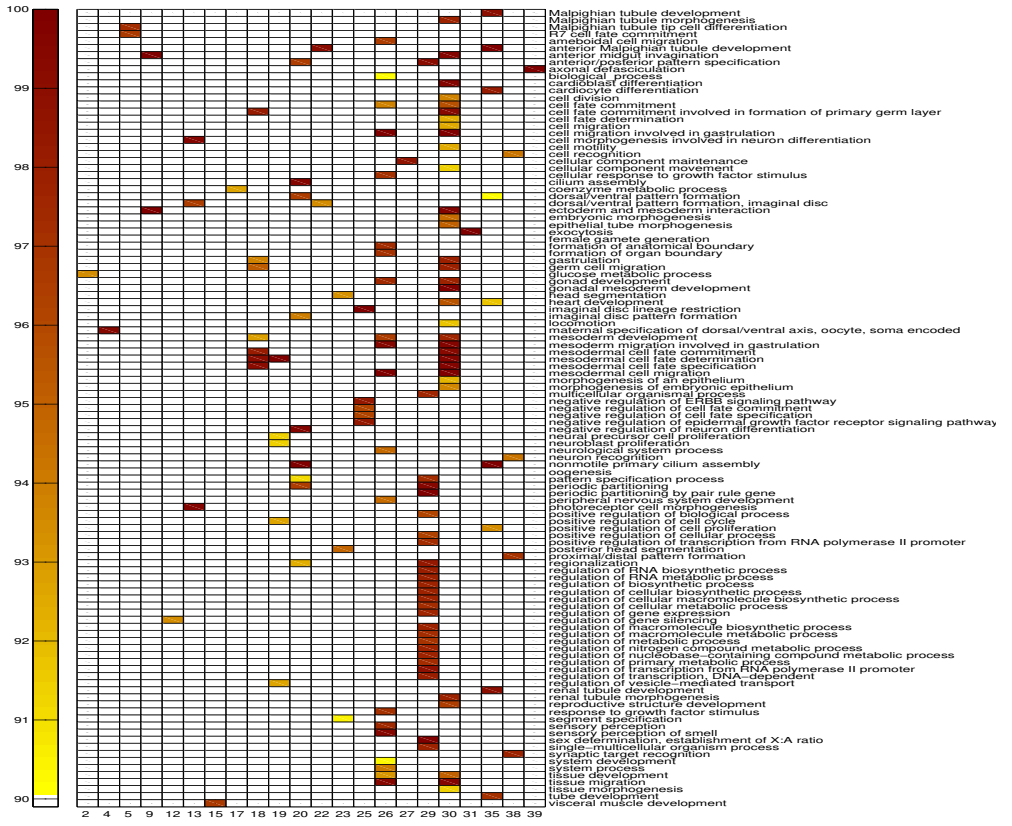


FIG. 13: The clusters with enriched terms and the corresponding terms. I use a p-value threshold of 0.001 to obtain the enriched GO terms (biological process) and then apply the one-sided significance test to retain the enriched terms with $\geq$90% significance. Figure 11 shows the corresponding mesh clusters.

More importantly, my evolutionary co-clustering is able to produce smoothly varying clustering boundaries across time points, while MSSRCC is not able to achieve such effect. Note that, theoretically, the EM algorithm might converge to different optimal points when it is initialized to different values. however, I find in experiments that the clustering results are the same when the EM algorithm is randomly initialized multiple times. This empirical evidence shows that the clustering results are not

sensitive to the initial values.

3.4.4 EVOLUTIONARY CO-CLUSTERING OF GENES AND MESH ELEMENTS

I evaluate the co-clustering of mesh elements and genes and show how they are correlated with developmental events of *Drosophila* embryogenesis. I apply my mesh generation and evolutionary co-clustering methods to the data set of 2675 images of gene expression in stage 4-6. Following [18], I set the number of co-clusters to 39. I compute the enriched Gene Ontology terms (biological process) [60] and evaluate the terms with p-value <0.001. I subsequently apply the one-sided significance test and retain the enriched terms with $\geq 90\%$ significance. Among the 39 clusters, 22 of them have at least one enriched term. The enriched terms in the 22 clusters are shown in Figure 13, and the corresponding mesh clusters are given in Figure 11.

I can see that terms such as gene regulation, pattern formation and embryo development appear in the enriched term list. Note that stage 4-6 is the cellularization and gastrulation stage, and thus the enrichment of these terms makes biological sense. With the fixed stage 4-6, I can map the enriched GO terms back into the mesh cluster visualization (Figure 11). I can see that similar terms are located in spatially adjacent clusters. I also find a subset of well known genes that are activated in the ventral region of the embryo during stage 4-6 containing *twist*, *snail*, *Mes2*, *brinker*, and *tinman*. My findings are consistent with the biological results reported in [61, 62].

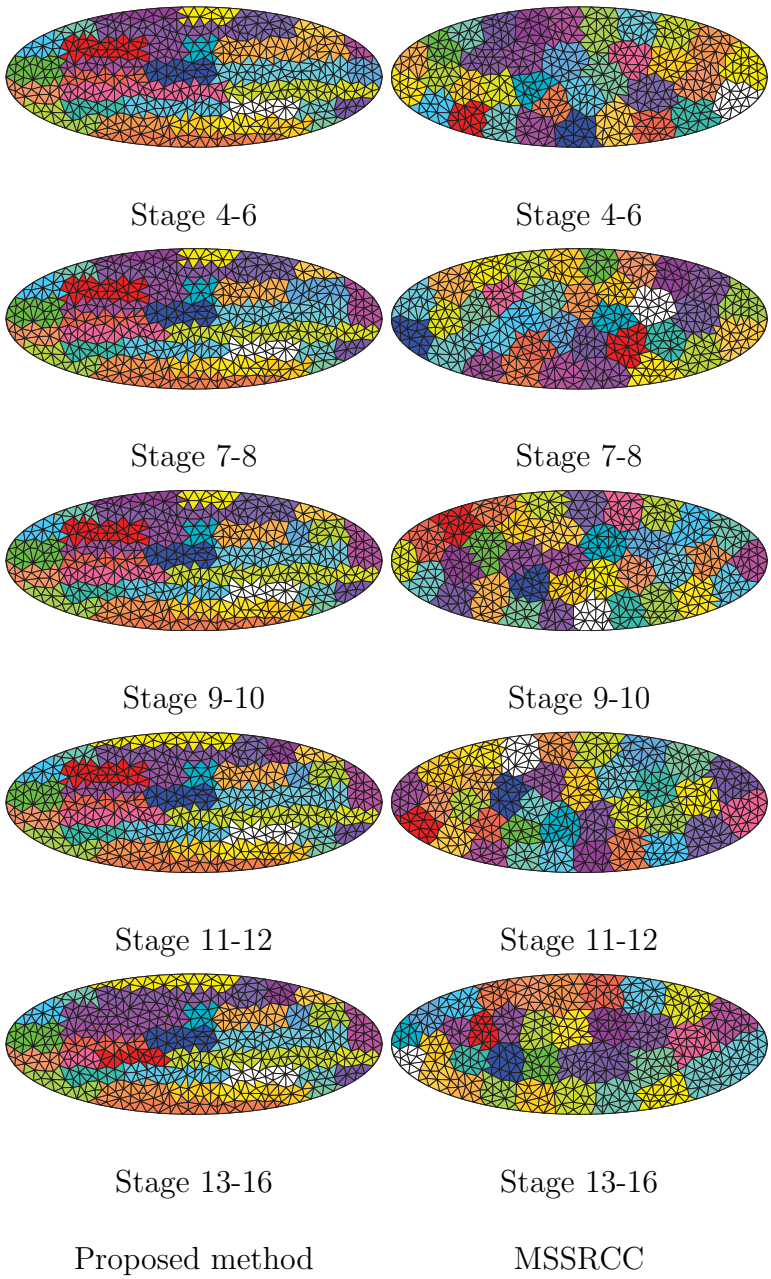


FIG. 14: Clusters of mesh elements when the number of clusters is fixed to 35, and the time points are changed from stage 4-6 to stage 13-16 (top to bottom, stage 4-6, stage 7-8, stage 9-10, stage 11-12, and stage 13-16). The left column shows the results of the proposed method and the right column shows the results of NBIN+RI+MSSRCC+LS.

CHAPTER 4

A PROBABILISTIC LATENT SEMANTIC ANALYSIS MODEL FOR CO-CLUSTERING THE MOUSE BRAIN ATLAS

The mammalian brain contains cells of a large variety of types. The phenotypic properties of cells of different types are largely the results of distinct gene expression patterns. Therefore, it is of critical importance to characterize the expression patterns in the mammalian brain. The Allen Developing Mouse Brain Atlas provides spatiotemporal in situ hybridization gene expression data across multiple stages of mouse brain development, yielding effectively a four-dimensional atlas. It provides a framework to explore spatiotemporal regulation of gene expression during development. I develop a probabilistic co-clustering model to cluster the genes and the brain voxels simultaneously. My model is based on a graph approximation formulation and admits a probabilistic latent variable interpretation. I show that the model parameters can be estimated by an expectation-maximization algorithm. To provide a quantitative comparison with prior methods, I evaluate my model on a set of standard synthetic data sets. Results indicate that my model consistently outperforms prior methods. I apply my method to co-cluster the Allen Developing Mouse Brain Atlas data. Results indicate that my clustering of voxels is more consistent with classical neuroanatomy than prior methods. My analysis also yields sets of genes that are co-expressed in a subset of the brain voxels.

4.1 BACKGROUND

The mammalian brain controls cognition, emotion, and perception and is one of the most complex yet least understood biological systems [27]. It is known that there are at least several hundreds of distinct types of cells in the mammalian brain. These cell types are arranged into complex circuits, which ultimately are responsible for generating brain function. The phenotypic properties of cells of different types are largely the consequences of unique combinations of expressed gene products; therefore, analysis of gene expression patterns provides an informative modality to study developmental gene regulation and cellular diversity. To date, the Allen Brain Atlas (ABA) [28] contains one of the most comprehensive collection of genome-scale, cellular-resolution, three-dimensional (3D) gene expression patterns in the brain of a mouse, a core model for mammalian brain development and behavioral genetics. Analysis of this data set would shed light on the anatomic and genetic organizations of the mammalian brain. Currently, the Allen Brain Atlas provides gene expression data for the developing and adult mouse and human brains [28–30]. Building upon the foundation established by the Allen adult mouse brain atlas [28], the Allen Developing Mouse Brain Atlas provides spatiotemporal *in situ* hybridization (ISH) gene expression data across multiple stages of mouse brain development [30], yielding effectively a four-dimensional brain atlas. It provides a framework to explore temporal and spatial regulation of gene expression during development.To establish a common coordinate framework for analyzing the ISH data, the ISH image series are aligned to the Allen Developing Mouse Brain Reference Atlas (the Reference Atlas). The

Reference Atlas was created based on the "prosomeric model" [31], which proposes that the neural tube is divided into grid-like pattern of longitudinal and transverse regions. These divisions form the primary histogenetic domains upon which further elaboration of expression are developed independently [32]. It is, therefore, of fundamental importance to study the gene regulations that lead to the formation of these domains. In this chapter, I aim at investigating the genes that are co-expressed at each of the primary longitudinal and transverse domains. The data for each developmental stage is organized as a data matrix in which one dimension corresponds to the genes, and the other dimension corresponds to the brain voxels. I apply the proposed co-clustering model to cluster the genes and the voxel simultaneously, thereby elucidating the genetic and anatomic interactions governing mouse brain development.To provide a quantitative comparison with prior methods, I first evaluate the co-clustering method on a set of standard synthetic data sets. I compare my model with seven prior co-clustering methods on the synthetic data sets. Experimental results show that my model consistently outperforms prior methods. In addition, my results demonstrate that the performance of my model does not degrade as the noise level increases, suggesting that the proposed method is robust to noise in the data. I then apply my method to co-cluster the Allen Developing Mouse Brain Atlas data. To provide a quantitative assessment, I compare the voxel clusters with the classical neuroanatomy reflected in the Allen Developing Mouse Reference Atlas. Experimental results show that the voxel clusters produced by my method are more consistent with the longitudinal and transverse domains in neuroanatomy.

4.2 A CO-CLUSTERING FRAMEWORK

In this section, I describe the co-clustering method based on graph approximation. I then show the relationship with symmetric PLSA. I demonstrate in later section that my method consistently outperforms prior methods on both the synthetic and the Allen Developing Mouse Brain Atlas data.

4.2.1 A GRAPH APPROXIMATION FORMULATION

The Allen Developing Mouse Brain data at a particular developmental age can be organized as a matrix in which one dimension corresponds to the genes and the other dimension corresponds to the brain voxels. I encode this data matrix as a bipartite graph in which the two sets of vertices correspond to the genes and the brain voxels, respectively. The expression level of genes at brain voxels are encoded into the weights of edges connecting the corresponding genes and voxels in the bipartite graph. I propose to approximate this bipartite graph using a tripartite graph. This gives rise to a formalism to cluster the genes and the voxels simultaneously.

Suppose that I am given a set of m genes $g_1, g_2, \cdots, g_m$ and a set of n brain voxels $v_1, v_2, \cdots, v_n$. The expression level of these genes on the given voxels can be captured by the matrix $W \in \mathbb{R}_+^{m \times n}$, where w_{ij} denotes the expression level of the ith gene at the jth voxel, and $\mathbb{R}_+^{m \times n}$ denotes the set of $m \times n$ matrices with nonnegative elements. This data set can be represented as a bipartite graph in which one set of vertices represent the genes, and the other set of vertices correspond to the voxels. In the following, I use the vertices and the genes or voxels that they represent exchangeably

to simplify the description. The edge connecting the ith gene with the jth voxel carries a weight of w_{ij}. This representation is graphically illustrated in Figure 15 (a).

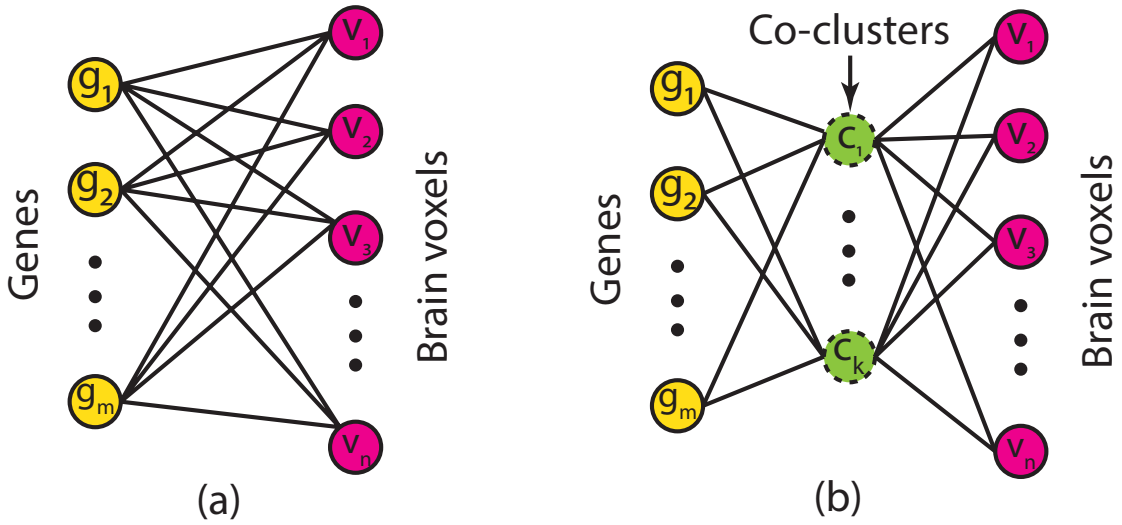


FIG. 15: Illustration of the probabilistic co-clustering model. (a) The data matrix at a particular developmental age is represented as a bipartite graph in which the two sets of vertices correspond to the genes and the brain voxels, respectively. The expression level of genes at brain voxels are encoded into the weights of edges connecting the corresponding genes and voxels in the bipartite graph. (b) The gene-voxel co-cluster structure can be captured by a tripartite graph in which the vertices with dashed edges correspond to co-clusters. I use this tripartite graph to approximate the bipartite graph in (a), thereby leading to a co-clustering formulation.

I propose to construct a tripartite graph as in Figure 15 (b) to approximate the bipartite graph. In this tripartite graph a new set of vertices $c_1, c_2, \cdots, c_k$ are introduced to represent the k co-clusters. The edges in this tripartite graph consist of two disjoint subsets. The first subset consists of edges connecting genes with co-clusters,

and the second subset consists of edges connecting voxels with co-clusters. Accordingly, the edge weights of this tripartite graph can be captured by two matrices. Let $A \in \mathbb{R}^{m\times k}$ encodes the weights of edges connecting genes with co-clusters in which a_{iq} denotes the weight of edge connecting the ith gene with the qth co-cluster, and $B \in \mathbb{R}^{n\times k}$ encodes the weights of edges connecting voxels with co-clusters in which b_{jq} denotes the weight of edge connecting the jth voxel with the qth co-cluster. Note that, similar to all methods considered in this chapter, the number of nodes in the co-cluster layer (i.e., the number of co-clusters) in my model needs to be specified by the user.

To construct gene and voxel co-clusters, I propose to approximate the relationship between genes and voxels in the bipartite graph using the constructed tripartite graph. It is clear from the tripartite graph that there are no direct links between the genes and voxels, and they can only be connected via the co-cluster vertices. Hence, the expression of the ith gene at the jth voxel can be approximated as [63]

$$w_{ij} \approx \sum_{q=1}^{k} \frac{a_{iq} b_{jq}}{\sigma_q}, \tag{16}$$

where $\sigma_q = \sum_{i=1}^{m} a_{iq} + \sum_{j=1}^{n} b_{jq}$ denotes the degree of vertex c_q. This approximation can be concisely expressed in matrix form as

$$W \approx A\Sigma B^T, \tag{17}$$

where $\Sigma \in \mathbb{R}_{+}^{k\times k}$ is a diagonal matrix with $(\Sigma)_{qq} = \frac{1}{\sigma_q}$.

A natural way to compute A, B, and Σ is to minimize the approximation error with respect to a loss function $\ell(\cdot,\cdot)$ as $\min_{A,B,\Sigma} \ell(W, A\Sigma B^T)$. Two commonly used

loss functions are the sum-of-squares loss and the divergence loss. In this paper, I consider the divergence loss as it leads to a probabilistic interpretation [64]. This gives rise to the following objective function:

$$\ell(W, A\Sigma B^T) = \sum_{i=1}^{m}\sum_{j=1}^{n}(w_{ij}\log\frac{w_{ij}}{(A\Sigma B^T)_{ij}} - w_{ij} + (A\Sigma B^T)_{ij}).$$

Note that the divergence loss function is not symmetric, and it achieves the minimum value of zero only when $W = A\Sigma B^T$.

4.2.2 RELATIONSHIP WITH PLSA

I show that my graph approximation formulation can be interpreted using random walks [63]. This interpretation establishes an equivalence relationship between my formulation and a variant of PLSA [65], thereby allowing us to use the expectation-maximization(EM) algorithm for PLSA to estimate the co-clustering parameters.

Without loss of generality [66], let W be normalized so that $\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij} = 1$. Then w_{ij} denotes the stationary probability of direct transitions between g_i and v_j in the bipartite graph. In the tripartite graph, the random walk needs to follow a two-edge path for making a transition from g_i to v_j. This leads to the following transition probability:

$$\begin{aligned} p(g_i, v_j) &= p(g_i)p(v_j|g_i) \\ &= p(g_i)\sum_{q=1}^{k} p(c_q|g_i)p(v_j|c_q) \\ &= \sum_{q=1}^{k}\frac{p(c_q, g_i)p(v_j, c_q)}{p(c_q)} \\ &= \sum_{q=1}^{k}\frac{p(g_i, c_q)p(v_j, c_q)}{\sigma_q}. \end{aligned} \tag{18}$$

Alternatively, the transition probability can be characterized in a symmetric manner, since the genes and the voxels are conditionally independent given the co-clusters.

$$p(g_i, v_j) = \sum_{q=1}^{k} p(g_i|c_q)p(v_j|c_q)p(c_q) \quad (19)$$

$$= \sum_{q=1}^{k} \frac{p(g_i, c_q)p(v_j, c_q)}{\sigma_q}. \quad (20)$$

By comparing Eqs. (16), (18) and (20), it is clear that a_{iq} can be interpreted as a quantity characterizing the transition probability from g_i to c_q, and b_{jq} can be interpreted as quantifying the transition probability from v_j to c_q. Interestingly, it can be verified that the formulation in (18) and (20) are equivalent to the asymmetric and symmetric variants, respectively, of PLSA [65]. This allows us to use the EM algorithm for PLSA to estimate the co-clustering parameters.

The EM algorithm consists of two steps that are alternated until convergence. For clarity, I use variables with hat to denote the values obtained from the previous iteration in the following. In the E-step, I compute the expectation of the latent variable given the parameter values from the previous iteration. This can be achieved by applying the Bayes' Theorem to Eq. (19), giving rise to the following result:

$$p(c_q|g_i, v_j) = \frac{p(c_q)p(g_i|c_q)p(v_j|c_q)}{\sum_{r=1}^{k} p(c_r)p(g_i|c_r)p(v_j|c_r)} = \frac{\hat{a}_{iq}\hat{b}_{jq}\hat{\sigma}_q}{(\hat{A}\hat{\Sigma}\hat{B}^T)_{ij}}.$$

In the M-step, I maximize the expected complete data log likelihood. In addition, the following constraints need to be enforced for a probabilistic interpretation:

$$\sum_{q=1}^{k} p(c_q) = 1, \quad \sum_{i=1}^{m} p(g_i|c_q) = 1, \quad \sum_{j=1}^{n} p(v_j|c_q) = 1. \quad (21)$$

It can be verified that this optimization can be achieved by applying the following update rules:

$$p(c_q) \propto \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{r=1}^{k} \frac{w_{ij} p(c_q) p(g_i|c_q) p(v_j|c_q)}{p(c_r) p(g_i|c_r) p(v_j|c_r)}, \tag{22}$$

$$p(g_i|c_q) \propto \sum_{j=1}^{n} \sum_{r=1}^{k} \frac{w_{ij} p(c_q) p(g_i|c_q) p(v_j|c_q)}{p(c_r) p(g_i|c_r) p(v_j|c_r)}, \tag{23}$$

$$p(v_j|c_q) \propto \sum_{i=1}^{m} \sum_{r=1}^{k} \frac{w_{ij} p(c_q) p(g_i|c_q) p(v_j|c_q)}{p(c_r) p(g_i|c_r) p(v_j|c_r)}. \tag{24}$$

and then normalizing the results so that the constraints in Eq. (21) are satisfied.

The E-step and the M-step are repeated until a locally optimal solution is obtained. It can be shown that this procedure monotonically increases the log likelihood until a locally optimal solution is reached. Then $p(g_i|c_q)$ and $p(v_j|c_q)$ can be considered as soft clustering assignments for the genes and voxels, respectively. This naturally leads to a soft co-clustering of genes and voxels. In addition, hard co-clustering results can be obtained by assigning each gene or voxel to the co-cluster with the largest probability.

4.3 RELATED WORK

Simultaneous row and column clustering for identifying block structures from matrix data has been initially studied in [67]. Recent surge of interests in co-clustering is motivated by biological applications, which aim at identifying subset of genes co-expressed in a subset of samples from microarray gene expression data [57, 68]. Co-clustering has also been applied in many other applications, including simultaneous clustering of words and documents [38, 69], authors and conference [4], etc. Early work on co-clustering focuses on defining an error measure and then identifying blocks that minimize this measure using heuristic search algorithms [57, 67].

These early work has recently been reformulated using matrix and optimization techniques [70,71]. Following the spectral clustering formalism, it has been shown recently that co-clustering is closely related to the singular value decomposition (SVD) of the data matrix [72]. It is shown in [73] that sparsity-inducing regularization can be employed to compute sparse singular vectors, which in turn can be used to form co-clusters.

The proposed probabilistic model is related to the spectral co-clustering formulation [38, 39] in which a bipartite graph is used to encode the word-document matrix. In these studies, co-clustering is formulated as a bipartite graph cut problem, and the data are projected onto the left and right singular vector spaces before they are concatenated and clustered to identify row and column co-clusters. A major difference between my model and the spectral co-clustering formulation is that the cluster assignment matrix in spectral method is computed from the eigen-decomposition of graph Laplacian matrix. As a result, the spectral method can produce negative cluster assignments that are hard to interpret [64, 74]. In contrast, the parameters estimated by my model are nonnegative and admit a probabilistic interpretation.

My work is also connected to nonnegative matrix factorization [74] and probabilistic latent semantic analysis [64]. In [75], a co-clustering model is developed for analyzing the adult mouse brain ISH data. The proposed model is, however, computationally very expensive and is applicable only to small-scale data sets. In [76], voxels from the adult mouse brain expression data are clustered, and the results are compared to classical neuroanatomy. The motivation of my work is different from

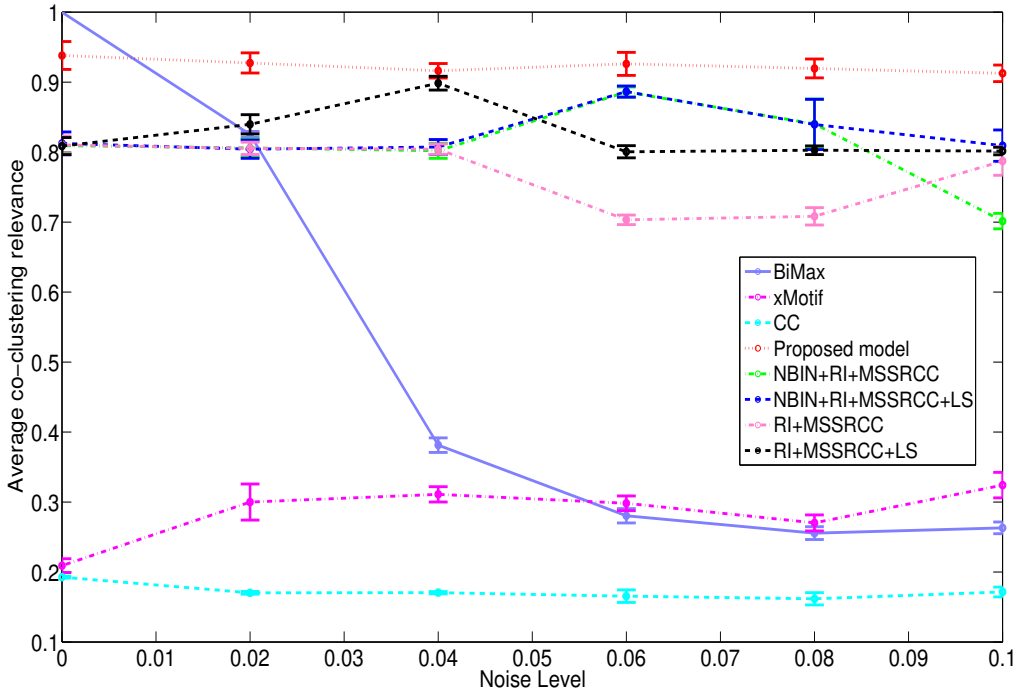


FIG. 16: Co-clustering performance of eight methods on the synthetic data sets. At each noise level (horizontal axis), the results are the average performance across ten data matrices. The performance is measured using average co-clustering relevance as in [58].

that of the work by [76], since their goal was to only cluster the voxels, and my goal is to identify gene and voxel co-clusters.

4.4 EXPERIMENTAL EVALUATION

4.4.1 EXPERIMENTAL EVALUATION ON SYNTHETIC DATA

In [77], five selected co-clustering methods are evaluated on a set of synthetic gene expression data sets. In the synthetic data, co-clusters represent transcription

modules, which are defined by a set of genes regulated by common transcription factors and a set of conditions under which these transcription factors are active [78]. In [77], 10 non-overlapping transcription modules, each extending over 10 genes and 5 conditions, are used to generate 10 co-clusters. To study the robustness of the co-clusters methods, noise is introduced into the data by adding random values drawn from a Gaussian distribution to each element of the data matrix. The noise level is controlled by the standard deviation of the Gaussian distribution, and various noise levels have been considered in [77]. In [58], a new co-clustering method, known as the "minimum sum-squared residue co-clustering (MSSRCC)", have been compared with the methods in [77], and results indicate that MSSRCC achieves better performance.

To provide a quantitative evaluation of my co-clustering model, I compare my approach with the methods in [77] and [58] on the synthetic data sets. Specifically, I choose three methods from [77]. These are (1) "BiMax" proposed in [77], (2) "x-Motif" developed in [79], and (3) "CC" described in [57]. I also compare my method with four different variants of the minimum sum-squared residue co-clustering (MSSRCC) method. As in [58], I consider "RI+MSSRCC+LS", "RI+MSSRCC", "NBIN+RI+MSSRCC", and "NBIN+RI+MSSRCC+LS", where "RI" corresponds to random initialization; NBIN denotes the binormalization method in [59]; LS denotes local search.

I briefly describe these methods and their parameters in the following. I also provide references to the original work, where more details can be found.

- "BiMax" [77] is an efficient divide-and-conquer implementation of the binary

inclusion-maximal biclustering algorithm (BiMax). It requires the number of co-clusters as a user-specified parameter.

- "xMotif" [79] is an iterative search method that computes co-clusters containing approximately constant expression values. It requires two user-specified parameters α and β. α specifies the minimum fraction of samples in each co-cluster, and β specifies the maximum fraction of genes not in a co-cluster that can be conserved in samples in the current co-cluster. These two parameters can be used in combination to control the number of co-clusters.

- "CC" denotes the node-deletion algorithm in [57] to compute blocks in expression data by minimizing the mean squared residue scores. It requires the maximum acceptable mean squared residue score $\delta \geq 0$ as an input parameter. The number of co-clusters can be controlled by adjusting the δ value.

- "RI+MSSRCC+LS" denotes the heuristic algorithm to compute the minimum sum-squared residue co-clustering (MSSRCC) [58] with random initialization and local search. This method requires the number of co-clusters to be specified by the user.

- "RI+MSSRCC" denotes the heuristic algorithm to compute the minimum sum-squared residue co-clustering (MSSRCC) [58] with random initialization. This method requires the number of co-clusters to be specified by the user.

- "NBIN+RI+MSSRCC" denotes the heuristic algorithm to compute the minimum sum-squared residue co-clustering (MSSRCC) [58] with random initialization and data binormalization [59]. This method requires the number of co-clusters to be specified by the user.

- "NBIN+RI+MSSRCC+LS" denotes the heuristic algorithm to compute the minimum sum-squared residue co-clustering (MSSRCC) [58] with random initialization, local search, and data binormalization [59]. This method requires the number of co-clusters to be specified by the user.

All these methods either require the number of co-clusters to be directly specified by the user, or require other parameters that are related to the number of resulting co-clusters. In the experiments, I have tuned the parameters so that the number of resulting co-clusters in all methods is equal to the number of optimal co-clusters.

Following [77], I use the average co-cluster relevance to measure the co-cluster quality. This measure is defined in Definition 2 in [77] and reflects the extent to which the generated co-clusters represent true co-clusters in the gene dimension. It takes a maximum value of 1 when the true co-clusters are perfectly recovered. The noise level is varied from 0 to 0.1, and I report the average performance and error bars over 10 input matrices at each noise level in Figure 16. I can observe that my probabilistic model outperforms other methods consistently across most noise levels. More importantly, the results show that the performance of my model does not degrade with increased level of noise. Consistent with the results in [58], my results also show that variants of MSSRCC outperform most methods evaluated

in [77]. This set of experiments demonstrate that my model achieves consistently higher performance than prior methods, and that my method is robust to noise in the data.

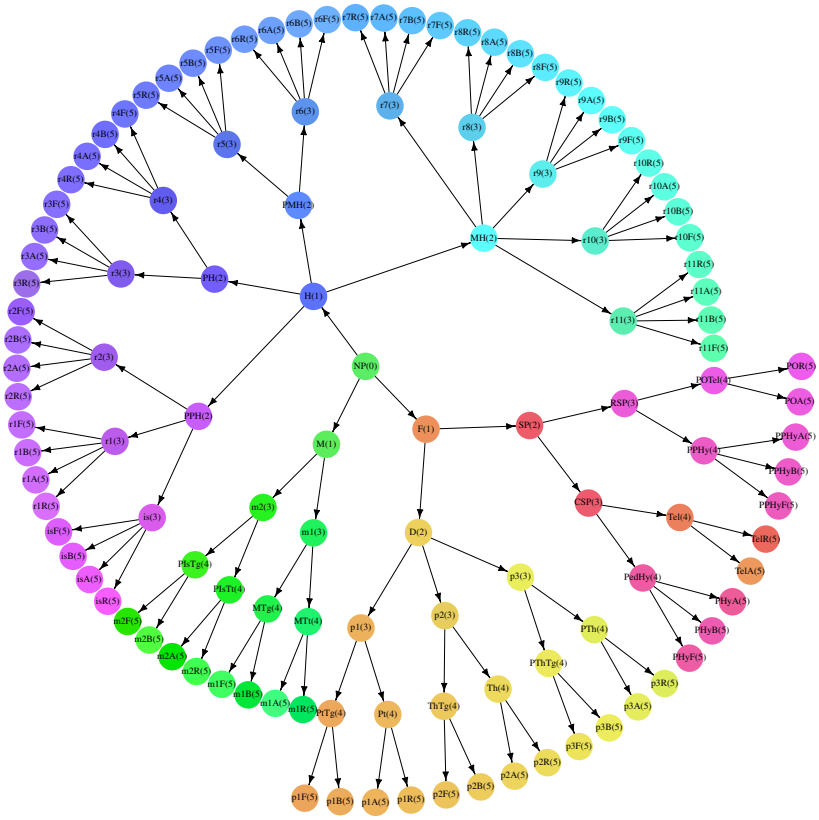


FIG. 17: The Allen Developing Mouse Brain Reference Atlas ontology hierarchy through level 5.

4.4.2 EXPERIMENTAL EVALUATION ON ALLEN DEVELOPING MOUSE BRAIN ATLAS

The Allen Developing Mouse Brain Atlas (the Atlas) contains spatiotemporal *in situ* hybridization (ISH) gene expression data across multiple stages of mouse brain development [30]. The primary data consist of three-dimensional (3D), cellular resolution ISH expression patterns of approximately 2000 genes in sagittal plane

across four embryonic (E11.5, E13.5, E15.5, and E18.5) and three early postnatal ages (P4, P14, and P28). To provide a novel neuroanatomical framework, the Allen Developing Mouse Brain Reference Atlas (the Reference Atlas) was developed to create 3D models of the mouse brain (Figure 18). The Reference Atlas is based upon a systematic developmental ontology that is organized in a 13-level hierarchy. To establish a common coordinate framework for analyzing the ISH data, the ISH image series are aligned to the Reference Atlas in 3D space. A regular grid is then applied to the aligned ISH images to generate voxel-level expression summaries. My analysis in this work is based on the grid data.

The Reference Atlas was created based on the "prosomeric model" [31]. This model proposes that the neural tube is constructed from serial transversal divisions sitting across the primary longitudinal zones. Four longitudinal zones, known as the floor plate, basal plate, alar plate and roof plate, are generated by the dorsoventral patterning signals. Transverse molecular boundaries subdivide it into a set of anteroposterior segments. Specifically, the prosencephalon consists of 3 prosomeres (p1-p3) in the diencephalon, and a bipartition of the secondary prosencephalon. The rhombencephalon is subdivided into 12 segments, termed rhombomeres (r1-r11 with isthmus counted as r0). The mesencephalon divides into m1 and m2 mesomere. This grid-like pattern of longitudinal and transverse regions form the primary histogenetic domains upon which further elaboration of expression are developed independently [32, 80]. It is, therefore, of fundamental importance to study the gene regulations that lead to the formation of these domains.

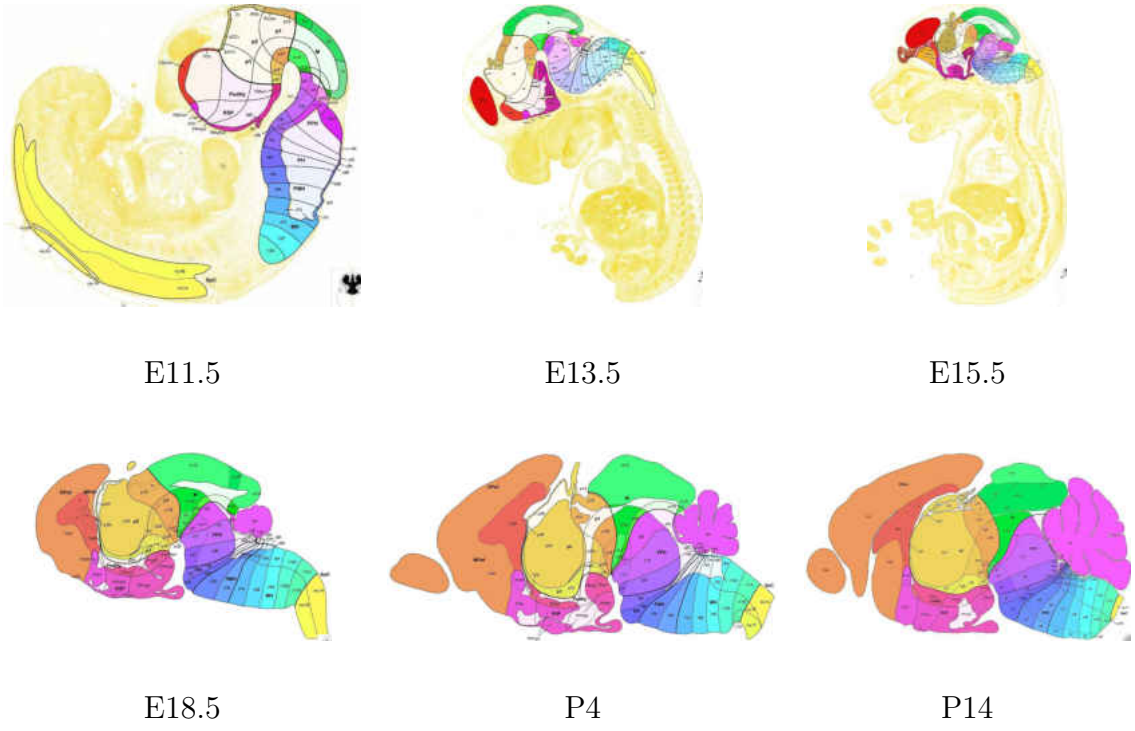


FIG. 18: Sample sections of the Allen Developing Mouse Brain Reference Atlas at six stages of mouse brain development in the sagittal plane. The reference atlas for stage P28 is not available from the Allen Brain Atlas data portal. In the Reference Atlas, the colors of brain structures are selected such that ontologically related structures are given visually related colors by allocating segments of the color wheel to major subdivisions of the brain.

To provide a visualization of the Allen Developing Mouse Brain Reference Atlas ontology, I show the hierarchy from level 0 to level 5 in Figure 17. In this figure, each ontological term corresponds to a node in the hierarchy, labeled by the abbreviation followed by the level number inside a parenthesis. The nodes are color-coded as in the original atlas in Figure 3. The transverse segments lie at level 3, and they are combined with the longitudinal zones at level 5 to generate the grid-like pattern. I

up-propagate the voxel annotations to levels 3 and 5, respectively, in my experiments in order to study the gene expressions in the grid-like longitudinal and transverse domains. In this chapter, I aim at investigating the genes that are co-expressed at each

TABLE 2: Statistics of the developing mouse brain data.

	E11.5	E13.5	E15.5	E18.5	P4	P14	P28
# of genes	1948	1948	1930	1946	1918	1906	1944
# of voxels	7122	13194	12148	12045	21845	24325	28023
# of Level 3 structures	20	20	20	20	20	19	20
# of Level 5 structures	82	77	76	65	64	71	74

of the primary longitudinal and transverse domains. To this end, I up-propagate the voxel annotations to levels corresponding to the longitudinal and transverse domains. In the Reference Atlas, the transverse segments lie at level 3, and they are combined with the longitudinal zones to form the grid-like pattern at level 5. I thus up-propagate the annotations to level 3 and 5, respectively, for each brain voxel. I retrieved the ISH expression energy grid files for seven developmental stages from the Allen Brain Atlas data portal and treat the energy values as expression levels. The data for each developmental stage is organized as a data matrix, where one dimension corresponds to the genes, and the other dimension corresponds to the brain voxels. Each voxel is annotated with a level 3 structure and a level 5 structure. Statistics of the data are given in Table 2. I consider the voxel annotation labels as ground truth to evaluate the performance of co-clustering methods, since it has been shown that

brain voxels in the same structure usually form a cluster [76].

I compare the clustering of the brain voxels with the up-propagated level 3 and 5 structure annotations using a variety of measures, including the purity, normalized mutual information (NMI), and the Rand index (RI). These measures are commonly used as external criteria of evaluating clustering quality [81]. In addition, I use the S-index introduced in [76] for comparing the voxel clustering results with classical neuroanatomy. The numbers of level 3 and 5 structures that are actually present in each data set might be different, since not all structures are annotated at all developmental stages. I show the number of level 3 and 5 structures in Table 2 and set the number of co-clusters to be the same as the number of structures at the corresponding level, since my primary goal is to identify the longitudinal and transverse domains. Note that the purity, Rand index, and S index are dependent on the number of clusters, so clustering results with different numbers of clusters cannot be compared using these measures. The NMI is independent of the number of clusters. So, this measure can be used to compare results with different numbers of clusters.

I compare my model with the four variants of MSSRCC method used in my synthetic study, since MSSRCC achieved consistently better performance than other co-clustering methods in [77]. MSSRCC requires the number of co-clusters as an input parameter, so I set the number of co-clusters in MSSRCC and in my method to be the number of brain structures at the corresponding level in all experiments. I summarize the voxel co-clustering performance using level 3 and level 5 structure

annotations as ground truth in Tables 3 and 4, respectively. I can observe that my model consistently outperforms variants of the MSSRCC method in almost all cases across various performance measures. Specifically, for the results in Table 3, my method outperforms all variants of MSSRCC in 23 out of the 28 cases (data sets and performance measure combinations). Similarly, my method outperforms the four variants of MSSRCC in 21 out of the 28 cases in Table 4. This demonstrates that the co-clustering results produced by my method are more consistent with the primary longitudinal and transverse domains reflected in the Allen Developing Mouse Brain Reference Atlas than those generated by variants of the MSSRCC method.

I can also observe that, in general, the co-clustering performance is higher for data sets corresponding to late stages of development. This result is consistent with the general principle of development in which gene regulatory mechanisms act sequentially to form more and more refined expression patterns. Thus, expression patterns of voxels in the same structure become more and more similar, while the those in different structures diverge continuously as development progresses [82]. Therefore, voxels in the same structure tend to form increasingly clear clusters that can be easily identified by computational methods.

My results also show that the clustering performance increases dramatically from stages E11.5 to E13.5. This is consistent with the observation that there are major developmental events happened during this time interval [32, 82]. It has long been hypothesized that molecular mechanisms for regionalization of the neural plate act well before the actual structures can be visually identified [27]. But this hypothesis

remain untested due to the lack of systematic data and analysis. My global analysis of the developing mouse brain data suggest that the genetic signals for regionalization at E11.5 are still weak, and they increase dramatically at stage E13.5. My results are consistent with the fact that, by E14.5, most of the varieties of neurons have been generated and have migrated into the mantel layer. I next investigate how each gene is associated with multiple co-clusters probabilistically, since each gene might be expressed in multiple regions. To this end, I collect the region-level gene expression data and obtain a ranked list of regions for each gene according to the expression levels. To compare these with the soft co-clustering results, I label each co-cluster with the annotation of the majority voxels in that co-cluster. For each gene, I then rank the co-clusters using the probabilities with which this gene is associated with each co-cluster. I observe that these two lists contain significant numbers of overlapping regions for many genes. Table 5 reports the top 10 regions for 3 sample genes. This shows that my soft co-clustering method is able to associate genes with multiple voxel clusters probabilistically to reflect the fact that genes can be expressed in multiple regions.

TABLE 3: Experimental results on the Allen Developing Mouse Brain Atlas data when the voxel annotations are up-propagated to level 3. In each case, the method with the highest performance is highlighted in bold face. The level 3 Reference Atlas ontological terms are shown in Figure 17. See the caption of Figure 16 for details.

Measures	Methods	E11.5	E13.5	E15.5	E18.5	P4	P14	P28
	Proposed model	0.2928	**0.5934**	0.5529	**0.5709**	**0.5652**	**0.6941**	0.7091
	RI+MSSRCC+LS	0.2613	0.5012	**0.5679**	0.5532	0.5571	0.6779	**0.7121**
Purity	RI+MSSRCC	**0.3018**	0.4976	0.5324	0.5601	0.5438	0.6802	0.7003
	NBIN+RI+MSSRCC	0.2916	0.4829	0.5078	0.5479	0.5501	0.6793	0.7021
	NBIN+RI+MSSRCC+LS	0.2708	0.5364	0.5431	0.5328	0.5512	0.6778	0.6918
	Proposed model	0.1349	**0.41**	**0.3594**	**0.3233**	**0.3671**	0.3829	0.4036
	RI+MSSRCC+LS	0.1027	0.3726	0.3229	0.3112	0.3331	0.3771	**0.4121**
NMI	RI+MSSRCC	0.1005	0.3658	0.3478	0.3097	0.3498	**0.3913**	0.4005
	NBIN+RI+MSSRCC	**0.1479**	0.3871	0.3196	0.3129	0.3291	0.3816	0.3783
	NBIN+RI+MSSRCC+LS	0.1258	0.3596	0.3005	0.3008	0.3479	0.3662	0.3996
	Proposed model	0.3097	**0.6291**	**0.5614**	**0.5805**	**0.5724**	**0.7029**	**0.7128**
	RI+MSSRCC+LS	0.2694	0.6001	0.5005	0.5613	0.5557	0.7004	0.7091
Rand index	RI+MSSRCC	0.2371	0.5194	0.5478	0.5129	0.5491	0.6778	0.6847
	NBIN+RI+MSSRCC	**0.3215**	0.4947	0.5078	0.5479	0.5501	0.6793	0.7021
	NBIN+RI+MSSRCC+LS	0.2708	0.5364	0.5431	0.5328	0.5512	0.6778	0.6918
	Proposed model	**0.5219**	**0.7843**	**0.6825**	**0.7142**	**0.7093**	**0.8428**	**0.8637**
	RI+MSSRCC+LS	0.4218	0.6591	0.6049	0.6876	0.5942	0.7495	0.7593
S-Index	RI+MSSRCC	0.3682	0.6432	0.6241	0.6639	0.5387	0.7816	0.8104
	NBIN+RI+MSSRCC	0.3981	0.6341	0.6518	0.6902	0.6023	0.8091	0.8346
	NBIN+RI+MSSRCC+LS	0.4863	0.7016	0.6673	0.6963	0.6593	0.8013	0.8549

TABLE 4: Experimental results on the Allen Developing Mouse Brain Atlas data when the voxel annotations are up-propagated to level 5.

Measures	Methods	E11.5	E13.5	E15.5	E18.5	P4	P14	P28
Purity	Proposed model	**0.3278**	**0.5589**	**0.6482**	**0.6017**	0.6129	**0.7268**	**0.7418**
	RI+MSSRCC+LS	0.2845	0.5354	0.6235	0.5435	0.6035	0.6834	0.7246
	RI+MSSRCC	0.3021	0.4983	0.5935	0.5567	0.5927	0.6946	0.7145
	NBIN+RI+MSSRCC	0.3104	0.5436	0.6356	0.5835	**0.6164**	0.7037	0.7326
	NBIN+RI+MSSRCC+LS	0.2925	0.5146	0.6424	0.5934	0.6157	0.7167	0.7298
NMI	Proposed model	0.1475	**0.4578**	**0.5063**	**0.4168**	0.4085	**0.4276**	**0.4468**
	RI+MSSRCC+LS	0.1245	0.4024	0.3856	0.3456	0.3567	0.3985	0.4145
	RI+MSSRCC	0.1534	0.4235	0.3982	0.3013	0.3732	0.3725	0.4045
	NBIN+RI+MSSRCC	0.1467	0.4174	0.3698	0.3982	0.3987	0.3945	0.4325
	NBIN+RI+MSSRCC+LS	**0.1678**	0.4156	0.4015	0.3714	**0.4315**	0.3982	0.4345
Rand index	Proposed model	0.2789	0.6034	**0.6143**	**0.6496**	0.6178	**0.7427**	**0.7268**
	RI+MSSRCC+LS	0.3023	0.5987	0.5896	0.5896	0.6015	0.7246	0.6946
	RI+MSSRCC	0.2987	**0.6135**	0.5438	0.5903	0.6143	0.7167	0.6836
	NBIN+RI+MSSRCC	**0.3158**	0.5897	0.5863	0.6086	0.5996	0.7357	0.7032
	NBIN+RI+MSSRCC+LS	0.3087	0.5963	0.6047	0.6346	**0.6246**	0.7305	0.7156
S-Index	Proposed model	**0.4125**	**0.7257**	**0.6784**	**0.7017**	**0.7158**	**0.8245**	0.8045
	RI+MSSRCC+LS	0.3856	0.6935	0.6356	0.6674	0.5966	0.8034	0.7645
	RI+MSSRCC	0.3773	0.6853	0.6259	0.6547	0.6034	0.7845	0.7945
	NBIN+RI+MSSRCC	0.4025	0.7034	0.6596	0.6678	0.6534	0.7945	0.8046
	NBIN+RI+MSSRCC+LS	0.4096	0.7135	0.6674	0.6934	0.6854	0.8036	**0.8236**

TABLE 5: Ranked region lists of gene expression and co-cluster associations for three sample genes. Columns headed by "Expression" show the regions ranked by gene expression, and those headed by "Co-cluster" show the regions ranked by soft co-clustering probabilities.

Egr2		Gabrg1		Meis2	
Expression	Co-cluster	Expression	Co-cluster	Expression	Co-cluster
r2A	my1A	p2A	TelA	r3B	r2A
r4A	r4A	m1A	p1A	r3A	r4B
r5A	r2A	p1A	my1A	r4B	r8A
r3A	TelA	r4R	TelA	r4A	r3A
r6A	r6A	TelA	p3A	r4F	r4A
r7A	r1A	r4A	r5A	r2A	r6A
r1A	r9B	p2R	TelA	r3F	TelA
r8A	m1A	r5R	p2A	r6A	r2B
TelA	r1A	r5A	r4A	r7A	r5A
m1A	r4A	r3A	m1A	r5A	my1A

CHAPTER 5

DEEP MODEL BASED TRANSFER AND MULTI-TASK LEARNING FOR BIOLOGICAL IMAGE ANALYSIS

A central theme in learning from image data is to develop appropriate image representations for the specific task at hand. Traditional methods used handcrafted local features combined with high-level image representations to generate image-level representations. Thus, a practical challenge is to determine what features are appropriate for specific tasks. For example, in the study of gene expression patterns in *Drosophila melanogaster*, texture features based on wavelets were particularly effective for determining the developmental stages from *in situ* hybridization (ISH) images. Such image representation is however not suitable for controlled vocabulary (CV) term annotation because each CV term is often associated with only a part of an image. Here, I develop problem-independent feature extraction methods to generate hierarchical representations for ISH images. My approach is based on the deep convolutional neural networks (CNNs) that can act on image pixels directly. To make the extracted features generic, the models are trained using a natural image set with millions of labeled examples. These models are transferred to the ISH image domain and used directly as feature extractors to compute image representations. Furthermore, I employ multi-task learning method to fine-tune the pre-trained models with labeled ISH images, and also extract features from the fine-tuned models.

Experimental results show that feature representations computed by deep models based on transfer and multi-task learning significantly outperform other methods for annotating gene expression patterns at different stage ranges. I also demonstrate that the intermediate layers of deep models produce the best gene expression pattern representations.

5.1 BACKGROUND

A general consensus in image-related research is that different recognition and learning tasks may require different image representations. Thus, a central challenge in learning from image data is to develop appropriate representations for the specific task at hand. Traditionally, a common practice is to hand-tune features for specific tasks, which is time-consuming and requires substantial domain knowledge. For example, in the study of gene expression patterns in *Drosophila melanogaster*, texture features based on wavelets, such as Gabor filters, were particularly effective for determining the developmental stages from *in situ* hybridization (ISH) images [83]. Such image representation, often referred to as "global visual features", is not suitable for controlled vocabulary (CV) term annotation because each CV term is often associated with only a part of an image, thereby requiring an image representation of local visual features [24,84]. Current state-of-the-art systems for CV term annotation first extracted local patches of an image and computed local features which are invariant to certain geometric transformations (e.g., scaling and translation). Each image was then represented as a bag of "visual words", known as the "bag-of-words"

representation [26], or a set of "sparse codes", known as the "sparse coding" representation [25, 85, 86]. In addition to being problem-dependent, a common property

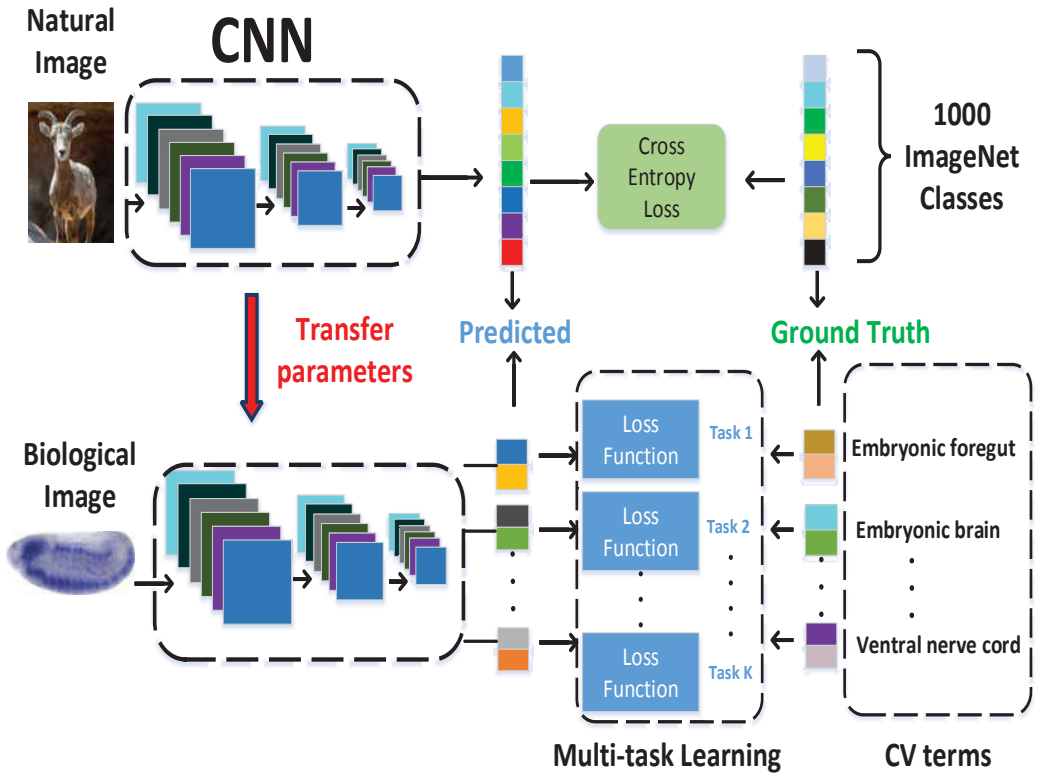
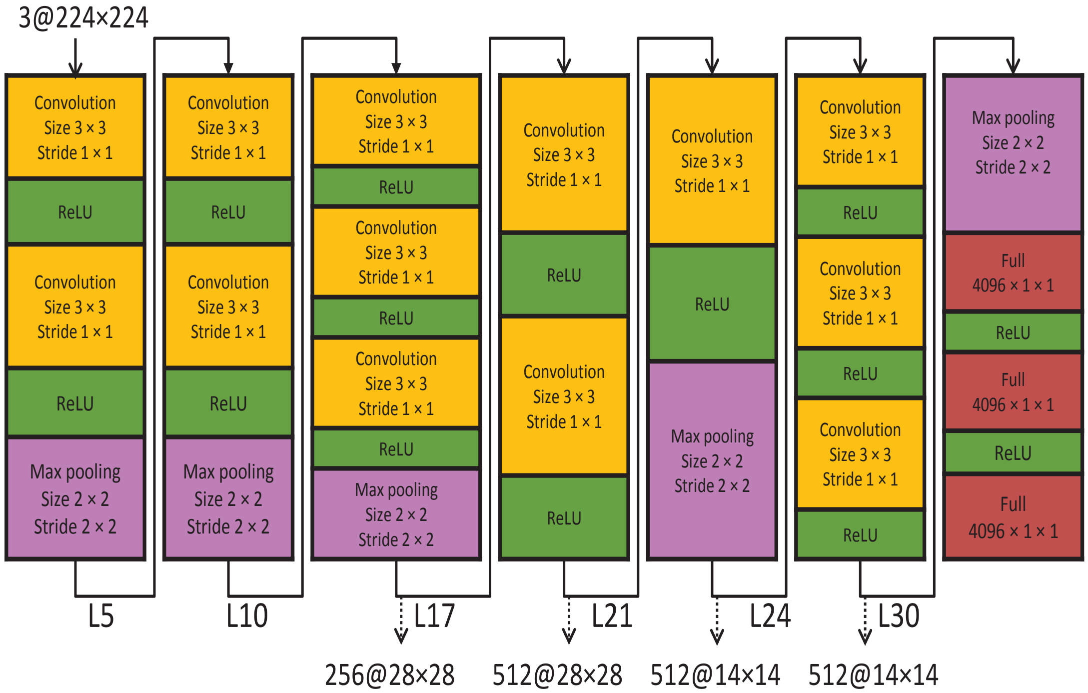


FIG. 19: Pipeline of deep models for transfer learning and multi-task learning.

of traditional feature extraction methods is that they are "shallow", because only one or two levels of feature extraction was applied, and the parameters for computing features are usually not trained using supervised algorithms. Given the complexity of patterns captured by biological images, these shallow models of feature extraction may not be sufficient. Therefore, it is desirable to develop a multi-layer feature extractor, alleviating the tedious process of manual feature engineering and enhancing the representation power. In this chapter, I propose to employ the deep learning methods to generate representations of ISH images. Deep learning models are a class



Input size	L17	L21	L24	L30
224×224	200704	401408	100352	100352

FIG. 20: Detailed architecture of the VGG model. “Convolution”, “Max pooling” and “ReLU” denote convolutional layer, max pooling layer and rectified linear unit function layer, respectively. This model consists of 36 layers. I extract features from layers 17, 21, 24, and 30.

of multi-level systems that can act on the raw input images directly to compute increasingly high-level representations. One particular type of deep learning models that have achieved practical success is the deep convolutional neural networks (CNNs) [87]. These models stack many layers of trainable convolutional filters and pooling operations on top of each other, thereby computing increasingly abstract representations of the inputs. Deep CNNs trained with millions of labeled natural images

using supervised learning algorithms have led to dramatic performance improvement in natural image recognition and detection tasks [88–90]. However, learning a deep CNN is usually associated with the estimation of millions of parameters, and this requires a large number of labeled image samples. This bottleneck currently prevents the application of CNNs to many biological problems due to the limited amount of labeled training data. To overcome this difficulty, I propose to develop generic and problem-independent feature extraction methods , which involves applying previously obtained knowledge to solve different but related problems. This is made possible by the initial success of transferring features among different natural image data sets [91–93]. These studies trained the models on the ImageNet data set that contains millions of labeled natural images with thousands of categories. The learned models are then applied to other image data sets for feature extraction, since layers of the deep models are expected to capture the intrinsic characteristics of visual objects.

In this chapter, I explore whether the transfer learning property of CNNs can be generalized to compute features for biological images. I propose to transfer knowledge from natural images by training CNNs on the ImageNet data set. To take this idea one step further, I propose to fine-tune the trained model with labeled ISH images, and resume training from already learned weights using multi-task learning schemes. The two models are then both used as a feature extractors to compute image features from *Drosophila* gene expression pattern images. The resulting features are subsequently used to train and validate my machine learning method for annotating gene expression patterns.

The overall pipeline of this work is given in Figure 19. The network is trained on the ImageNet data containing millions of labeled natural images with thousands of categories (top row). The pre-trained parameters are then transferred to the target domain of biological images. I first directly use the pre-trained model to extract features from *Drosophila* gene expression pattern images. I then fine-tune the trained model with labeled ISH images. I then employ the fine-tuned model to extract features to capture CV term-specific discriminative information (bottom row).

Experimental results show that my approach of using CNNs outperforms the sparse coding methods [86] for annotating gene expression patterns at different stage ranges. In addition, my results indicate that the transfer and fine-tuning of knowledge by CNNs from natural images is very beneficial for producing high-level representations of biological images. Furthermore, I show that the intermediate layers of CNNs produced the best gene expression pattern representations. This is because the early layers encode very primitive image features that are not enough to capture gene expression patterns. Meanwhile, the later layers capture features that are specific to the training natural image set, and these features may not be relevant to gene expression pattern images.

5.2 DEEP MODELS FOR TRANSFER LEARNING AND FEATURE EXTRACTION

Deep learning models are a class of methods that are capable of learning hierarchy of features from raw input images. Convolutional neural networks (CNNs)

are a class of deep learning models that were designed to simulate the visual signal processing in central nervous systems [87, 89, 94]. These models usually consist of alternating combination of convolutional layers with trainable filters and local neighborhood pooling layers, resulting in a complex hierarchical representations of the inputs. CNNs are intrinsically capable of capturing highly nonlinear mappings between inputs and outputs. When trained with millions of labeled images, they have achieved superior performance on many image-related tasks [87, 89, 90].

A key challenge in applying CNNs to biological problems is that the available labeled training samples are very limited. To overcome this difficulty and develop a universal representation for biological image informatics, I propose to employ transfer learning to transfer knowledge from labeled image data that are problem-independent. The idea of transfer learning is to improve the performance of a task by applying knowledge acquired from different but related task with a lot of training samples. This approach of transfer learning has already yielded superior performance on natural image recognition tasks [91–93, 95, 96].

In this chapter, I explore whether this transfer learning property of CNNs can be generalized to biological images. Specifically, the CNN model is trained on the ImageNet data containing millions of labeled natural images with thousands of categories and used directly as feature extractors to compute representations for ISH images. In this chapter, I apply the pre-trained VGG model [90] that was trained on the ImageNet data to perform several computer vision tasks, such as localization, detection and classification. There are two pre-trained models in [90], which are "16"

and "19" weight layers models. Since these two models generate similar performance on my ISH images, I use the "16" weight layers model in my experiment. The VGG architecture contains 36 layers. This network includes convolutional layers with fixed filter sizes and different numbers of feature maps. It also apply rectified non-linearity, max-pooling to different layers.

More details on various layers in the VGG weight layer model are given in Figure 20. Since the output feature representations of layers before the third max pooling layer involve larger feature vectors, I use each *Drosophila* ISH image as input to the VGG model and extracted features from layers 17, 21, 24, and 30 to reduce the computational cost. I then flatten all the feature maps and concatenated them into a single feature vector. For example, the number of feature maps in layer 21 is 512, and the corresponding size of feature maps is 28×28. Thus, the corresponding size of feature vector for this layer is 401,408.

5.3 DEEP MODELS FOR MULTI-TASK LEARNING

In addition to the transfer learning scheme described above, I also propose a multi-task learning strategy in which a CNN is first trained in the supervised mode using the ImageNet data and then fine-tuned on the labeled ISH *Drosophila* images. This strategy is different from the pre-trained model I use above. To be specific, the pre-trained model is designed to recognize objects in natural images while I study the CV term annotation of *Drosophila* images instead. Although the leveraged knowledge from the source task could reflect some common characteristics shared in these two types of images such as corners or edges, extra efforts are also needed to capture the

specific properties of ISH images. The *Drosophila* gene expression pattern images are organized into groups, and multiple CV term annotations are assigned to multiple images in the same group. This multi-image multi-label nature poses significant challenges to traditional image annotation methodologies. This is partially due to the fact that there are ambiguous multiple-to-multiple relationships between images and CV term annotations, since each group of images are associated with multiple CV term annotations.

I propose to use multi-task learning strategy to overcome the above difficulty. To be specific, I first employ a CNN model that is pre-trained on natural images to initialize the parameters of a deep network. Then, I fine-tune this network using multiple annotation term prediction tasks to obtain CV term-specific discriminative representation. The pipeline of my method is illustrated in Figure 19. I have a single pre-trained network with the same inputs but with multiple outputs, each of which corresponds to a term annotation task. These outputs are fully connected to a hidden layer that they share. Because all outputs share a common layer, the internal representations learned by one task could be used by other tasks. Note that the back-propagation is done in parallel on these outputs in the network. For each task, I use its individual loss function to measure the difference between outputs and the ground truth. In particular, I am given a training set of k tasks $\{X_i, y_i^j\}_{i=1}^m$, $j = 1, 2, \ldots, k$, where $X_i \in R^n$ denotes the i-th training sample, m denotes the total number of training samples. Note that I use the same groups of samples for different tasks, which is a simplified version of traditional multi-task learning. The output

label y_i^j denotes the CV term annotation status of training sample, which is binary with the form

$$y_i^j = \begin{cases} 1 & \text{if } X_i \text{ is annotated with the } j\text{-th CV term,} \\ 0 & \text{otherwise.} \end{cases}$$

To quantitatively measure the difference between the predicted annotation results and ground truth from human experts, I use a loss function in the following form:

$$loss(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{m}\sum_{j=1}^{k}\left(y_i^j log f(\hat{y}_i^j) + (1-y_i^j) log(1-f(\hat{y}_i^j))\right),$$

where

$$f(q) = \begin{cases} \frac{1}{1+e^{-q}} & \text{if} \quad q \geq 0 \\ 1 - \frac{1}{1+e^{-q}} & \text{if} \quad q < 0, \end{cases}$$

and $\mathbf{y} = \{y_i^j\}_{i,j=1}^{m,k}$ denotes the ground truth label matrix over different tasks, and $\hat{\mathbf{y}} = \{y_i^j\}_{i,j=1}^{m,k}$ is the output matrix of my network through feedforward propagation. Note that $\hat{y}_i^j$ denotes the network output before the softmax activation function. This loss function is a special case of the cross entropy loss function by using sigmoid function to induce probability representation [97, 98]. Note that my multi-task loss function is the summation of multiple loss functions, and all of them are optimized simultaneously during training.

5.4 BIOLOGICAL IMAGE ANALYSIS

The *Drosophila melanogaster* has been widely used as a model organism for the study of genetics and developmental biology. To determine the gene expression patterns during *Drosophila* embryogenesis, the Berkeley *Drosophila* Genome Project

(BDGP) used high throughput RNA *in situ* hybridization (ISH) to generate a systematic gene expression image database [14, 16]. In BDGP, each image captures the gene expression patterns of a single gene in an embryo. Each gene expression image is annotated with a collection of anatomical and developmental ontology terms using a CV term annotation to identify the characteristic structures in embryogenesis. This annotation work is now mainly carried out manually by human experts, which makes the whole process time-consuming and costly. In addition, the number of available images is now increasing rapidly. Therefore, it is desirable to design an automatic and systematic annotation approach to increase the efficiency and accelerate biological discovery [17, 18, 20, 24, 99, 100].

TABLE 6: Statistics of the data set used in this chapter. The table shows the total number of images for each stage range and the numbers of positive samples for each term.

Stages	Number of images	# of positive samples for each term									
		No. 1	No. 2	No. 3	No. 4	No. 5	No. 6	No. 7	No. 8	No. 9	No. 10
4-6	4173	953	438	1631	1270	1383	1351	351	568	582	500
7-8	1953	782	741	748	723	753	668	510	340	165	209
9-10	2153	899	787	778	744	694	496	559	452	350	264
11-12	7441	2945	2721	2056	1932	1847	1741	1400	1129	767	1152
13-17	7564	2572	2169	2062	1753	1840	1699	1273	1261	891	1061

Prior studies have employed machine learning and computer vision techniques to automate this task. Due to the effects of stochastic process in development, every

embryo develops differently. In addition, the shape and position of the same embryonic part may vary from image to image. Thus, how to handle local distortions on the images is crucial for building robust annotation methods. The seminal work in [101] employed the wavelet-embryo features by using the wavelet transformation to project the original pixel-based embryonic images onto a new feature domain. In subsequent work, local patches were first extracted from an image and local features which are invariant to certain geometric transformations (e.g., scaling and translation) were then computed from each patch. Each image was then represented as a bag of "visual words", known as the "bag-of-words" representation [26], or a set of "sparse codes", known as the "sparse coding" representation [25,86]. All prior methods used handcrafted local features combined with high-level methods, such as the bag-of-words or sparse coding schemes, to obtain image representations. These methods can be viewed as two-layer feature extractors. In this chapter, I propose to employ the deep CNNs as a multi-layer feature extractor to generate image representations for CV term annotation.

I show here that a universal feature extractor trained on problem-independent data set can be used to compute feature representations for CV term annotation. Furthermore, the model trained on problem-independent data set, such as the ImageNet data, can be fine-tuned on labeled data from specific domains using the error back propagation algorithm. This will ensure that the knowledge transferred from problem-independent images is adapted and tuned to capture domain-specific features in biological images. Since generating manually annotated biological images is

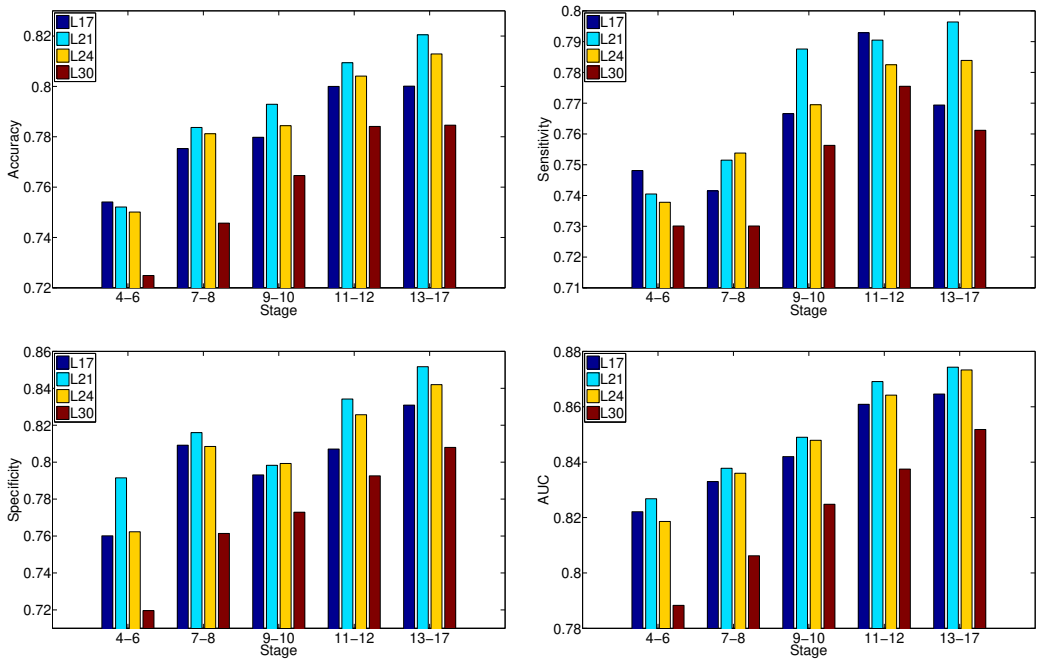


FIG. 21: Comparison of annotation performance achieved by features extracted from different layers of deep models for transfer learning over five stage ranges. "Lx" denotes the hidden layer from which the features were extracted.

both time-consuming and costly, the transfer of knowledge from other domains, such as the natural image world, is essential in achieving competitive performance.

5.5 EXPERIMENTAL EVALUATION

5.5.1 EXPERIMENTAL SETUP

In this study, I use the *Drosophila* ISH gene expression pattern images provided by the FlyExpress database [20, 21], which contains genome-wide, standardized images from multiple sources, including the Berkeley *Drosophila* Genome Project (BDGP). For each *Drosophila* embryo, a set of high-resolution, two-dimensional image series

were taken from different views (lateral, dorsal, and lateral-dorsal and other intermediate views). These images were then subsequently standardized semi-manually. In this study, I focus on the lateral-view images only, since most of images in FlyExpress are in lateral view.

In the FlyExpress database, the embryogenesis of *Drosophila* has been divided into six discrete stage ranges (stages 1-3, 4-6, 7-8, 9-10, 11-12, and 13-17). I use those images in the later 5 stage ranges in the CV term annotation, since only a very small number of keywords were used in the first stage range. One characteristic of these images is that a group of images from the same stage and same gene are assigned with the same set of keywords. Prior work in [86] has shown that image-level annotation outperformed group-level annotation using the BDGP images. In this chapter, I focus on the image-level annotation only and used the same top 10 keywords that are most frequently annotated for each stage range as in [86]. The statistics of the numbers of images and most frequent 10 annotation terms for each stage range are given in Table 6.

For CV term annotation, my image data set is highly imbalanced with much more negative samples than positive ones. For example, there are 7564 images in stages 13-17, but only 891 of them are annotated the term "dorsal prothoracic pharyngeal muscle". The commonly-used classification algorithms might not work well for my specific problem, because they usually aimed to minimizing the overall error rate without paying special attention to the positive class. Prior work in [86] has shown that using under-sampling with ensemble learning could produce better prediction

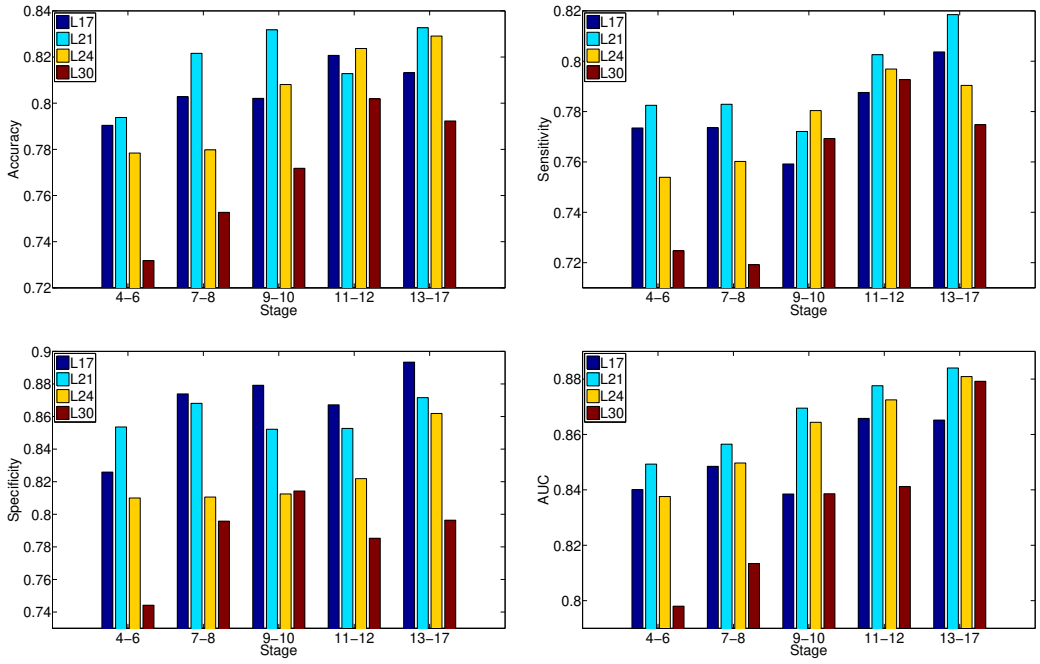


FIG. 22: Comparison of annotation performance achieved by features extracted from different layers of the deep models for multi-task learning over five stage ranges. "Lx" denotes the hidden layer from which the features were extracted.

performance. In particular, I selectively under-sample the majority class to obtain the same number of samples as the minority class and built a model for each sampling. This process is performed many times for each keyword to obtain a robust prediction. Following [86], I employ classifier ensembles built on biased samples to train robust models for annotation. In order to further improve the performance, I produce the final prediction by using majority voting, since this sample scheme is one of the widely used methods for fusion of multiple classifiers. For comparison purpose, I also implement the existing sparse coding image representation method studied in [86]. The annotation performance is measured using accuracy, specificity, sensitivity and area under the ROC curve (AUC) for CV term annotation. For all of these measures,

a higher value indicates better annotation performance. All classifiers used in this chapter are the ℓ_2-norm regularized logistic regression.

5.5.2 COMPARISON OF FEATURES EXTRACTED FROM DIFFERENT LAYERS

The deep learning model consists of multiple layer of feature maps for representing the input images. With this hierarchical representation, a natural question is which layer has the most discriminative power to capture the characteristics of input images. When such networks were trained on natural image data set such as the ImageNet data, the features computed in lower layers usually correspond to local features of objects such as edges, corners or edge/color conjunctions. In contrast, the features encoded at higher layers mainly represent class-specific information of the training data. Therefore, for the task of natural object recognition, the features extracted from higher layers usually yield better discriminative power [93].

In order to identify the most discriminative features for the gene expression pattern annotation tasks, I compare the features extracted from various layers of the VGG network. Specifically, I use the ISH images as inputs to the pre-trained VGG network and extracted features from layers 17, 21, 24, and 30 for each ISH image. These features are used for the annotation tasks, and the results are given in Figure 21. I can observe that for all stage ranges, layer 21 features outperformed other features in terms of overall performance. Specifically, the discriminative power increases from layer 17 to layer 21, and then drops afterwards as the depth of network increases. This indicates that gene expression features are best represented in the

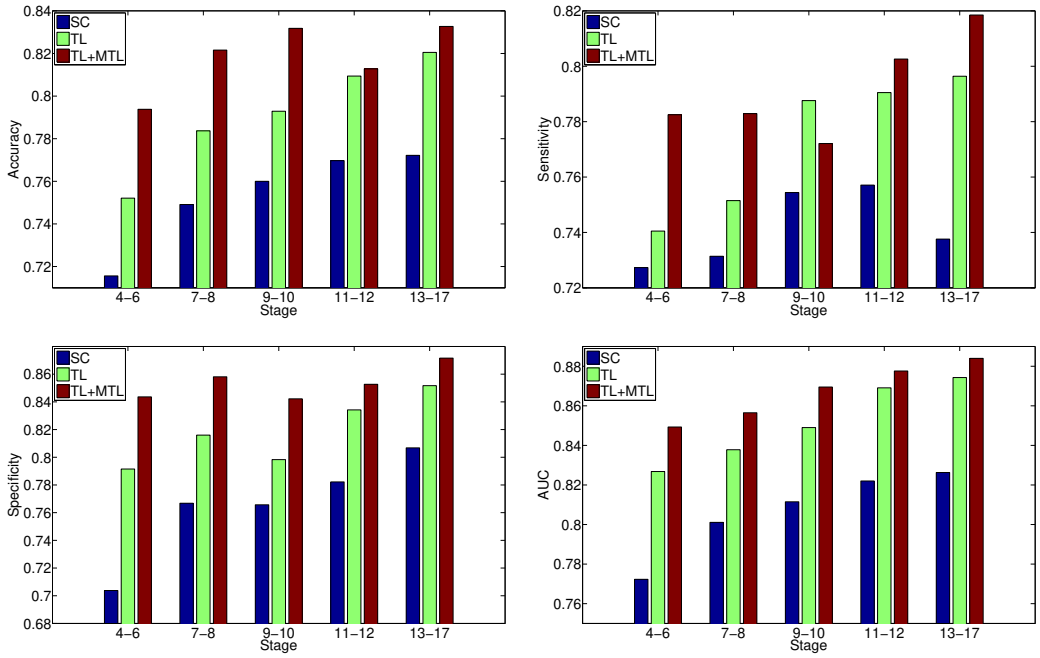


FIG. 23: Performance comparison of different methods. "SC" denotes sparse coding. "TL" and "TL + MTL" denote the performance achieved by transfer learning and multi-task learning models, respectively. I only consider the features extracted from layer 21 of these two deep models.

intermediate layers of CNN that was trained on natural image data set. One reasonable explanation about this observation is the lower layers compute very primitive image features that are not enough to capture gene expression patterns. Meanwhile, the higher layers capture features that are specific to the training natural image set, and these features may not be relevant for gene expression pattern images.

Then I propose to use multi-task learning strategy to fine-tune the pre-trained network with labeled ISH images. In order to show the gains through fine-tuning on pre-trained model, I extract features from the same hidden layers that are used for the pre-trained model. I report the predictive performance achieved by features of

different layers in the proposed fine-tuned model in Figure 22. It can be observed from the results that the predictive performance was generally higher on middle layers in the deep architecture. In particular, layer 21 outperforms other layers significantly. This result is consistent with the observation found on the pre-trained model.

TABLE 7: Performance comparison in terms of accuracy, sensitivity, specificity, and AUC achieved by CNN models and Sparse Coding features for all stage ranges. "TL+MTL" and "TL" denote the features extracted from layer 21 of the deep model for multi-task learning and transfer learning. "SC" denotes the performance of the sparse coding features.

Measures	Methods	Stage 4-6	Stage 7-8	Stage 9-10	Stage 11-12	Stage 13-17
Accuracy	TL+MTL	0.7938±0.0381	0.8216±0.0231	0.8318±0.0216	0.8128±0.0325	0.8327±0.0256
	TL	0.7521±0.0326	0.7837±0.0269	0.7929±0.0231	0.8094±0.0331	0.8205±0.0304
	SC	0.7217±0.0352	0.7401±0.0351	0.7549±0.0303	0.7659±0.0326	0.7681±0.0231
Sensitivity	TL+MTL	0.7825±0.0372	0.7829±0.0368	0.7721±0.0412	0.8026±0.0401	0.8185±0.0259
	TL	0.7405±0.0293	0.7515±0.0342	0.7876±0.0401	0.7905±0.0389	0.7964±0.0317
	SC	0.7321±0.0408	0.7190±0.0331	0.7468±0.0298	0.7576±0.0329	0.7328±0.0235
Specificity	TL + MTL	0.8436±0.0376	0.8581±0.0380	0.8422±0.0284	0.8527±0.0252	0.8716±0.0256
	TL	0.7915±0.0247	0.8160±0.0316	0.7983±0.0315	0.8342±0.0237	0.8517±0.0306
	SC	0.7140±0.0389	0.7605±0.0392	0.7629±0.0298	0.7749±0.0329	0.8005±0.0298
AUC	TL + MTL	0.8493±0.0427	0.8565±0.0279	0.8695±0.0276	0.8776±0.0291	0.8824±0.0197
	TL	0.8344±0.0439	0.8401±0.0346	0.8508±0.0257	0.8702±0.0271	0.8746±0.0299
	SC	0.7687±0.0432	0.7834±0.0358	0.7921±0.0294	0.8061±0.0342	0.8105±0.0280

5.5.3 COMPARISON WITH PRIOR METHODS

I also compare the performance achieved by different methods including sparse

coding, transfer learning model and multi-task learning. These results demonstrate that my deep model with multi-task learning are able to accurately annotate gene expression images over all embryogenesis stage ranges. To compare my generic features with the domain-specific features used in [86], I compare the annotation performance of my deep learning features with that achieved by the domain-specific sparse coding features. Deep learning models include transfer learning and multi-task learning. In this experiment, I only consider the features extracted from layer 21 since they yielded the best performance among different layers. The performance of these three types of features averaged over all terms is given in Figure 23 and Table 7. I can observe that the deep model for multi-task learning features outperform the sparse coding features and transfer learning features consistently and significantly in all cases. To examine the performance differences on individual anatomical terms, I show the AUC values on each term in Figure 24 for different stage ranges. I can observe that my features extracted from layer 21 of the VGG networks for transfer learning and multi-task learning outperformed the sparse coding features over all stage ranges for all terms consistently. These results demonstrate that my generic features of deep models are better at representing gene expression pattern images than the problem-specific features based on sparse coding.

In Figure 25, I provide a term-by-term and image-by-image comparison between the results of the deep model for multi-task learning and the sparse coding features for the 10 terms in stages 13-17. The x-axis corresponds to the 10 terms. The y-axis corresponds to a subset of 50 images in stages 13-17 with the largest numbers of

annotated terms. The gene names and the FlyExpress image IDs in parentheses are displayed. The prediction results of different methods compared with the ground truth are distinguished by different colors. The white entries correspond to predictions agreed upon by these two methods, while non-white entries were used to denote different types of disagreements. Specifically, the green and blue entries correspond to correct predictions by the multi-task learning features but incorrect predictions by the sparse coding features. Green and blue indicate positive and negative samples, respectively, in the ground truth. Similarly, the red and pink entries correspond to incorrect predictions by the multi-task learning features but correct predictions by the sparse coding features. Red and pink indicate positive and negative samples, respectively, in the ground truth. Overall, it is clear that the total number of green and blue entries is much more than the number of red and pink entries, indicating that, among all predictions disagreed by these two methods, the predictions by the multi-task learning features are correct most of the time.

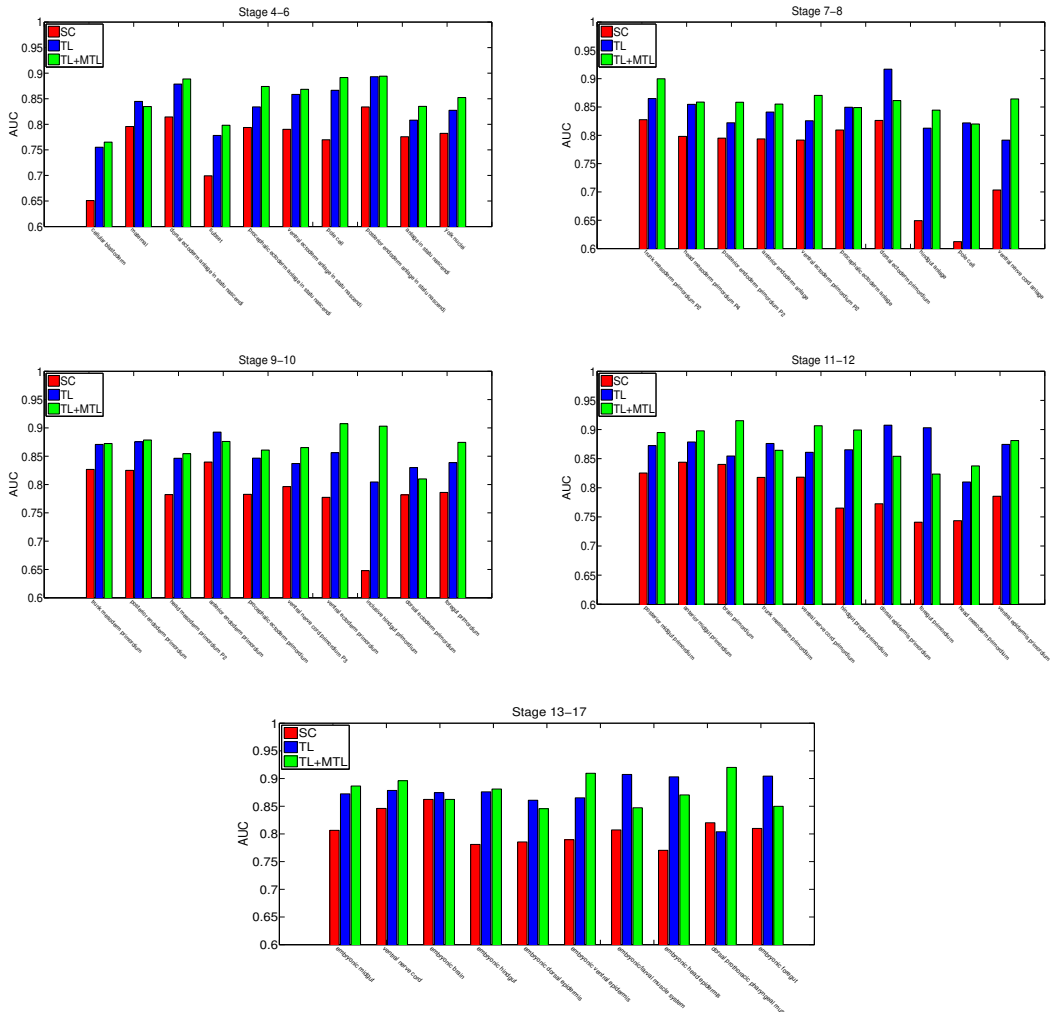


FIG. 24: Performance comparison of different methods for all stage ranges. "SC", "TL" and "TL + MTL" denote sparse coding, transfer learning and multi-task learning models, respectively.

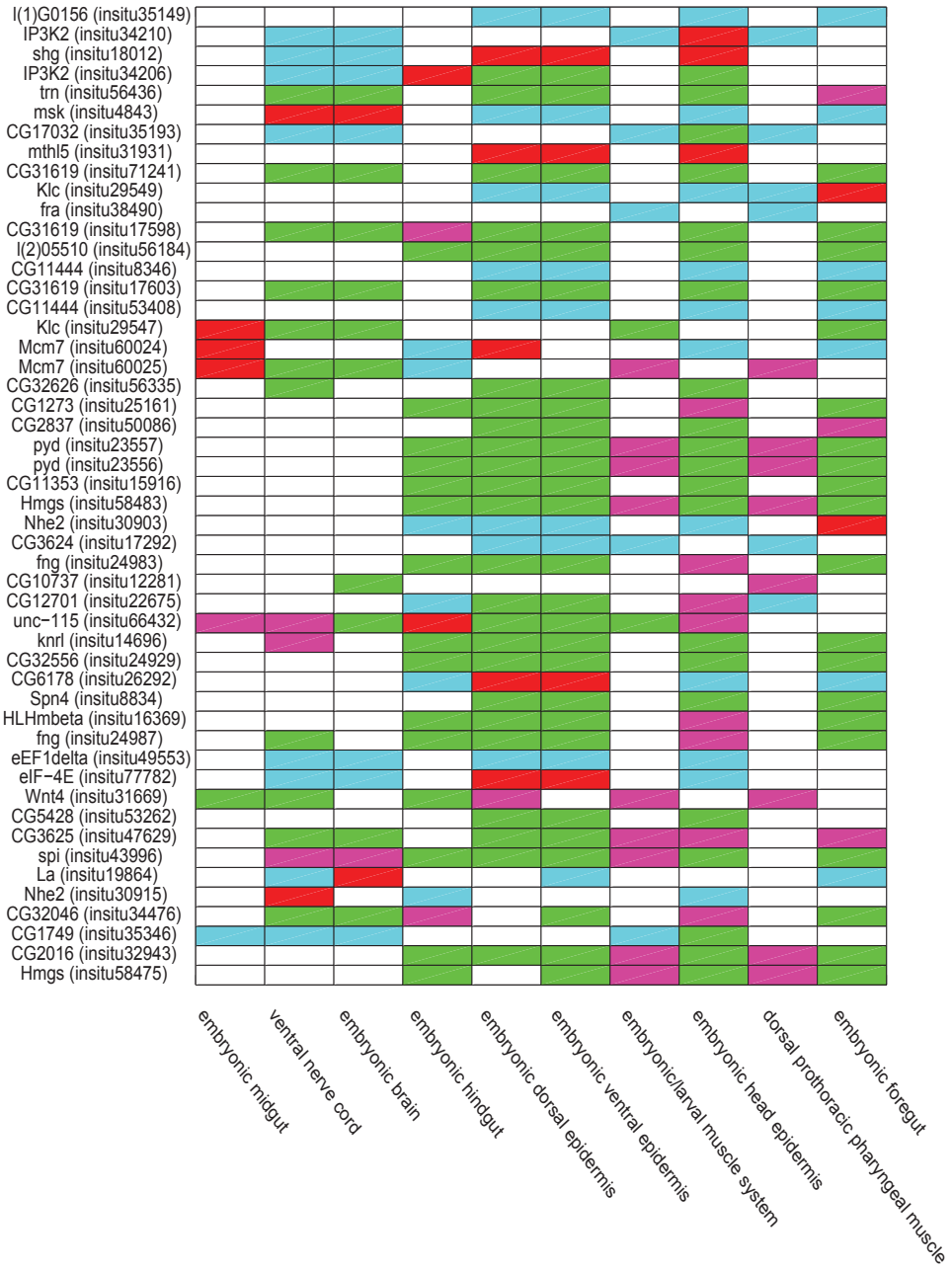


FIG. 25: Comparison of prediction results between the deep models for multi-task learning and the sparse coding features for the 10 terms in stages 13-17.

CHAPTER 6

CONCLUSION AND OUTLOOK

The major theme of this dissertation is to demonstrate several computational approaches can be applied in large scale and complex biological data. I propose computational approaches for identifying co-expressed embryonic domains and the associated genes simultaneously across multiple developmental stages. I also develop problem-independent feature extraction methods to generate hierarchical representations for ISH images.

In model construction, I propose a probabilistic model for evolutionary co-clustering. I propose an EM algorithm to perform maximum likelihood parameter estimation for the probabilistic model. The proposed methods are evaluated on both synthetic and real date sets. Results show that the proposed method consistently outperforms prior methods. I describe a method for unsupervised learning from bipartite graphs. In many applications, the relational data are more conveniently captured by k-partite graphs. I will extend my method for unsupervised mining of dynamic k-partite graphs.

In the analysis of *Drosophila* gene expression pattern images , I develop a mesh generation pipeline that maps the expression patterns of many genes into the same coordinate space. I then employ a co-clustering formulation to cluster the mesh

elements and the genes. This identifies co-expressed genes and spatial embryonic domains simultaneously. Experimental results show that the embryonic domains identified in this purely data-driven manner correspond to many embryonic structures. Results also show that the co-clusters of gene and embryonic domains accurately reflect the underlying biology.

In the Allen Developing Mouse Brain Atlas, I develop a co-clustering method and evaluate the method on both synthetic and real developing mouse brain ISH data. The model is motivated from a matrix fatorization perspective and admits a probabilistic interpretation. Experimental results on synthetic data demonstrate that my method is superior to prior methods. Application of my method to the developing mouse brain identifies brain voxel clusters that are more consistent with neuroanatomical results than other methods. Currently I do not consider the time varying nature of the developing mouse brain data. This is primarily due to the difficulty that the brain voxels are not registered across developmental stages. I will explore advanced methods that can incorporate temporal smoothness into clustering. Although I mainly focus on the developing mouse brain data, the proposed co-clustering method is generic and can be applied to other domains. I will explore more applications in the future.

In the biological image analysis, I propose to employ the deep convolutional neural networks as a multi-layer feature extractor to generate generic representations for ISH images. I use the deep convolutional neural network trained on large natural image set as feature extractors for ISH images. I first directly use the model trained on

natural images as feature extractors. I then employ multi-task classification methods to fine-tune the pre-trained model with labeled ISH images. Although the number of annotated ISH images is small, it nevertheless improved the pre-trained model. I compare the performance of my generic approach with the problem-specific methods. Results show that my proposed approach significantly outperforms prior methods on ISH image annotation. I also show that the intermediate layers of deep models produce the best gene expression pattern representations. In the current study, I focus on using deep models for CV annotation. There are many other biological image analysis tasks that require appropriate image representations such as developmental stage prediction. I will consider broader applications in the future. I consider a simplified version of the problem in which each term is associated with all images in the same group. I will extend my model to incorporate the image group information in the future.

REFERENCES

[1] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, March 2007.

[2] S. Asur, S. Parthasarathy, and D. Ucar, "An event-based framework for characterizing the evolutionary behavior of interaction graphs," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 913–921, 2007.

[3] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu, "GraphScope: parameter-free mining of large time-evolving graphs," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 687–696, 2007.

[4] H. Tong, S. Papadimitriou, P. S. Yu, and C. Faloutsos, "Proximity tracking on time-evolving bipartite graphs," in *Proceedings of the SIAM International Conference on Data Mining*, pp. 704–715, 2008.

[5] A. Saha and V. Sindhwani, "Learning evolving and emerging topics in social media: a dynamic nmf approach with temporal regularization," in *Proceedings of the fifth ACM international conference on Web search and data mining*, pp. 693–702, ACM, 2012.

[6] Q. Mei and C. Zhai, "Discovering evolutionary theme patterns from text: an exploration of temporal text mining," in *Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 198–207, 2005.

[7] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in *Proceedings of the 29th International Conference on Very Large Data Bases*, pp. 81–92.

[8] D. Chakrabarti, R. Kumar, and A. Tomkins, "Evolutionary clustering," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 554–560.

[9] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng, "On evolutionary spectral clustering," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, pp. 17:1–17:30, December 2009.

[10] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, "Analyzing communities and their evolutions in dynamic social networks," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, pp. 8:1–8:31, April 2009.

[11] F. Wang, H. Tong, and C.-Y. Lin, "Towards evolutionary nonnegative matrix factorization," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[12] H. Lodish, A. Berk, C. A. Kaiser, M. Krieger, M. P. Scott, A. Bretscher, H. Ploegh, and P. Matsudaira, *Molecular Cell Biology*. W. H. Freeman, 6th ed., 2007.

[13] E. H. Davidson, *The Regulatory Genome: Gene Regulatory Networks in Development and Evolution*. Academic Press, 2006.

[14] P. Tomancak, A. Beaton, R. Weiszmann, E. Kwan, S. Shu, S. E. Lewis, S. Richards, M. Ashburner, V. Hartenstein, S. E. Celniker, and G. M. Rubin, "Systematic determination of patterns of gene expression during *Drosophila* embryogenesis," *Genome Biology*, vol. 3, no. 12, pp. research0088.1–0088.14, 2002.

[15] E. Lécuyer, H. Yoshida, N. Parthasarathy, C. Alm, T. Babak, T. Cerovina, T. Hughes, P. Tomancak, and H. Krause, "Global analysis of mRNA localization reveals a prominent role in organizing cellular architecture and function," *Cell*, vol. 131, pp. 174–187, 2007.

[16] P. Tomancak, B. Berman, A. Beaton, R. Weiszmann, E. Kwan, V. Hartenstein, S. Celniker, and G. Rubin, "Global analysis of patterns of gene expression during *Drosophila* embryogenesis," *Genome Biology*, vol. 8, no. 7, p. R145, 2007.

[17] S. Kumar, K. Jayaraman, S. Panchanathan, R. Gurunathan, A. Marti-Subirana, and S. J. Newfeld, "BEST: a novel computational approach for comparing gene expression patterns from early stages of *Drosophila melanogaster* develeopment," *Genetics*, vol. 169, pp. 2037–2047, 2002.

[18] E. Frise, A. S. Hammonds, and S. E. Celniker, "Systematic image-driven analysis of the spatial *Drosophila* embryonic expression landscape," *Molecular Systems Biology*, vol. 6, p. 345, 2010.

[19] E. Lécuyer and P. Tomancak, "Mapping the gene expression universe," *Current Opinion in Genetics & Development*, vol. 18, no. 6, pp. 506–512, 2008.

[20] S. Kumar, C. Konikoff, B. Van Emden, C. Busick, K. T. Davis, S. Ji, L.-W. Wu, H. Ramos, T. Brody, S. Panchanathan, J. Ye, T. L. Karr, K. Gerold, M. McCutchan, and S. J. Newfeld, "FlyExpress: visual mining of spatiotemporal patterns for genes and publications in *Drosophila* embryogenesis," *Bioinformatics*, vol. 27, no. 23, pp. 3319–3320, 2011.

[21] B. Van Emden, H. Ramos, S. Panchanathan, S. Newfeld, and S. Kumar, "Flyexpress: an image-matching web-tool for finding genes with overlapping patterns of expression in drosophila embryos," *Tempe, AZ*, vol. 85287530, 2006.

[22] T. Walter, D. W. Shattuck, R. Baldock, M. E. Bastin, A. E. Carpenter, S. Duce, J. Ellenberg, A. Fraser, N. Hamilton, S. Pieper, M. A. Ragan, J. E. Schneider, P. Tomancak, and J.-K. Hériché, "Visualization of image data from cells to organisms," *Nature Methods*, vol. 7, pp. S26–S41, 2010.

[23] H. Peng, "Bioimage informatics: a new area of engineering biology," *Bioinformatics*, vol. 24, no. 17, pp. 1827–1836, 2008.

[24] S. Ji, L. Sun, R. Jin, S. Kumar, and J. Ye, "Automated annotation of *Drosophila* gene expression patterns using a controlled vocabulary," *Bioinformatics*, vol. 24, no. 17, pp. 1881–1888, 2008.

[25] L. Yuan, A. Woodard, S. Ji, Y. Jiang, Z.-H. Zhou, S. Kumar, and J. Ye, "Learning sparse representations for fruit-fly gene expression pattern image annotation and retrieval," *BMC Bioinformatics*, vol. 13, no. 107, 2012.

[26] S. Ji, Y.-X. Li, Z.-H. Zhou, S. Kumar, and J. Ye, "A bag-of-words approach for *Drosophila* gene expression pattern annotation," *BMC Bioinformatics*, vol. 10, no. 1, p. 119, 2009.

[27] L. W. Swanson, *Brain Architecture: Understanding the Basic Plan.* Oxford University Press, 2nd ed., 2011.

[28] E. S. Lein and *et al.*, "Genome-wide atlas of gene expression in the adult mouse brain," *Nature*, vol. 445, no. 7124, pp. 168–176, 2007.

[29] M. J. Hawrylycz and *et al.*, "An anatomically comprehensive atlas of the adult human brain transcriptome," *Nature*, vol. 489, no. 7416, pp. 391–399, 2012.

[30] S. M. Sunkin, L. Ng, C. Lau, T. Dolbeare, T. L. Gilbert, C. L. Thompson, M. Hawrylycz, and C. Dang, "Allen Brain Atlas: an integrated spatio-temporal portal for exploring the central nervous system," *Nucleic Acids Research*, 2012.

[31] L. Puelles and J. L. Rubenstein, "Forebrain gene expression domains and the evolving prosomeric model," *Trends in neurosciences*, vol. 26, no. 9, pp. 469–476, 2003.

[32] C. Watson, G. Paxinos, and L. Puelles, *The Mouse Nervous System*. Academic Press, 2011.

[33] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 888–905, August 2000.

[34] U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395–416, December 2007.

[35] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems 14*, pp. 849–856, 2001.

[36] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 551–556, 2004.

[37] F. R. K. Chung, *Spectral Graph Theory*. 1997.

[38] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 269–274, 2001.

[39] H. Zha, X. He, C. Ding, H. Simon, and M. Gu, "Bipartite graph partitioning and data clustering," in *Proceedings of the tenth International Conference on Information and Knowledge Management*, pp. 25–32, 2001.

[40] K. Yu, S. Yu, and V. Tresp, "Soft clustering on graphs," in *Advances in Neural Information Processing Systems 18* (Y. Weiss, B. Schölkopf, and J. Platt, eds.), pp. 1553–1560, Cambridge, MA: MIT Press, 2006.

[41] N. Green, M. Rege, X. Liu, and R. Bailey, "Evolutionary spectral co-clustering," in *The 2011 International Joint Conference on Neural Networks*, pp. 1074–1081, 2011.

[42] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[43] F. R. Bach and M. I. Jordan, "Learning spectral clustering, with application to speech separation," *J. Mach. Learn. Res.*, vol. 7, December 2006.

[44] G. H. Golub and C. F. van Loan, *Matrix computations (3. ed.)*. Johns Hopkins University Press, 1996.

[45] J. R. Shewchuk, "Triangle: Engineering a 2D quality mesh generator and delaunay triangulator," in *Applied Computational Geometry: Towards Geometric Engineering* (M. C. Lin and D. Manocha, eds.), vol. 1148 of *Lecture Notes in Computer Science*, pp. 203–222, Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry.

[46] J. R. Shewchuk, "Delaunay refinement algorithms for triangular mesh generation," *Computational Geometry: Theory and Applications*, vol. 22, pp. 21–74, May 2002.

[47] P. Foteinos, A. Chernikov, and N. Chrisochoides, "Fully Generalized 2D Constrained Delaunay Mesh Refinement," *SIAM Journal on Scientific Computing*, vol. 32, pp. 2659–2686, 2010.

[48] A. Chernikov and N. Chrisochoides, "Generalized Insertion Region Guides for Delaunay Mesh Refinement," *SIAM Journal on Scientific Computing*, vol. 34, pp. A1333–A1350, 2012.

[49] L. M. Goering, P. K. Hunt, C. Heighington, C. Busick, P. S. Pennings, J. Hermisson, S. Kumar, and G. Gibson, "Association of orthodenticle with natural variation for early embryonic patterning in *Drosophila melanogaster*," *Journal of Experimental Zoology Part B-Molecular and Developmental Evolution*, vol. 312B, pp. 841–854, 2009.

[50] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The Finite Element Method: Its Basis and Fundamentals*. Butterworth-Heinemann; 6 edition, 2005.

[51] H. Peng and E. W. Myers, "Comparing *in situ* mRNA expression patterns of *Drosophila* embryos," in *Proceedings of the Eighth Annual International Conference on Resaerch in Computational Molecular Biology*, pp. 157–166, 2004.

[52] V. Hartenstein, *Atlas of* Drosophila *Development*. Cold Spring Harbor Laboratory Press, 1995.

[53] J. A. Campos-Ortega and V. Hartenstein, *The Embryonic Development of Drosophila Melanogaster*. Springer, second ed., 1997.

[54] L. Wolpert, J. Smith, T. Jessell, P. Lawrence, E. Robertson, and E. Meyerowitz, *Principles of Development*. Oxford University Press, 3rd ed., 2006.

[55] M. Ashburner and *et al.*, "Gene Ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, pp. 25–29, 2000.

[56] E. I. Boyle and *et al.*, "GO::TermFinder–open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes," *Bioinformatics*, vol. 20, no. 18, pp. 3710–3715, 2004.

[57] Y. Cheng and G. M. Church, "Biclustering of expression data," in *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pp. 93–103, 2000.

[58] H. Cho and I. S. Dhillon, "Coclustering of human cancer microarrays using minimum sum-squared residue coclustering," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 5, pp. 385–400, July 2008.

[59] O. E. Livne and G. H. Golub, "Scaling by binormalization," *Numerical Algorithms*, vol. 35, pp. 97–120, 2004.

[60] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene Ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, pp. 25–29, 2000.

[61] A. Stathopoulos and M. Levine, "Genomic regulatory networks and animal development," *Developmental Cell*, vol. 9, no. 4, pp. 449–462, 2005.

[62] T. Sandmann, C. Girardot, M. Brehme, W. Tongprasit, V. Stolc, and E. E. Furlong, "A core transcriptional network for early mesoderm development in *Drosophila melanogaster*," *Genes & Development*, vol. 21, no. 4, pp. 436–449, 2007.

[63] K. Yu, S. Yu, and V. Tresp, "Soft clustering on graphs," in *Advances in Neural Information Processing Systems 18* (Y. Weiss, B. Schölkopf, and J. Platt, eds.), pp. 1553–1560, Cambridge, MA: MIT Press, 2006.

[64] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Mach. Learn.*, vol. 42, no. 1-2, pp. 177–196, 2001.

[65] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine learning*, vol. 42, no. 1-2, pp. 177–196, 2001.

[66] L. Finesso and P. Spreij, "Nonnegative matrix factorization and I-divergence alternating minimization," *Linear Algebra and its Applications*, vol. 416, no. 2-3, pp. 270–287, 2006.

[67] J. A. Hartigan, "Direct clustering of a data matrix," *Journal of the American Statistical Association*, vol. 67, no. 337, pp. 123–129, 1972.

[68] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein, "Spectral biclustering of microarray data: Coclustering genes and conditions," *Genome Research*, vol. 13, no. 4, pp. 703–716, 2003.

[69] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," in *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 89–98, 2003.

[70] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra, "Minimum sum-squared residue co-clustering of gene expression data," in *Proceedings of the Fourth SIAM International Conference on Data Mining*, 2004.

[71] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha, "A generalized maximum entropy approach to Bregman co-clustering and matrix approximation," *J. Mach. Learn. Res.*, vol. 8, pp. 1919–1986, Dec. 2007.

[72] S. Busygin, O. Prokopyev, and P. M. Pardalos, "Biclustering in data mining," *Computers and Operations Research*, vol. 35, pp. 2964–2987, September 2008.

[73] M. Lee, H. Shen, J. Z. Huang, and J. S. Marron, "Biclustering via sparse singular value decomposition," *Biometrics*, vol. 66, pp. 1087–1095, 2010.

[74] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[75] M. Jagalur, C. Pal, E. Learned-Miller, R. T. Zoeller, and D. Kulp, "Analyzing *in situ* gene expression in the mouse brain with image registration, feature extraction and block clustering," *BMC Bioinformatics*, vol. 8, no. Suppl 10, p. S5, 2007.

[76] J. W. Bohland and *et al.*, "Clustering of spatial gene expression patterns in the mouse brain and comparison with classical neuroanatomy," *Methods*, vol. 50, no. 2, pp. 105–112, 2010.

[77] A. Prelić, S. Bleuler, P. Zimmermann, A. Wille, P. Bühlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler, "A systematic comparison and evaluation of biclustering methods for gene expression data," *Bioinformatics*, vol. 22, pp. 1122–1129, 2006.

[78] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: A survey," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 1, pp. 24–45, January 2004.

[79] T. M. Murali and S. Kasif, "Extracting conserved gene expression motifs from gene expression data," in *Pacific Symposium on Biocomputing*, pp. 77–88, 2003.

[80] G. Paxinos, *The Rat Nervous System*. Academic Press, 3rd ed., 2004.

[81] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[82] K. Theiler, *The House Mouse: Atlas of Embryonic Development*. Springer, 1972.

[83] L. Yuan, C. Pan, S. Ji, M. McCutchan, Z.-H. Zhou, S. Newfeld, S. Kumar, and J. Ye, "Automated annotation of developmental stages of *Drosophila* embryos in images containing spatial patterns of expression," *Bioinformatics*, vol. 30, no. 2, pp. 266–273, 2014.

[84] W. Zhang, D. Feng, R. Li, A. Chernikov, N. Chrisochoides, C. Osgood, C. Konikoff, S. Newfeld, S. Kumar, and S. Ji, "A mesh generation and machine learning framework for *Drosophila* gene expression pattern image analysis," *BMC Bioinformatics*, vol. 14, p. 372, 2013.

[85] S. Ji, L. Yuan, Y.-X. Li, Z.-H. Zhou, S. Kumar, and J. Ye, "*Drosophila* gene expression pattern annotation using sparse features and term-term interactions," in *Proceedings of the Fifteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 407–416, 2009.

[86] Q. Sun, S. Muckatira, L. Yuan, S. Ji, S. Newfeld, S. Kumar, and J. Ye, "Image-level and group-level models for *Drosophila* gene expression pattern annotation," *BMC Bioinformatics*, vol. 14, p. 350, 2013.

[87] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, November 1998.

[88] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[89] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), pp. 1106–1114, 2012.

[90] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[91] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proceedings of the 31st International Conference on Machine Learning*, pp. 647–655, 2014.

[92] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014.

[93] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proceedings of the European Conference on Computer Vision*, pp. 818–833, Springer, 2014.

[94] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[95] M. Oquab, I. Laptev, L. Bottou, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings*

of the 27th IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[96] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems*, pp. 3320–3328, 2014.

[97] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[98] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, pp. 160–167, ACM, 2008.

[99] I. Pruteanu-Malinici, D. L. Mace, and U. Ohler, "Automatic annotation of spatial expression patterns via sparse Bayesian factor models," *PLoS Comput Biol*, vol. 7, p. e1002098, 07 2011.

[100] K. Puniyani, C. Faloutsos, and E. P. Xing, "SPEX2: automated concise extraction of spatial gene expression patterns from fly embryo ISH images," *Bioinformatics*, vol. 26, no. 12, pp. i47–56, 2010.

[101] J. Zhou and H. Peng, "Automatic recognition and annotation of gene expression patterns of fly embryos," *Bioinformatics*, vol. 23, no. 5, pp. 589–596, 2007.

APPENDIX A

MANUAL OF MESH CLUSTERING

This open source software includes three modules. I already put my whole package on "github", which can be found at https://github.com/DIVE-WSU/MeshClustering.

A.1 FLYMESH

Step 1: Unpack the archive

The package contains the source code for implementing image-to-mesh generation.

Step 2: Build the triangulator

I use "Triangle", a two-dimensional quality delaunay triangulator as the basic triangulator of my image-to-mesh generation software.

Change the directory to "/some directory of your unpack file/I2MGenerator", and type the following commands in the shell:

- make distclean
- make

After this step, you will see a binary "triangle" file in the directory. Open the file "MeshEllipse.m", change the variable "path" to the directory where you build the triangulator. Now, the triangulator is ready to use.

Step 3: Run the file "run.m" in MATLAB

The variable "area" in file "imageBoundaryMesh.m" represents the upper area bound of triangles in mesh. Through changing the value of this variable, the user can change the number of triangles in mesh.

A.2 EVOLUTIONARY SOFT CO-CLUSTERING

Input Parameters:

- A: matrix of data $m \times n$
- cluster: number of cluster
- mu: alpha=0 is pure co-clustering
- repli: repeat computing times
- iter: number of iterations
- torr: when the errors are smaller than "torr", the algorithm stops

Output:

- IDX: row indicator cluster matrix
- IDY: column indicator cluster matrix
- err: index error
- ferr: feature error

A.3 SHOW MESH

The visualization tool that displays the resulting mesh of gene expression after co-clustering.

Step 1: Unpack the archive

Step 2: Run the file "ShowMesh.m" in MATLAB

Figure 26 is one sample example of the showmesh visualization for 40 clusters including the triangle number (1000) on stage 4-6.

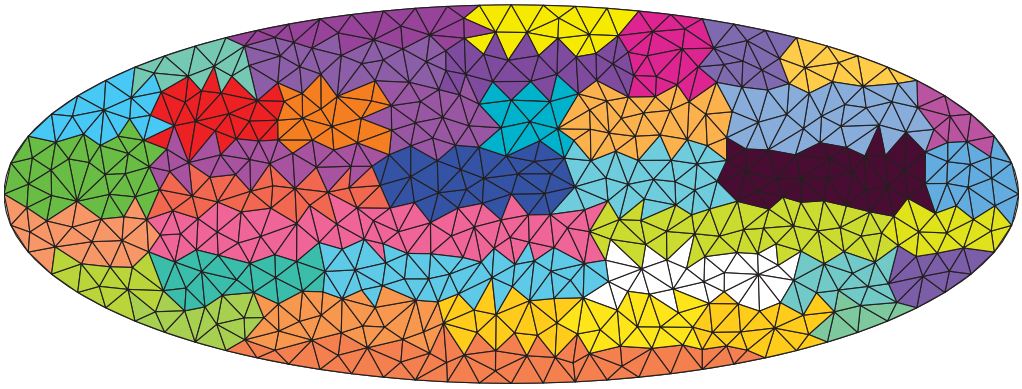


FIG. 26: Clusters of mesh elements when the number of clusters is 40 on the stage 4-6 expression patterns.

APPENDIX B

MANUAL OF SOFTWARE: CAFFE

Caffe is an open source framework for state-of-the-art deep learning algorithms. The framework is released under the BSD 2-Clause license, which is mainly written in C++ with Python and MATLAB bindings. Caffe is maintained and developed by the Berkeley Vision and Learning Center (http://caffe.berkeleyvision.org/).

B.1 INSTALLATION

B.1.1 PREREQUISITES

Before installing Caffe, several dependencies are required. CUDA is required for GPU mode. Library version 7+ and the later driver version are recommended.

Pycaffe and Matcaffe interfaces have their own natural needs.

- For Python Caffe: python 2.7 or python 3.3+
- For MATLAB Caffe: MATLAB with mex compiler

Other dependencies:

- OpenCV >= 2.4 including 3.0
- BLAS via ATLS, MKL, or OpenBLAS

B.1.2 COMPILATION AND TEST

- cp Makefile.config.example Makefile.config
- make clean
- make all
- make runtest

B.2 TRAIN A NETWORK

A key challenge in applying Caffe to biological problems is that the available labled training samples are very limited. To overcome this difficulty and develop a universal representation for biological image informatics, I employ transfer learning and multi-task learning to make extracted features generic.

I select pre-trained VGG model that was trained on the ImageNet data to perform several computer vision tasks. Several other pre-trained models can be found in "Model Zoo" of Caffe. You can choose a pre-trained model based on your specific image tasks.

Inputer Parameters:

- Solver.protxt includes the CNN architecture and biological images directory.
- VGG.caffemodel is the pre-trianed model.
- -gpu: the index of GPU that is used.

Example Command:

./build/tools/caffe train -solver models/solver.prototxt -weights models/vgg/VGG.caffemodel -gpu 8

VITA

Wenlu Zhang

Department of Computer Science

Old Dominion University

Norfolk, VA 23529

I received my Bachelor degree from Information Engineering University in China and Master degree from City College of New York, both in Computer Science. In Summer 2011, I joined in Computer Science Department of Old Dominion University and started my research in machine learning, data mining and computational biology. I have worked on clustering of time-varying data and have applied my new methods to a number of biological applications, including gene expression pattern analysis in fruit fly embryo and mouse brain. I have also worked on deep convolutional neural networks and have designed multiple 2D and 3D CNN models for medical and biological image analysis. I have already published six papers in highly-regarded conferences and journals, and I serve as primary or co-author of a total of eleven papers.

Typeset using LaTeX.