Spatiotemporal gene networks from ISH images

Kriti Puniyani

CMU-LTI-13-016

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

Thesis Committee:

Eric P. Xing, Chair
Jaime Carbonell
Robert Murphy
John Lafferty (University of Chicago)
Uwe Ohler (Duke University)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
In Language and Information Technologies

Abstract

As large-scale techniques for studying and measuring gene expressions have been developed, automatically inferring gene interaction networks from expression data has emerged as a popular technique to advance our understanding of cellular systems. Accurate prediction of gene interactions, especially in multicellular organisms such as Drosophila or humans, requires temporal and spatial analysis of gene expressions, which is not easily obtainable from microarray data. New image based techniques using in-situ hybridization(ISH) have recently been developed to allow large-scale spatial-temporal profiling of whole body mRNA expression. However, analysis of such data for discovering new gene interactions still remains an open challenge.

This thesis studies the question of predicting gene interaction networks from ISH data in three parts. First, we present SPEX^2, a computer vision pipeline to extract informative features from ISH data. Next, we present an algorithm, GINI, for learning spatial gene interaction networks from embryonic ISH images at a single time step. GINI combines multi-instance kernels with recent work in learning sparse undirected graphical models to predict interactions between genes.

Finally, we propose NP-MuScL (nonparanormal multi source learning) to estimate a gene interaction network that is consistent with multiple sources of data, having the same underlying relationships between the nodes. NP-MuScL casts the network estimation problem as estimating the structure of a sparse undirected graphical model. We use the semiparametric Gaussian copula to model the distribution of the different data sources, with the different copulas sharing the same covariance matrix, and show how to estimate such a model in the high dimensional scenario.

We apply our algorithms on more than 100,000 Drosophila embryonic ISH images from the Berkeley Drosophila Genome Project. Each of the 6 time steps in Drosophila embryonic development is treated as a separate data source. With spatial gene interactions predicted via GINI, and temporal predictions combined via NP-MuScL, we are finally able to predict spatiotemporal gene networks from these images.

To my parents, Jisha, and Tejaswi.

Acknowledgments

If I have seen further, it is by standing on the shoulders of giants.
– Isaac Newton

I would like to thank my advisor Eric Xing for his mentorship, and particularly, for giving me the freedom to explore different research paths. Eric helped me understand how to pick good problems, and how to communicate good results, both invaluable skills to have. I would also like to thank my thesis committee: Jaime Carbonell, John Lafferty, Robert Murphy, and Uwe Ohler for their very valuable suggestions and advice that dramatically improved this thesis.

During my years spent at CMU, my collaborators gave me many insightful comments, and through meaningful discussions, helped me develop a broader view of the community. These include Seyoung Kim, Christos Faloutsos, John Lafferty, Jacob Eisenstein, Shay Cohen, Sonal Gupta, Mahesh Joshi, Seunghak Lee, Sivaraman Balakrishnan, and my Google internship mentor, Danny Wyatt. Soumen Chakrabarti introduced me to research, and defined my world view on what it means to be a researcher, and to do good research. I would also like to thank the entire SAILING group, who during the course of our research and reading meetings, helped me develop and improve my research ideas.

Carnegie Mellon University has been a very friendly and collaborative place, with random conversations leading to interesting ideas all the time. Matt Harrison, Carlos Guestrin, Larry Wasserman, Geoff Gordon, John Woolford, and William Cohen were amazing in the courses they taught. I am also thankful to Bob Frederking, Michelle Martin, Stacey Young, and the entire LTI staff for all their help over the years.

Ramnath, Jose, Sivaraman, Sourish, and Mladen started the Ph.D. with me, and defended at approximately the same time as well! I have been fortunate to have such amazing friends the entire time I spent at CMU, and they made my life in Pittsburgh exciting and fun. I was lucky enough to have the best housemates possible - Bhavana, Meghana, and Ruta, they made 5838 Hobart a home, and became a second family to me. The friends I made at CMU will hopefully stay with me for life - Beena, Eakta, Gaurav, Nikita, Alankrita, Leman, Dan, Gorana, Wooyoung, Pradipta, Hetunandan, Suyash, Athula, Sunayana, Anjali, Archna, Kenneth, Amr, Miro, Arthur, Swapnil, Yulia, Siddharth, and others. The MM (Aditee, Amruta, Kuhoo, Meghana, Neela, Reena, Rukma, Sapna), Kuldeep, Aniket, Akshat, and Rahul were friends I could always count on.

My mom has always been my inspiration and a constant source of encouragement, my sister Jisha and my mother-in-law have been the people who made me smile no matter what. My husband Tejaswi has been the sounding board, the cheerleader, the devil's advocate, and the motivational speaker, all in one. This thesis is dedicated to them, and all my family and friends.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In multicellular organisms such as the metazoans, many important biological processes such as development and differentiation depend fundamentally on the spatial and temporal control of gene expression (Davidson, 2001; Gilbert, 2003). To date, the molecular basis and regulatory circuitry underlying metazoan gene regulation remains largely unknown. Numerous algorithmic approaches have been attempted to infer "networks" of regulatory elements from high-throughput experimental data, such as microarray profiles (Dobra et al., 2004; Ong, 2002; Segal et al., 2003), ChIP-chip genome localization data (Bar-Joseph et al., 2003; Harbison et al., 2004), and protein-protein interaction data (Causier, 2004; Giot et al., 2003; Kelley et al., 2004), based on formalisms such as Bayesian networks (Cowell et al., 1999) or graph mining (Tanay et al., 2004). Comparisons of different methods used for reverse engineering gene networks have been performed (Bansal et al., 2007; Hache et al., 2009), and predictions made by automatically learned gene networks have been experimentally validated (Carro et al., 2010; Wang et al., 2009), thus increasing the credibility of such approaches.

This progress notwithstanding, a key deficiency of these approaches is that they rely heavily on high-throughput biological data like microarrays that only capture average behavior of the genes and proteins in a large cell population from, e.g., a cell culture, a dissected tissue, or even a homogenized whole animal. For multicellular organisms such as Drosophila and human, gene expressions must be described in a spatiotemporal context, which reveals the histological specificities and temporal dynamics of the activities of the gene. Such information is not available from the standard whole-animal microarray data which record only the average expression of each gene over all cells in the body, nor is it easily obtainable from "tissue-specific" microarray assays using advanced micro-dissection and cell-sorting techniques (Figure 1.1).

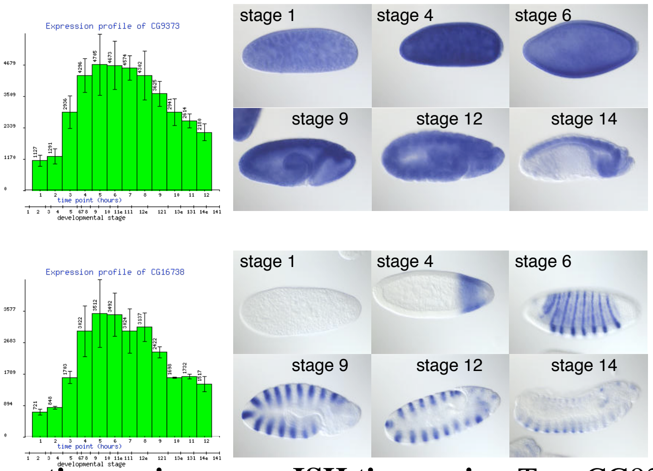


Figure 1.1. Microarray time series versus ISH time series. Top: CG9373 (RNA binding protein). Bottom: CG16738 (RNA polymerase II TF). Although the 2 time courses of whole-body mRNA ambulance measured by microarray are nearly indistinguishable, the ISH data reveals distinctive spatio-temporal patterns. (Courtesy of Dr. Hanchuan Peng who pointed these genes to us.)

In situ hybridization (ISH) assay is an imaging method to visualize mRNA expression in tissues and cells without homogenizing the specimens to be analyzed and therefore retains the original histological context of gene expression. Recent advancements in image-based genome-scale profiling technology such as whole-body mRNA abundance micro-imaging via *in situ hybridization* (ISH) have begun to reveal a more holistic view of the activities and functions of genes in rich spatial-temporal contexts. ISH has been used to characterize whole genome expression patterns for different species such as Drosophila embryos (Tomancak et al., 2002b, 2007), C. elegans (Yuji, 2001), and adult mouse brain (Lein et al., 2007), and at smaller scales for Arabidopsis flowers (Wellner and Meyerowitz, 2004), testicular germ cell tumors (Almstrup and Leffers, 2004), and others. The availability of this form of gene expression data calls for development of next-generation image analysis systems to facilitate not only efficient pattern mining such as image clustering or retrieval, but also in-depth reasoning of complex spatial-temporal relationships between gene expression patterns, which will be essential for functional genomics and regulatory network inference in higher organisms. Such information is indispensable for in-depth analysis of gene regulation networks, developmental mechanisms, and oncogenic processes in higher eukaryotic organisms (Montalta-He and Reichert, 2003).

In this thesis, we focus on a particularly interesting, but previously unaddressed challenge along this direction: inferring a statistically sound gene network from gene expression micro-imaging data, in the same sense of inferring a gene network from microarray data as widely studied in the literature. Analyzing ISH data allows us to infer a network by computing similarities in the spatial and temporal distributions of gene expressions in Drosophila embryo. Analyzing the temporal changes of the spatial distributions of genes could reveal how a gene regulation network evolves over time during dynamic biological processes such as embryogenesis (Ahmed and Xing, 2009a).

Systematic profiling of ISH images capturing gene expressions over the entire span of Drosophila embryogenesis are now being undertaken at a whole genome scale, offering an unprecedented opportunity for investigators to compare the spatio-temporal behavior of genes and begin assembling realistic pictures of gene regulatory networks underlying the developmental process (Tomancak et al., 2002b). The fast growing "Expression Pattern" database under the Berkeley Drosophila Genome Project (BDGP, 2005) now contains over 100,000 digital images of expression patterns of ~ 7500 genes. As of now, the only analysis done on co-expressed genes in the BDGP is based on manual-labeling of the images by a domain expert using a controlled vocabulary. However, with the rapid growth of data volume, manual analysis is no longer feasible, and automatic analysis techniques are sorely needed, which require the development of new systems capable of noise removal, pattern extraction, feature description, and similarity measures.

1.1 Gene networks

Networks have been used to represent interactions between nodes in a large variety of application areas including gene co-expression networks (Stuart et al., 2003), protein-protein interaction networks (Giot et al., 2003; Uetz et al., 2000), cell interaction networks (Yook et al., 2004), etc. While microarrays measure co-expression, which can then be used to predict gene co-expression networks, current technology in ISH data measures only one gene (or a small set of genes) at a time, and not co-expression of thousands of genes simultaneously. The techniques developed in this thesis allow us to develop a network from ISH data, where an edge is represented between two genes, if their gene expression has similar spatial and temporal profiles. In essence, the gene network we wish to predict is one that captures co-localization of genes across various conditions.

Current work on analyzing ISH data has only focussed on extracting features from the ISH images (Kumar et al., 2002; Peng and Myers, 2004), clustering the images (Frise et al., 2010; Heffel et al., 2008), classifying images (Ji et al., 2009; Zhou and Peng, 2007) etc. This thesis is the first work that defines the notion of co-localization based gene networks, and proposes algorithms to learn a gene network from such ISH data.

1.2 About the BDGP data

While the techniques discussed in this thesis are generic to all ISH data, a major motivation of our work is the extensive imagery documentation of all the genes expressed during Drosophila em-

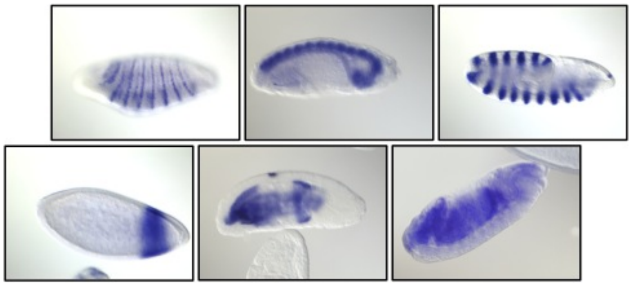


Figure 1.2. Examples of embryonic ISH images from the BDGP data. Presence of gene expression is indicated by the blue stain on the different regions of the embryo.

bryogenesis via ISH imaging by the Berkeley Drosophila Genome Project (BDGP) (Tomancak et al., 2002b). BDGP is an ongoing effort to determine gene expression patterns during embryogenesis for Drosophila genes. In February 2013, the data contained more than 110,000 ISH images capturing the expression pattern of 7516 genes. Each image is annotated with time information, indicating the development of the embryo in six development stage ranges. Each image documents the gene expression pattern of a single gene in an embryo. Most images have a single embryo, however some images capture partial views of the embryo, others have overlapping or touching embryos. This is an extremely interesting but difficult dataset that reveals unprecedented details of gene activities during metazoan embryogenesis, but at the same time posts large unanswered challenges on methodologies for systematic and principled analysis.

1.3 Thesis overview

This thesis addresses the challenge of inferring a network of gene interactions by analyzing spatio-temporal data obtained from ISH images. The overall problem can be summarized in cartoon form, in Figure 1.3. The main challenges in the correct estimation of the spatio-temporal network from such data can be divided into three sub-areas

1. **Image-specific feature extraction**: The ISH image data must be represented by a set of

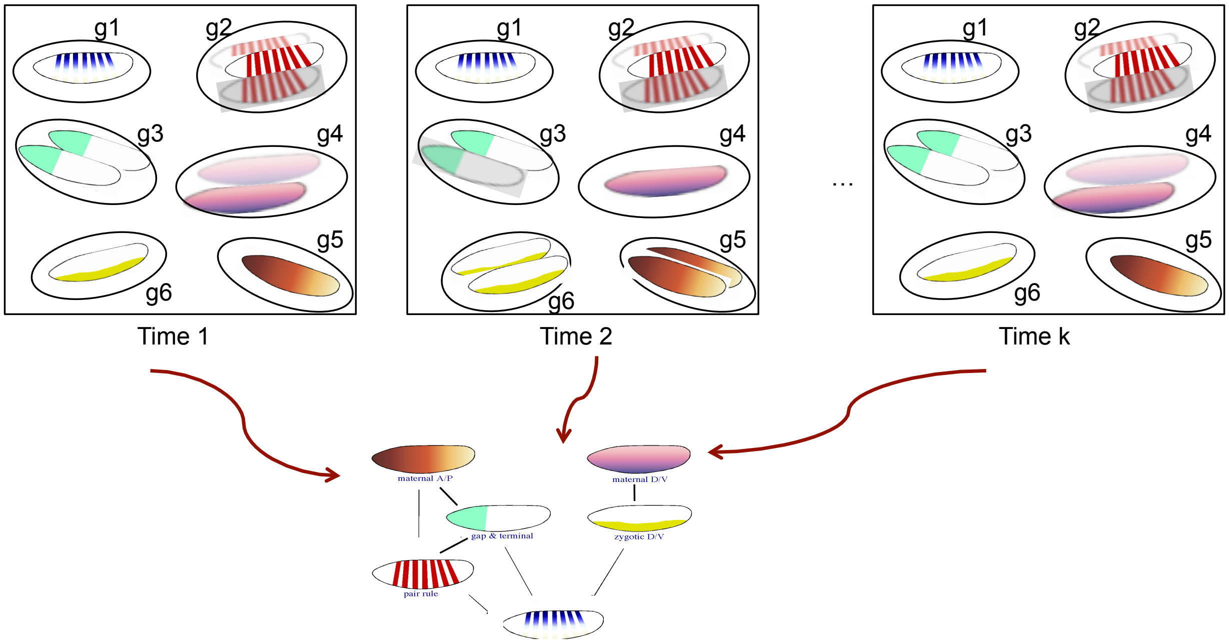


Figure 1.3. Gene interaction network prediction from ISH data. A cartoon figure summarizing the challenges in learning a spatio-temporal network from image data at multiple time points, where each gene at each time point may have a different number of images.

features that are independent of the size, shape, location, and orientation of the embryo, and accounts for differences in illumination etc. across images.

2. **Comparing images at a single time point**: At each time-point, we may have different number of images per gene. How do we compare such data, while learning a network that represents sparse, global conditional independencies in the data?

3. **Combining information across multiple time points** : Given an algorithm to estimate a gene network from a single time point, how can we extend it to combine information across multiple time points, with a reasonable generative model that explains the existence of multiple data sources/time points.

We hence propose a three-step solution to our problem - each step deals with challenges in each of the categories above. Chapter 2 proposes SPEX^2, an automatic image processing system for embryonic ISH image processing, which inputs Drosophila ISH images, and outputs the precise gene expression pattern observed in the image. Chapter 3 proposes GINI, an algorithm to learn a network from ISH images at a single time-point, using the features obtained from SPEX^2. Chapter 4 proposes NP-MuScL, an algorithm to combine data from multiple sources to infer a single network. Using data from each of the 6 time points in the BDGP data as a data source, we show how to infer a network from multiple time-points. Chapter 5 runs detailed experiments on the entire BDGP data set, verifying the overall effectiveness of our proposed methods. Finally, Chapter 6 summarizes our contributions, and proposes directions of future research.

Chapter 2

SPEX^2: Concise Gene Expression Pattern Extraction from Fly ISH

In this chapter, we present SPEX^2 (Puniyani et al., 2010), an automatic system for embryonic ISH image processing, which can extract, transform, compare, classify and cluster spatial gene expression patterns in Drosophila embryos. Our pipeline for gene expression pattern extraction outputs the precise spatial locations and strengths of the gene expression. We performed experiments on the *largest publicly available* collection of Drosophila ISH images, and show that our method achieves excellent performance in automatic image annotation, and also finds clusters that are significantly enriched, both for GO functional annotations, and for annotation terms from a controlled vocabulary used by human curators to describe these images.

2.1 Introduction

SPEX^2 (SPatial gene EXpression pattern EXtractor) is a highly effective and reliable image processing pipeline for automated and concise extraction of *bona fide* gene expression patterns (rather than generic *shaded areas* as usually recognized by naive pattern extracting procedures), from Drosophila embryonic ISH results imaged from the lateral view. Such patterns offer a high-fidelity surrogate of the spatial patterns of gene expression in a developing embryo or if necessary other subjects in question (Figure 2.1(c)), nearly free of misleading non-expression patterns due to poor quality staining/washing, body texture, color condensation caused by body anatomy, embryo shape and contour, etc., which often fool standard pattern extracting procedures, as endogenous gene expression patterns (Figure 2.1(b)). These patterns allow highly informative and specific feature representations of each gene to be generated, which can be used in a variety of downstream analysis like functional clustering, gene annotation, and network inference.

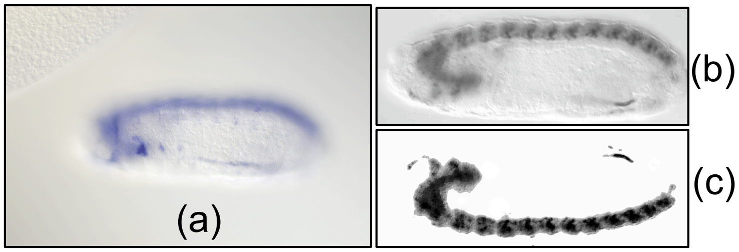


Figure 2.1. SPEX^2 example. (a) Original Image (b) Pattern extracted by standard procedures (c) Standardized gene expression pattern extracted by SPEX^2.

Specifically, we address the following questions in this paper:

1. Given an ISH image of a Drosophila embryo, how to find the pixels that correspond specifically to the spatial expression pattern, rather than other non-expressional entities such as body anatomies and textures, in the embryo?

2. How should a good representation of the gene expression pattern be constructed?

3. How should this representation be used for further clustering and classification tasks ?

Comparisons of gene expression patterns from different ISH images must be performed with respect to the embryo, and not the image. The position, orientation, size, shape contour, lighting condition, and texture of the embryo within the image do not matter, as long as the comparison is dependent on the location and strength of the gene expression within the embryo. This requires automated detection of the embryo in an image. Additionally, the orientation of the embryo needs to be identified and standardized, and the embryo must be registered to a standard shape. Furthermore, the ISH image contains noise in addition to the gene expression itself, due to staining artifacts. The correct expression pattern must be extracted from the registered image before conducting further analysis.

SPEX^2 converts every raw ISH image of Drosophila embryo into a feature representation of the spatial gene expression pattern thereof suitable for downstream quantitative analysis, based on the following 3 steps : (1) Embryo standardization, via embryo extraction, orientation correction, and registration. (2) Gene expression extraction via stain extraction and pattern segmentation, and (3) Feature extraction. Each step in the pipeline uses image processing and machine learning algorithms to extract the correct output. Automated error control methods detect and reject images if they are not being correctly analyzed, or if they are unsuitable for analysis due to imaging artifacts.

The resultant feature representation can be directly used for tasks like classification, clustering, standard correlation analysis and network inference of Drosophila genes in a metric space. Our techniques are automatic, and are not specific to any data set. Our pipeline also outputs spatial patterns of gene expression, that are amenable to easy interpretation by biologists.

As proof of concept, we demonstrate our technique on lateral view images from the Berkeley Drosophila Genome Project (BDGP) gene expression pattern database, from the time stage 13-16. To evaluate our pipeline, we cluster the genes based on the features extracted by SPEX^2, and report enrichment analysis, conducted using GO functional annotations, as well as enrichment of manual annotations describing the spatial expression localization using a controlled vocabulary. We also learn a classifier to annotate gene expression patterns during embryogenesis using a controlled vocabulary, and report classification accuracy. We find that we significantly outperform other standard feature extraction techniques from the computer vision community, as well as the techniques reported in previous work.

2.1.1 Related work

We build upon the first steps taken by earlier work to construct our analysis pipeline for Drosophila ISH images. The system *BEST*, developed by Kumar et al. (2002), performs a direct pixel-level comparison of binarized images, using the intersection of the foreground regions as a similarity measure for gene expression patterns. They develop an embryo enclosing algorithm to find the embryo outline, and extract the binary expression pattern via adaptive thresholding.

Li et al. (2009) propose multi-instance multi-label learning via appropriate kernels to improve performance specifically for annotating images using a controlled vocabulary. An extension was proposed by Ji et al. (2009) to model term-term interactions in a regression framework that has improved performance for this task. They extract position invariant features using a sparse codebook on aligned images, and apply a local regularization framework on these features for automatic image annotation.

Peng and Myers (2004); Zhou and Peng (2007) developed techniques to represent ISH images, based on Gaussian mixture models, principal component analysis, and wavelet functions. They use the wavelet features, with min-redundancy max-relevance feature selection, to automatically annotate images. Heffel et al. (2008) have also proposed a pipeline for this task, using embryo outline extraction, transformation of the embryo into a circular outline, and conversion to fourier-coefficients based feature representation. They report a visual clustering of seven images using their pipeline.

Tomancak et al. (2007) analyzed the global gene expression patterns in the BDGP data set, using only the manual annotations available for each gene from a controlled vocabulary. They reported clustering results on joint clustering of microarray data and annotation terms, and found interesting clusters that could not be found using microarray data alone.

Thus, these advances have offered important new insights and computational tools for mining image-based gene expression patterns captured by ISH, which we extend by conducting a detailed analysis of the information contained in ISH images, and how it can be captured in a good feature representation format.

2.2 Methods

The SPEX^2 system consists of three major components: (a) embryo standardization, (b) gene expression pattern extraction, and (c) feature representation. An illustration of the pipeline is given in Figure 2.2. Below, we describe each component in detail.

2.2.1 Embryo standardization

Given a raw ISH image, SPEX^2 uses an embryo standardization process to convert it into a standardized form suitable for subsequent expression extraction and pattern comparison. The embryo is extracted from the ISH image, and aligned along its A/P and D/V axis correcting for the orientation, thereby ensuring the anterior (of the embryo) is to the left and the dorsal surface is to the top of the image. Finally, the embryo is registered to a standard shape and size.

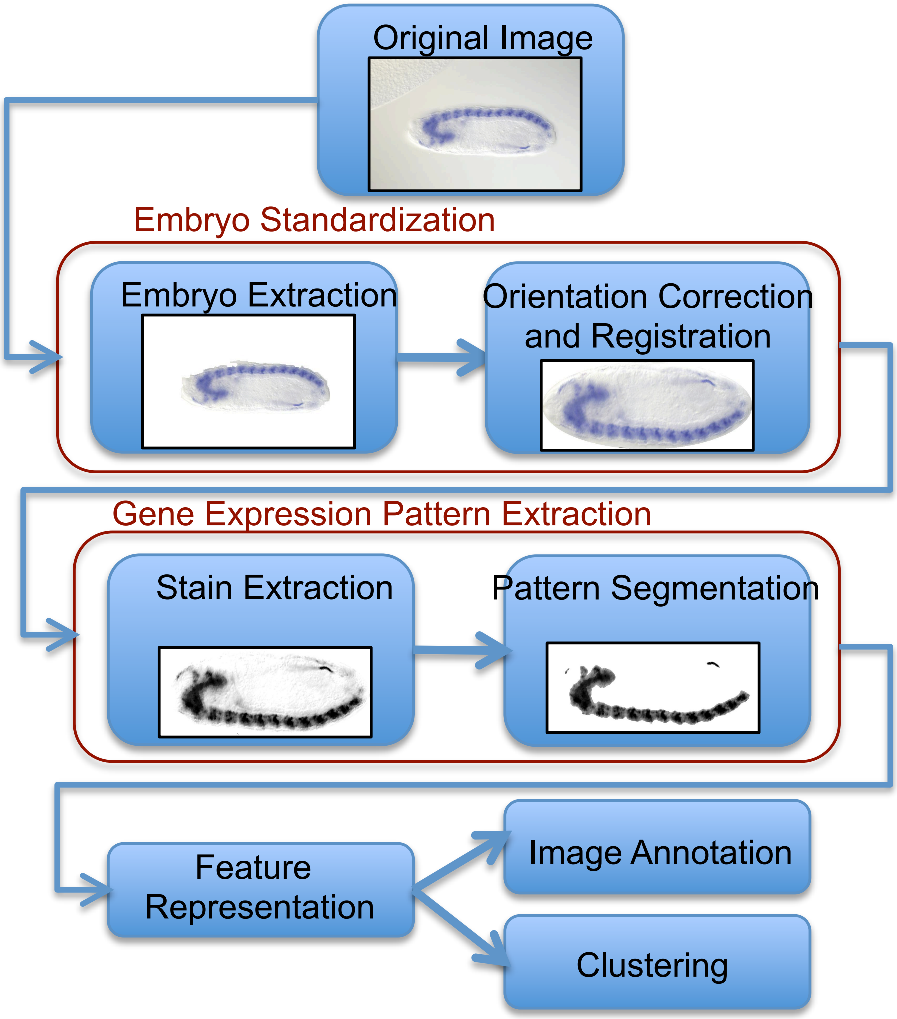


Figure 2.2. A schematic illustration of the SPEX^2 pipeline.

Embryo outline extraction

Our embryo extraction procedure works in two steps. First, a foreground object extractor is used to extract potential embryos in the image. Second, a series of increasingly complex tests filter out foreground objects that are not embryos, or are embryos not suitable for analysis.

The object extractor uses the Canny edge operator to identify regions with fast-changing color and high variance. A series of morphological operations (dilations and erosions) are used to smooth out the edges and close holes to find the foreground objects.

A sequence of tests are then applied to each foreground object to test whether it's an embryo suitable for further analysis; rejected items include erroneous outlines, partial embryos, multiple embryos physically touching or overlapping with each other, and excessively dried or otherwise mishandled embryos.

1. Objects touching the image boundary are rejected, since these may be partially imaged embryos.

2. Objects that are too small or too large are rejected. Small objects imply that a part of the actual embryo was potentially missed by the object extractor. Large objects are either partial embryos imaged using a large magnification, or incorrect outlines that include a portion of the background in the foreground object.

3. If the maximum distance between the object outline and the convex outline of the object is large, the image is rejected; ensuring that the embryo outline is almost convex.

4. Scale-independent shape features of the object outline are extracted and compared with expected shape features of a standard embryo. Scale independence is required since the size of the embryo varies across images. Examples of shape features include : (a) The ratio between the major and minor axes of the object must match the expected ratio for a Drosophila embryo. This ensures that the object is not too thin and narrow, nor is it too circular. (b) The centroid of the foreground object must be close to the centroid of its outlining rectangle (ensures symmetry). (c) The maximum (and mean) curvature of the object outline must be similar to the values expected for an embryo (filters out deformed embryos). If the value of any of the above features is more than 20% away from the feature value computed from a single correctly identified embryo, then the image is rejected.

Some examples of embryo outlines extracted by our algorithm are shown in Figure 2.3. Embryo extraction works well in presence of varying illumination (Figure 2.3(a)), when the background is

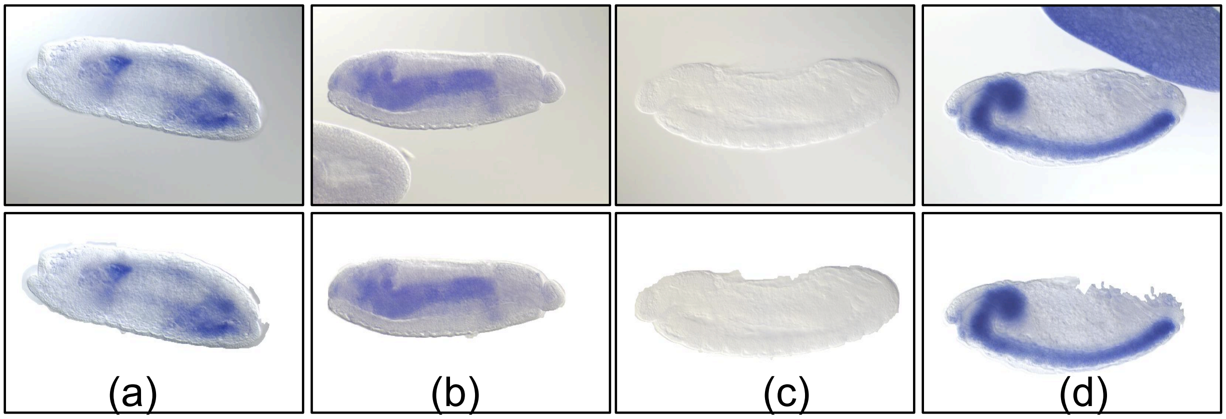


Figure 2.3. Embryo extraction. The top image shows the original image, and the bottom image shows the extracted embryo.

lighter than the foreground (Figure 2.3(b)), in the absence of stain in the embryo (Figure 2.3(c)), and when there are multiple embryos touching each other (Figure 2.3(d)).

Alignment, orientation detection and registration

To align all embryos for later comparisons, we assume the camera angle is perpendicular to the surface of the embryo, which is the case with most imaging technologies with zoom-in. An ellipse is fitted to the detected embryo outline, with the major axis of the ellipse assumed to be the A/P axis, and minor axis the D/V axis of the embryo; and the embryo is rotated so that the A/P axis is horizontal.

Next, the correct orientation of the aligned embryo is identified and standardized so that the head is to the left, tail to the right, dorsal part of the embryo at the top, and ventral part at the base. This is akin to a binary classification task, for which we need to determine whether to flip the embryo horizontally to correctly position the anterior part of the embryo to the left, and vertically to position the dorsal side to top. Gargesha et al. (2005) proposed a technique to automatically annotate the anterior/posterior sides of the embryo. However, their technique is supervised, requiring a large amount of pre-labeled data, which is tedious and expensive to generate. Additionally, their technique is based on a heuristic that does not utilize the knowledge of the expected gene expression patterns. As for finding the dorsal/ventral sides of the embryo, to our knowledge, no reported result is available so far.

We propose an algorithm for unsupervised embryo orientation detection, based on the insight that images of the same gene at the same time stage must have similar expression patterns. We start with a heuristic assignment to each embryo, and change the assignment of a particular embryo if it increases its similarity with other embryos stained with the same gene, in a greedy manner. The algorithm for A/P orientation detection for all embryos stained for a single gene is outlined in Figure 2.4, and is run for all genes being analyzed. A similar algorithm is used for D/V orientation, based on the heuristic that the dorsal side is less curved than the ventral side of the embryo. Though this is a greedy algorithm that assumes that the first embryo assignment is always correct, we found that it works well in practice. Some examples of orientation detection and correction of embryos is shown in Figure 2.5.

```
Data: $n$ embryos ($emb$) stained for the gene being analyzed
Result: Correct orientation assignment for each input embryo
for $i = 1...n$ do
    // heuristic assignment
    $assignment(i)$ = the thinner side of $emb(i)$ is the head;
    $confidenceScore(i)$ = the difference in width of the two sides;
end
Sort all $emb$ in descending order of $confidenceScore$;
for $i = 2...n$ do
    // compute mean similarity
    $s1 = \frac{1}{i-1} \sum_{j=1}^{i-1} sim(emb(i), emb(j))$ ;
    $s2 = \frac{1}{i-1} \sum_{j=1}^{i-1} sim(flip(emb(i)), emb(j))$ ;
    if $s2 > s1$ then
        // swap the heuristic assignment
        $assignment(i) = !assignment(i)$;
    end
end
```

Figure 2.4. Algorithm for A/P orientation detection.

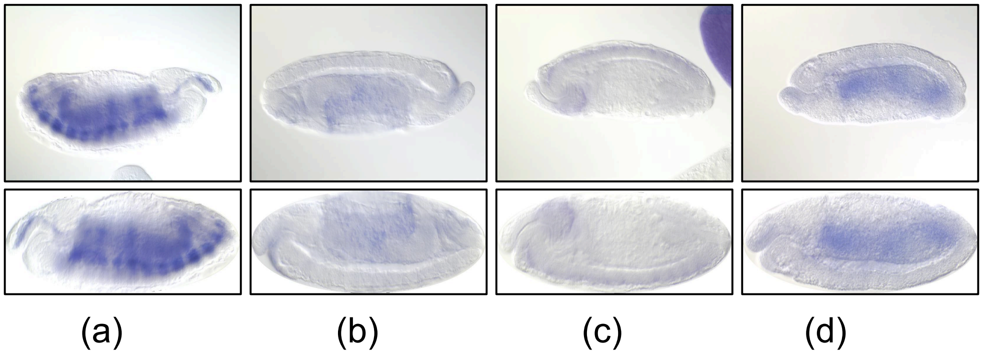


Figure 2.5. Orientation Detection. Flip (a) anterior/posterior, (b) both anterior/posterior and dorsal/ventral, (c) dorsal/ventral, (d) dorsal/ventral assignments of the embryo in the image. The top image shows the original image, and the bottom image shows the embryo outline after alignment, orientation detection and correction, and registration.

Finally, a registration algorithm using point-wise affine stretching is used to register the convex outline of the embryo to a standard ellipse shape. This enables us to obtain an exact map from pixel space to body part of the embryo. At the end of the standardization process, for all the processed images, there is a fixed correspondence between the image pixels and the various embryonic structures, enabling comparison of the spatial patterns of gene expression in different images by comparing the pixel-level expression values.

2.2.2 Concise gene expression pattern extraction

Given a standardized embryonic image, SPEX^2 extracts concise spatial gene expression patterns therein via a two-step procedure. First, standardized embryonic images are preprocessed to extract ISH stains. Then, noise in the stains are removed using a Markov Random Field (MRF) model based image segmentation. Our algorithm constructs the MRF graph structure and finds image-specific parameters for the image segmentation in a completely unsupervised way.

Stain extraction

The BDGP data set used digoxigenin-labeled RNA probes, that were visualized by using color substrates NBT/BCIP, giving blue-colored gene expression stains to the embryo. Accordingly, blue information present in the RGB image is extracted to quantify the amount of gene expression.

The image has R, B, and G channels for red, blue and green respectively, scaled to lie between 0 and 1. Using the grey scale image ($y = \frac{(R+G+B)}{3}$) as the amount of stain captures the stain correctly, but noise due to illumination and texture variance is considerable. In images where the stain is present in small regions of the embryo, it is unable to identify a good contrast between the presence and absence of stain. Another possible technique to extract blue information is to subtract the grey scale color of the pixels from the blue channel (referred to as blue-grey in Figure 2.6). Thus, the stain is $s = max(0, B - y)$ where y is the grey scale image as defined earlier. Though the illumination effects are reduced by this technique, this approach is unable to extract highly stained portions of the image because dark blue stains have small (and equal) values for all three components of RGB.

Since the above solutions seem inadequate, we propose an approach that captures the correct staining in images with ubiquitous staining, and correctly identify the contrast between stain and no-stain in images where small regions of the embryo are stained:

$$geneExpression = \begin{cases} \max(s, 1 - B) & B < 0.5 \\ s & \text{otherwise} \end{cases}$$

It can be seen that $geneExpression$ is always positive, bounded between 0 and 1, and captures the amount of stain present (the higher the amount of stain, the higher the value of $geneExpression$). For visualizations in this paper, we use $(1 - geneExpression)$ (no longer mentioned explicitly later) so that darker regions have more stain. Sample results of extracting gene expression stain using various techniques are shown in Figure 2.6.

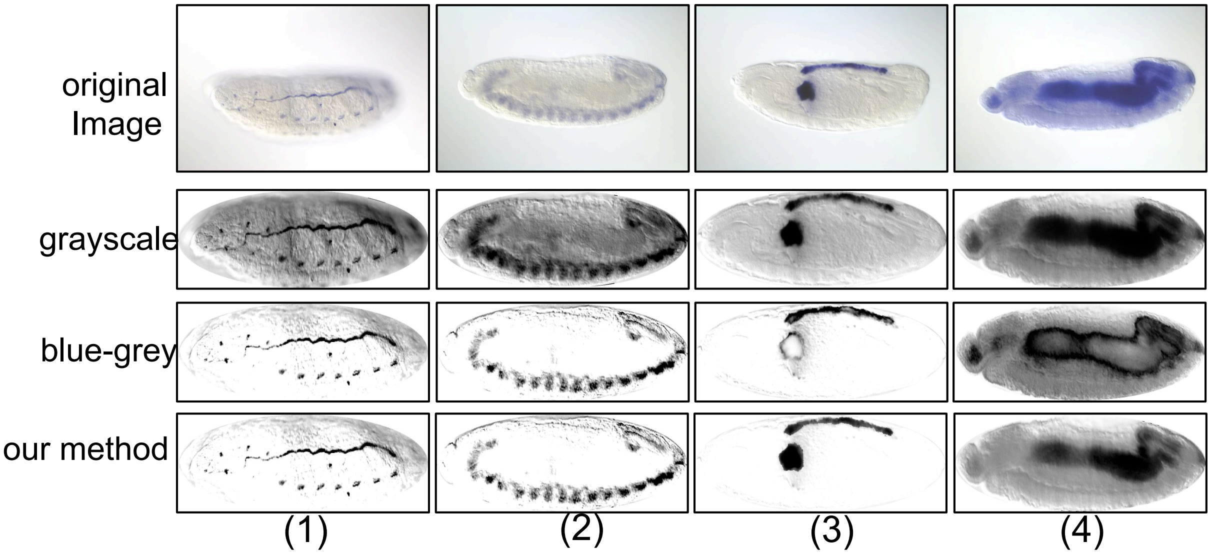


Figure 2.6. Gene expression extraction. For images (1) and (2), grey scale doesn't extract good results, while blue-grey gives good output. For images (3) and (4), grey scale does well, while blue-grey misses highly stained regions. In all images, our method performs at least as good as the best of the other two methods.

Gene expression segmentation with MRF

The expression stain found by preprocessing the image is a noisy measurement of the true expression value, distorted due to poor quality staining/washing, body texture, color condensation caused by body anatomy, embryo shape and contour, etc. Since the expression patterns are noisy with no sharp edges, standard edge-based segmentation algorithms are unable to find the correct stain pattern; adaptive thresholding methods also fail due to the presence of a large variance in the amount of staining in different images. Hence, we correct these issues by using a Markov Random Field (MRF)-based segmentation algorithm to remove noise from the expression pattern. Furthermore, given wide differences of expression patterns in different images, using a standard MRF with fixed parameters across images is hardly adaptive; therefore we fit image-specific MRFs in an unsupervised manner.

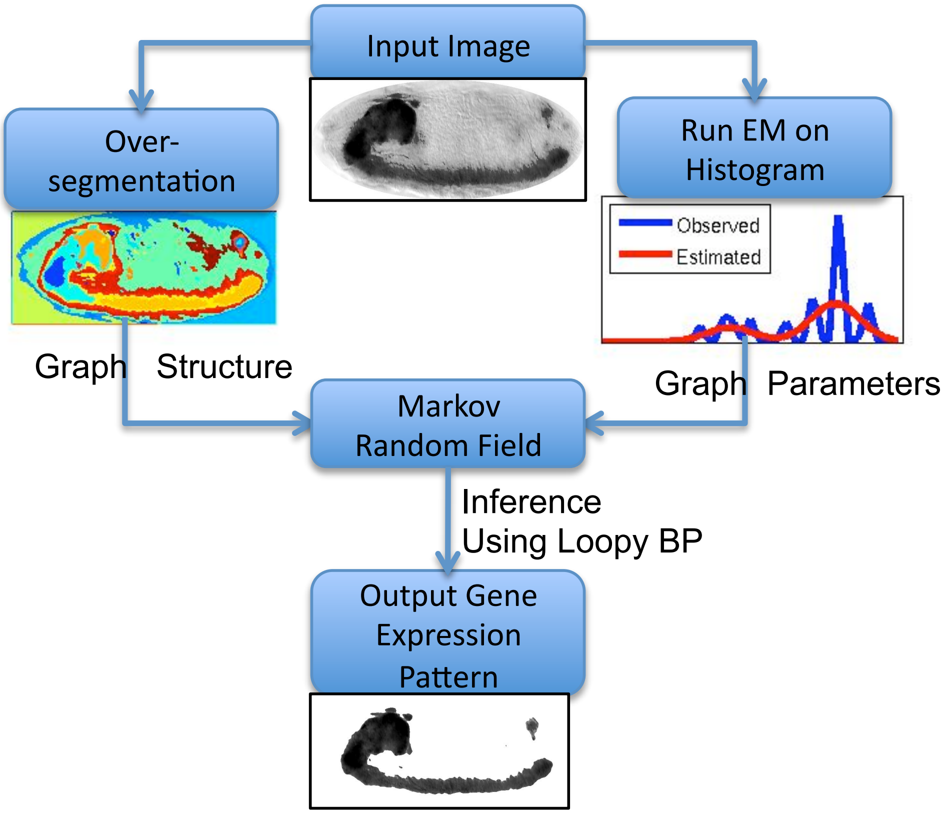


Figure 2.7. The gene expression pattern extraction process. The input image is first over-segmented, and the segments are converted into the MRF graph. The histogram of the image is analyzed using EM to find the MRF parameters. Loopy Belief Propagation is used for approximate inference to find the background pixels. Background pixels are noise, and their expression values are removed to get the gene expression pattern.

Building MRF structure Naively, for any image, each pixel can be treated as a single node in the MRF, and therefore the MRF naturally follows a grid structure. However, for large images, this technique generates very large graphical models, which are computationally infeasible. We define our image-specific MRF on "super-pixels" (Ren and Malik, 2003) instead, by first "over-segmenting" the image. A super-pixel is a collection of close-by pixels that have similar grey scale levels, and the same foreground/background label because our MRF assigns labels on super-pixels. Adjacent pixels whose values lie within $k * i$ and $k * (i + 1)$ for some integer i, are put in the same super-pixel. k is a thresholding parameter, which we set to 0.05. The MRF graph has each super-pixel corresponding to a nodal variable, and is connected to all its adjacent super-pixels, using 4-adjacency.

Let $X \equiv \{x_i\}_{i=1}^{S}$ denote the set of (binary) random variables representing class labels of super-pixels, and $Y \equiv \{y_i\}_{i=1}^{S}$ be the mean color values of super-pixels, where S is the total number of super-pixels in the image. The MRF defines the following distribution:

$$P(X,Y) = \frac{1}{Z} \prod_{i=1}^{S} \Phi(x_i, y_i) \prod_{(i,j) \in E} \Psi(x_i, x_j) \tag{2.1}$$

where Φ is the node potential, which captures the effect that pixel y_i has on the label of x_i; Ψ is the edge potential, which captures how the label of x_i is influenced by the labels of its neighbors, and E is the set of edges we found over the super-pixels.

The node potential $\Phi(x_i, y_i)$ is assumed to be Gaussian with parameters (μ_f, σ_f) if x_i is foreground, and (μ_b, σ_b) if x_i is background. The edge potential is defined as

$$\Psi(x_i, x_j) = \exp\left\{-\beta \times I\left(x_i \neq x_j\right)\right\}, \tag{2.2}$$

where I is an indicator function. Ψ defines the penalty given for neighboring pixels to disagree, i.e. one of the pixels is foreground and the other is background, and there is an edge connecting them. β captures the strength of the penalty, as β increases, we encourage smoother foreground assignments. We used $\beta = 2$, and found that it gave reasonably good performance.

Learning MRF parameters For the MRF defined above, the parameters $(\mu_f, \sigma_f, \mu_b, \sigma_b)$ must be defined for each image. Learning the MRF parameters for every image, by using classical unsupervised MRF learning techniques, is usually slow and inconvenient to process thousands of images.

We propose a simple heuristic to determine the graph parameters. If the penalty parameter β is

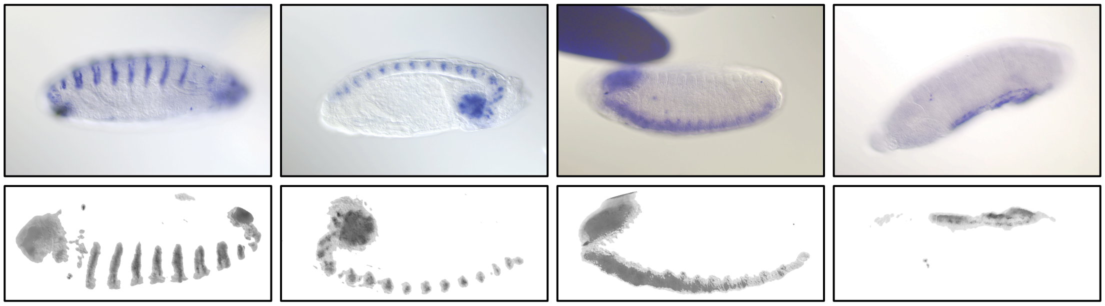


Figure 2.8. Gene Expression Pattern Extraction. The top row shows the original image, and the bottom row shows the extracted gene expression pattern at the end of our analysis. Note that, the embryo has been aligned to a standard shape before pattern extraction, and it may have been flipped by the orientation correction process.

zero, then the edge potentials are constant. The MRF then reduces to a mixture of Gaussians, where every super-pixel value is generated from one of two Gaussians, corresponding to the foreground and background respectively. The Gaussian parameters can then be learnt efficiently by computing the histogram of the image, and fitting a mixture of two Gaussians to the histogram using EM. To improve the smoothness of the estimates, we add a small uniform prior (1% of the mass of the histogram) to the image histogram before running EM. The parameters of the two Gaussians are then treated as approximations to the MRF parameters, i.e. μ_f, μ_b, σ_f, σ_b.

Loopy belief propagation for inference A standard approximate inference technique, loopy belief propagation (LBP), is used to find the maximum a posteriori (MAP) assignment to each x_i as foreground or background. Although LBP is not always guaranteed to converge, in our experiments, a small number (3-10) of iterations were sufficient for convergence, for all input images. At the end of this inference procedure, all background nodes are set to zero, and the foreground expression value is used as the final gene expression pattern obtained at the end of our image analysis pipeline. A small flowchart of our gene expression pattern extraction process is shown in Figure 2.7. Some examples of the gene expression patterns found by our MRF image segmentation algorithm are shown in Figure 2.8.

2.2.3 Feature extraction

Since all ISH images have been standardized to a standard shape, size, orientation, and position; and the gene expression pattern has been extracted, removing noise effects along the way, the feature representation needs to be position, orientation and scale dependent. The SIFT feature descriptor (Lowe, 1999) is used to derive a dense set of local visual features, using patches spaced regularly through the image, with a radius of 12 pixels (images are standardized to 128×320 pixels). Since the SIFT interest point detector is not used for finding features, the features found by this process are dependent on position, scale and orientation. Since this feature representation is high dimensional, we reduce dimensionality by using Singular Value Decomposition (SVD). A projection in 50 dimensional space was sufficient to capture most of the relevant information in these images, and gave good results.

2.3 Results

We apply SPEX^2 to the ISH images from the Berkeley Drosophila Genome Project (BDGP, 2005). Since our system performs automatic analysis for images in the lateral position, we picked 2689 images from the 13-16 time stage of the data set, which represent the expression patterns of 1432 genes. After automatic filtering of unqualified images in the standardization phase, 1904 images of 1011 genes entered the pattern extraction phase. We analyzed these expression patterns, and report results on two exemplary tasks: automatic annotation of images, and image clustering.

2.3.1 Image annotation

The expression patterns in BDGP Drosophila ISH images were annotated with anatomical and development ontology terms from a controlled vocabulary by human curators. Automatic annotation of images with terms from a controlled vocabulary represents a unique challenge itself. Since the main goal of SPEX^2 is to extract concise spatial expression patterns from ISH images for generic downstream applications of any user, rather than offering a perfect annotator, we will demonstrate the quality of the SPEX^2 output (e.g., expression features) using standard off-the-shelf annotation classifiers.

We focus on the 10 most frequent annotation terms in BDGP, and treat every term as an independent binary classification task. Each binary classifier is a standard SVM with a Gaussian kernel (we used libsvm (Chang and Lin, 2001) for our experiments). We use 10-fold cross-validation over a small set of values to pick the tuning parameter of SVMs - the cost of misclassification C.

We compare our results with two benchmark systems representing the state-of-the-art. In **System I**, we implement the feature extraction and classification procedure proposed by Zhou and Peng (2007). Their system extracts the embryo outline by using an adaptive thresholding method (Peng and Myers, 2004), and registers the embryo using affine transformation and intensity scaling. The anterior/posterior orientation is determined by maximizing total gene similarity across all images. Subsequently, two-dimensional wavelet embryo features are used, with min-redundancy max-relevance feature selection to pick the best features. Finally, binary classification on each annotation term is obtained via LDA (Linear Discriminant Analysis). In **System II**, Ji et al. (2009) used dense SIFT feature descriptors that are converted into sparse codes to form a codebook to represent their aligned images, and proposed an elegant local regularization (LR) procedure for multi-label learning. Details on how to obtain well-aligned images were not given, but the work by the same group in Ye et al. (2006) used a image standardization procedure outlined in Kumar et al. (2002), followed by histogram equalization for improved contrast in images. Hence, we use the above procedure when implementing this system, using the LR code from that group.

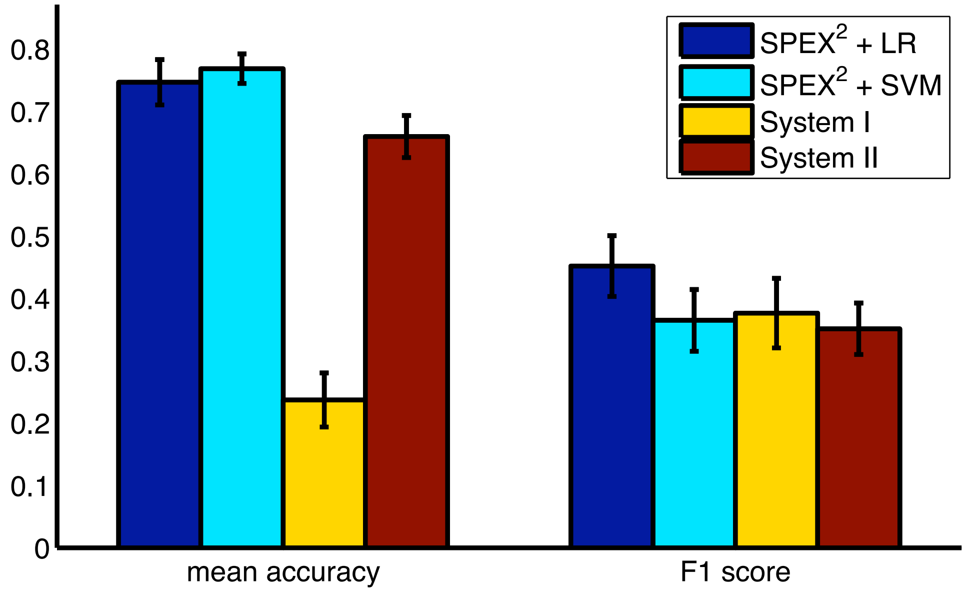


Figure 2.9. Image annotation. Mean Accuracy and F$_1$ using macro averaging, for predicting annotation terms.

We evaluate the performance using accuracy and F$_1$ score (Goutte and Gaussier, 2005). The F$_1$ score is the harmonic mean between the precision and recall of the results, and lies between 0 and 1, with higher F$_1$ representing better performance. Figure 2.9 shows the classification accuracy based on the SPEX^2 features, in comparison with the benchmarks. In terms of mean accuracy, SPEX^2 out-performs both the systems, while maintaining the same F$_1$ score. It is noteworthy that our result is obtained with a standard SVM, because our goal here is to demonstrate the quality of the SPEX^2 features, not that of the annotation algorithm. Indeed, we observe that using the sophisticated LR annotation algorithm of **System II** with our SPEX^2 features, increases our F$_1$ score, at the cost of a very small reduction in accuracy. Using the paired t test, the difference in accuracy between SPEX^2 with LR and **System II** was found significant with p-value=6.33e-6 and the difference in F$_1$ scores was significant with p-value=9.51e-5.

In addition, we visualize the information captured in the extracted expression patterns from SPEX^2 and the two systems we compare with, by computing the singular value decomposition (SVD) of the expression patterns (Pan et al., 2006). The set of eigen vectors can then be represented as images. We call these images eigen-expression patterns, like eigenfaces used in facial recognition (Pentland and Turk, 1991). The top 25 eigen-expression patterns are shown in Figure 2.10. Even though SVD performs global analysis of the feature space, the eigen-expression patterns produced by SPEX^2 seem to find localized regions of expression that correspond well to known gene expression patterns.

2.3.2 Gene expression clustering

Next we evaluate the SPEX^2 features on clustering, using a popular (but not necessarily optimal) clustering algorithm, the Spectral Clustering (SC). To avoid tuning parameters, we used self-tuning SC (Chen et al., 2008). Since the number of clusters must be specified in advance, and is hard to estimate for biological gene data, we tried different numbers of clusters from 5 to 100 (in steps of 5). We do most of our analysis on 15 clusters, the mean image of each cluster is shown in Figure 2.11. Visual inspection shows that the mean of each cluster has a distinctive pattern, each image looks salient enough to be a potential ISH image, even though it is the mean of tens to hundreds of images. This suggests that we have obtained high purity clusters. Details of the content of each cluster (i.e., represented by 10 images therein) are available in the supplemental material, which substantiate the above assessment.

The literature on clustering specifies a variety of evaluation measures, however all of them are distance-based and not biologically intuitive. In this specific data set, we observe that we have two external sources of information associated with each image (that are not used by the clustering algorithm), which can help build an intuition of what good clusters should look like. The first source of information is the manual curation of these images, which has annotated each gene pattern with terms from a controlled vocabulary describing the localization of the expression pattern. The second source is the GO functional annotations, associated with the gene. We conduct enrichment analysis using both sets of information.

Hypothesis test for enrichment

Given a single cluster, and a single annotation term (from controlled vocabulary or GO ontology), a p-value can be obtained by using an exact hypergeometric test. However, since we test each cluster for multiple annotations, a correction for multiple hypothesis is needed. Standard corrections for multiple hypothesis testing are usually found to be either very conservative, or having low power. We instead convert the p-values into q-values, that control the positive false discovery rate (pFDR), by using the procedure described by Storey (2002). The pFDR is the expected proportion of erroneous rejections among all rejections, thus a pFDR value of 5% means that 5% of predicted significant features will be truly null. The q-value measures the strength of the observed statistic, with respect to pFDR, and automatically corrects for multiple hypothesis testing, it is therefore a much more powerful test scheme.

We conduct enrichment analysis using the procedure outlined by Arava et al. (2003), which allows us to estimate q-values for multiple hypothesis tests, even when the statistics being measured are correlated (as is the case for GO and pattern annotations).

Annotation terms enrichment

If the data is well clustered, then a single cluster of images must be enriched for specific annotation terms that the images have been annotated with. Table 2.1 shows a partial enrichment analysis for 15 clusters. All clusters were significantly enriched for at least one term, with a total of 90 enriched terms. Since the number of terms is higher than the number of clusters, each cluster is enriched for a combination of multiple terms. For example, cluster one with 149 images is enriched for images that have been annotated with embryonic brain and central nervous system, while cluster three with 100 images is enriched for a combination of embryonic brain with embryonic midgut and ventral nerve cord. Images annotated with only ventral nerve cord have been clustered into a separate cluster (having 139 images).

To assess the advantage of concise expression information extracted by SPEX^2 over benchmark systems, we performed the same clustering analysis based on features generated by the two systems discussed above. We counted the number of clusters therefrom that have at least one significant annotation at qvalue=0.05. Figure 2.12 shows the number of significant clusters found by the three methods, as we vary the number of clusters from 5 to 100. We observe that SPEX^2 works better than the other two methods, with an average of 18.39% more significant clusters obtained than its closest competitor **System I**.

GO functional enrichment

It is believed that similar spatial-temporal patterns of gene expression are related to similar functionality. Hence, we might expect that a good clustering will be enriched for gene functions, as defined by the GO ontology. Since we are analyzing data from stage 13-16 of Drosophila embryonic development, its not clear that the spatial expression information in this brief period is enough for gene functionality enrichment. Hence, we do a limited functional enrichment analysis of our cluster results, and leave extended analysis across time-stages for future work.

Since we are analyzing spatial patterns of genes that are differentially expressed in the embryonic stage, without any analysis across time, we expect to find enrichment of smaller, more precise functional annotations that are related to specific areas of embryonic development, and GO Slim is not appropriate. For our enrichment analysis, we used GO annotations that are present in at least 5 genes in our data set.

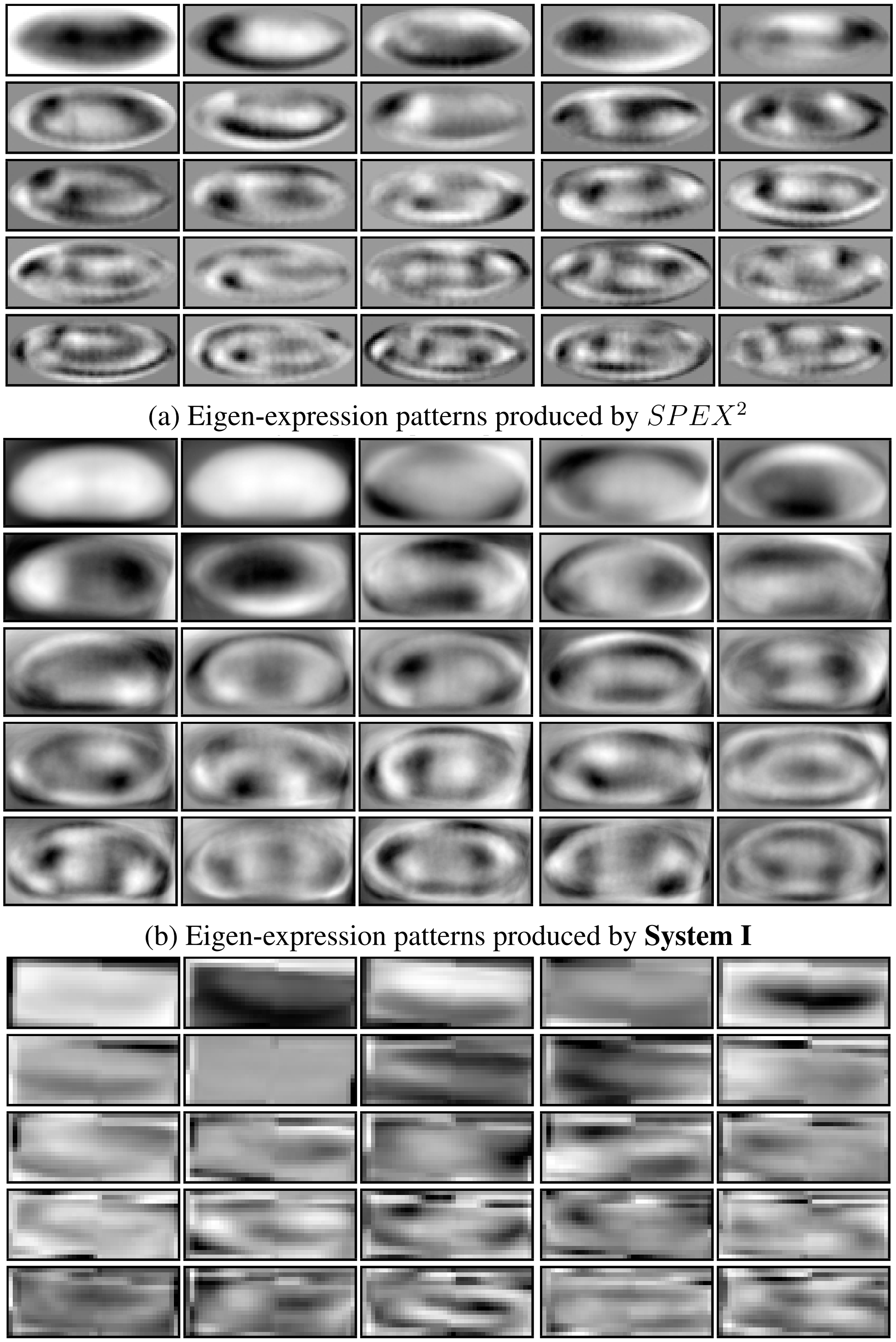
Table 2.2 shows a part of the enrichment analysis performed on 15 clusters. We observe that 9 out of the 15 clusters are significantly enriched (q-value=0.05) for various GO ontology functions, many of which are known to be explicitly relevant to Drosophila development. For example, 8 of the 12 genes related to myoblast fusion are found in a single cluster. Genes for the myoblast fusion are known to be expressed early in development, in embryos 0-4h after egg laying, and remain high during embryogenesis (but not in the larval stage) (Dworak and Sink, 2002). Additionally, it is known that during Drosophila embryogenesis, the development of the open tracheal system can be observed on the dorsal side; 18 of the 43 genes related to open tracheal system development are found in a single cluster.

All ten genes related to “progression of morphogenetic furrow during compound eye morphogenesis” are found in the same cluster, and 5 of the 9 genes related to segment specification, are also clustered together. Additionally, all genes related to “larval central nervous system remodeling” are found in a single cluster, and 5 of the 6 genes related to “bristle morphegenesis” are also co-clustered. This seems to imply that genes involved in larval stage development are already showing spatial coherence in the embryonic stage.

Thus, the SPEX^2 clusters are able to capture fine-grained GO functional annotations. In contrast, clustering using features extracted by **System I** found only 6 significant clusters out of 15. Our method thus improves the number of significantly enriched clusters by 50%. **System II** returned only 1 significantly enriched cluster out of 15, at q-value=0.05.

2.4 Sample cluster images

We present details of the content of each cluster, by representing each cluster by 10 representative images, as shown in Figures 2.13-2.26 below. As can be clearly seen, each cluster visually captures specific expression patterns. For example, cluster 1 consists of genes including CG11426, sna, mira, shep, NetB, CG1434, and others, and clearly shows patterning in ventral epidermis and nerve cord. Similarly, cluster 13 with genes like scarface, RhoBTB, CG1942, prc, CG5171 and Oat show patterning in the yolk nuclei, and amnioserosa.

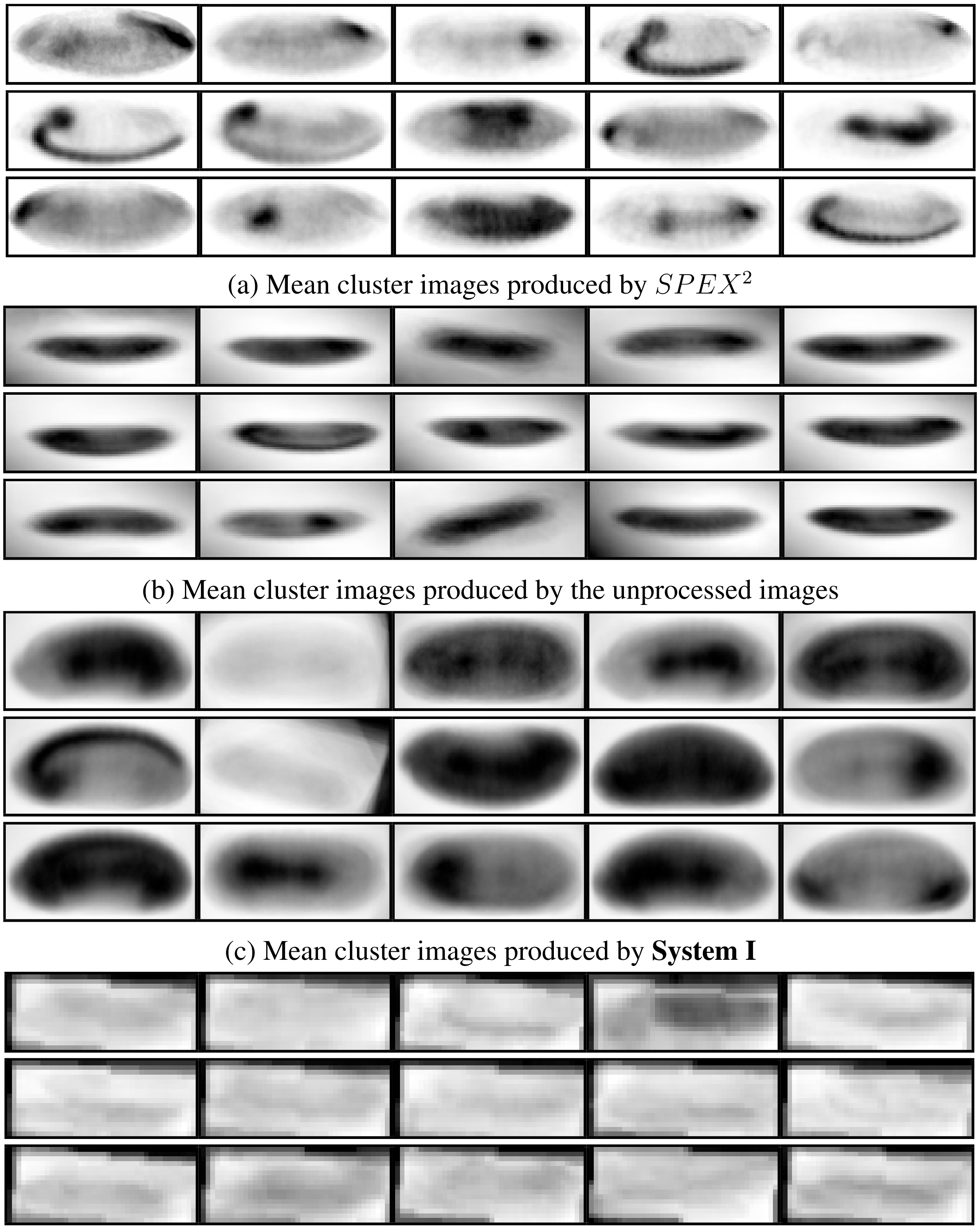


(a) Eigen-expression patterns produced by $SPEX^2$

(b) Eigen-expression patterns produced by **System I**

(c) Eigen-expression patterns produced by **System II**

Figure 2.10. Eigen-expression patterns used for low dimensional feature representation. The eigen-expression patterns produced by $SPEX^2$ show local pattern coherence and better capture spatial patterns observed in the data, which the other two methods are unable to capture.



(a) Mean cluster images produced by $SPEX^2$

(b) Mean cluster images produced by the unprocessed images

(c) Mean cluster images produced by **System I**

(d) Mean cluster images produced by **System II**

Figure 2.11. Mean image in gene clustering. Each image is the mean of a single cluster found by using processed images from different systems. The intensity of any pixel in the mean image is the average intensity of that pixel in all images assigned to this cluster. As can be seen, clustering using unprocessed images only finds clusters based on embryo position and illumination. The clusters produced by $SPEX^2$ have very low noise, and visually look pure in terms of patterns clustered.

Cluster Size	Term Annotation	Annotation Probability	Overlap	q-value
149	embryonic brain	0.298	133	5.14e-17
	embryonic central nervous system	0.117	49	9.27e-30
194	embryonic midgut	0.282	109	1.84e-20
	embryonic/larval muscle system	0.150	67	3.27e-13
	embryonic Malpighian tubule	0.074	41	6.59e-12
	embryonic anal pad	0.122	49	1.30e-5
	embryonic gastric caecum	0.028	23	5.61e-7
	dorsal prothoracic pharyngeal muscle	0.103	47	2.72e-15
100	embryonic midgut	0.282	56	1.03e-6
	embryonic brain	0.298	67	3.98e-13
	ventral nerve cord	0.327	68	6.64e-11
	ventral sensory complex primordium	0.084	23	3.95e-4
139	ventral nerve cord	0.327	75	5.80e-6
39	embryonic central brain pars intercerebralis	0.0094	5	6.94e-3
110	amnioserosa	0.01577	13	1.12e-6
140	embryonic esophagus	0.0678	27	4.11e-5
	embryonic hypopharynx	0.168	51	5.07e-6
	embryonic proventriculus	0.121	40	1.50e-5
165	embryonic/larval muscle system	0.15	74	3.27e-12
	dorsal prothoracic pharyngeal muscle	0.103	53	1.65e-17
168	yolk nuclei	0.073	64	2.78e-31
	gonadal sheath	0.0007	7	2.43e-2
78	embryonic brain	0.298	47	5.11e-6
	ventral nerve cord	0.327	56	7.31e-8
70	embryonic hypopharynx	0.168	28	2.77e-4
	labral sensory complex	0.009	7	2.77e-4
	embryonic maxillary sensory complex	0.0205	10	2.68e-4
128	embryonic salivary gland body	0.021	12	2.482e-3
96	embryonic large intestine	0.035	13	7.120e-3
163	embryonic/larval somatic muscle	0.070	31	6.06e-5
	dorsal prothoracic pharyngeal muscle	0.103	37	3.43e-4
93	ventral nerve cord	0.327	51	5.491e-3

Table 2.1. Enrichment Analysis for 15 clusters, using terms from the controlled vocabulary. The first column shows the size of the cluster, the next 2 columns show the term annotation, and the probability that a given gene will be annotated with this term. The 4th column gives the number of images in this cluster annotated with this term, with the last column giving the q-value of the overlap.

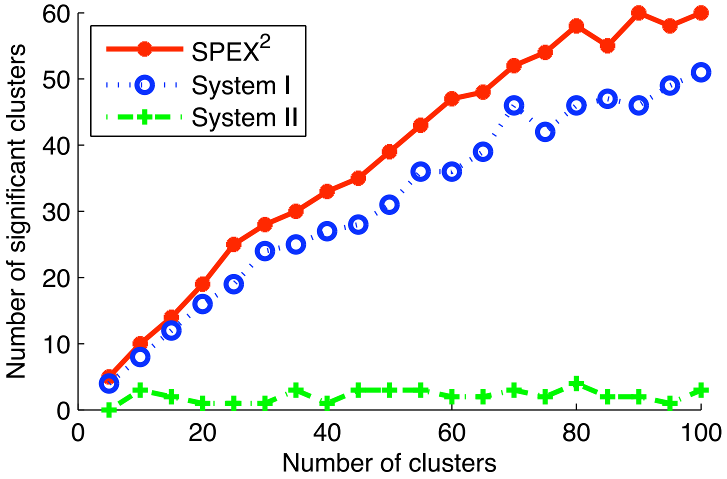


Figure 2.12. Significantly enriched clusters v/s total # of clusters (q-value=0.05)

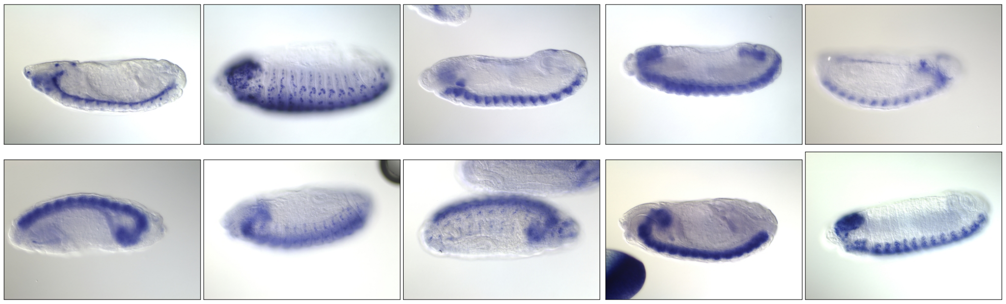


Figure 2.13. Sample images from cluster 1

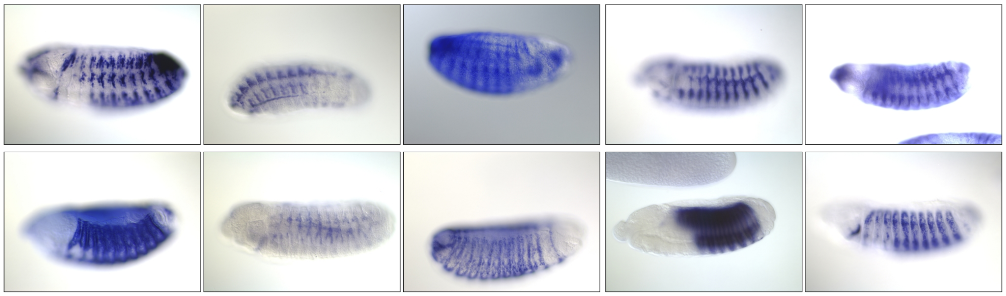


Figure 2.14. Sample images from cluster 2

Cluster Size	GO Category	GO Function	GO Category Size	Overlap	q-value
149	GO:0007520	myoblast fusion	12	8	0.00539
187	GO:0007424	open tracheal system development	43	18	0.011601
	GO:0008354	germ cell migration	8	5	0.081374
	GO:0035017	cuticle pattern formation	8	5	0.081374
126	GO:0008407	bristle morphogenesis	6	5	0.005878
102	GO:0035193	larval central nervous system remodeling	10	10	0.0010015
	GO:0006914	autophagy	10	10	0.0010015
	GO:0007350	blastoderm segmentation	9	5	0.046871
	GO:0007379	segment specification	9	5	0.046871
	GO:0007458	progression of morphogenetic furrow during compound eye morphogenesis	10	10	0.0010015
	GO:0007552	metamorphosis	17	10	0.068039
	GO:0007562	eclosion	10	10	0.0010015
	GO:0048808	male genitalia morphogenesis	10	10	0.0010015
174	GO:0005730	nucleolus	11	8	0.00961
116	GO:0017150	tRNA dihydrouridine synthase activity	5	4	0.021049
	GO:0003725	double-stranded RNA binding	5	4	0.021049
	GO:0003777	microtubule motor activity	9	6	0.006836
	GO:0005873	plus-end kinesin complex	5	4	0.021049
	GO:0016323	basolateral plasma membrane	8	8	0.044737
94	GO:0004866	endopeptidase inhibitor activity	5	4	0.021049
68	GO:0004497	monooxygenase activity	12	6	0.028244
44	GO:0006508	proteolysis	48	8	0.009222

Table 2.2. Enrichment Analysis for 15 clusters, using GO functional annotations. The first column shows the size of each cluster, the next 3 columns show the GO category, function, and number of genes in the dataset having that GO function. The fifth column gives the number of genes with the particular GO function present in this cluster, and the last column gives the q-value of the overlap.

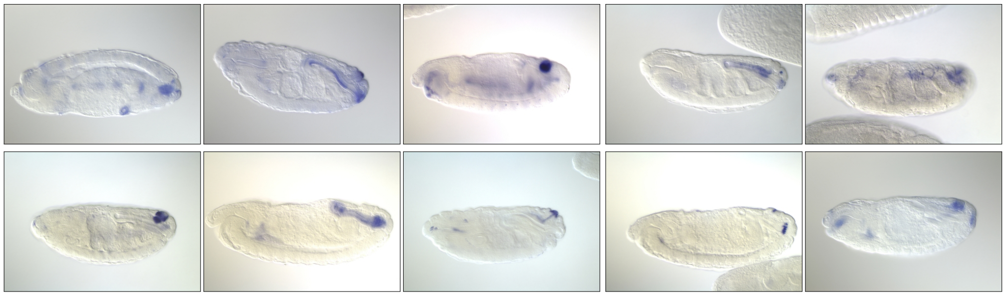


Figure 2.15. Sample images from cluster 3

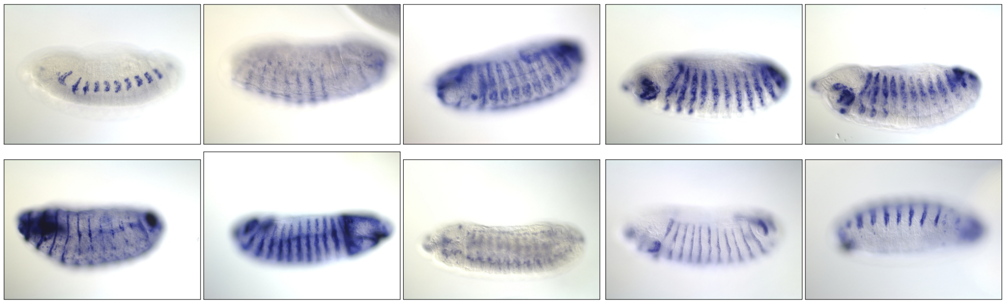


Figure 2.16. Sample images from cluster 4

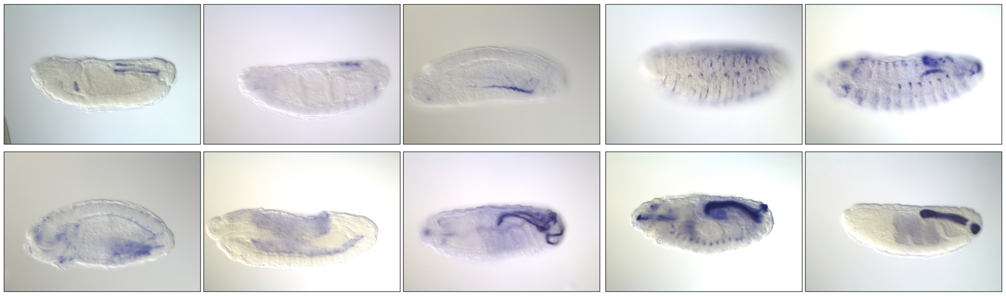


Figure 2.17. Sample images from cluster 5

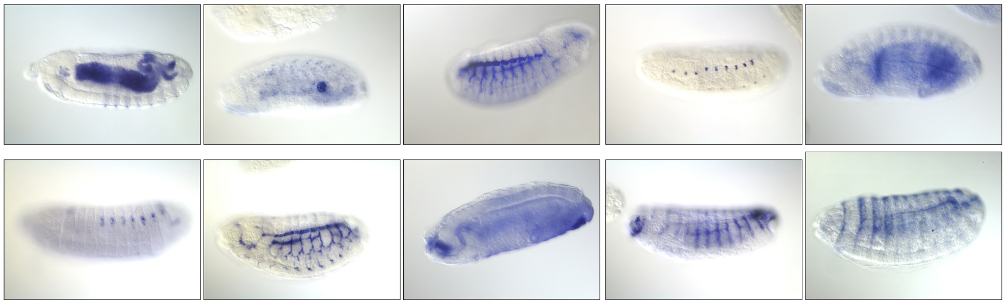


Figure 2.18. Sample images from cluster 6

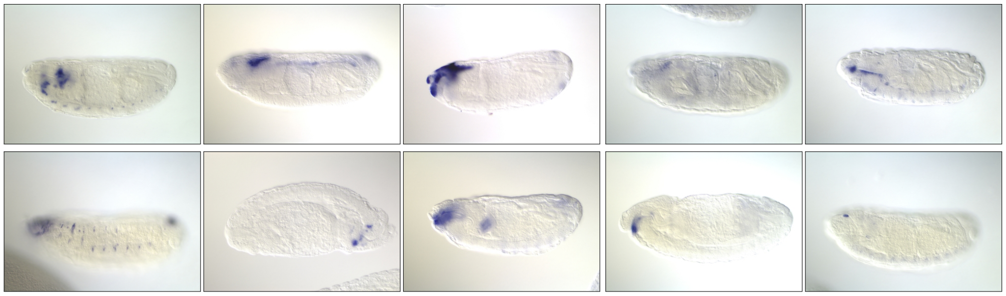


Figure 2.19. Sample images from cluster 7

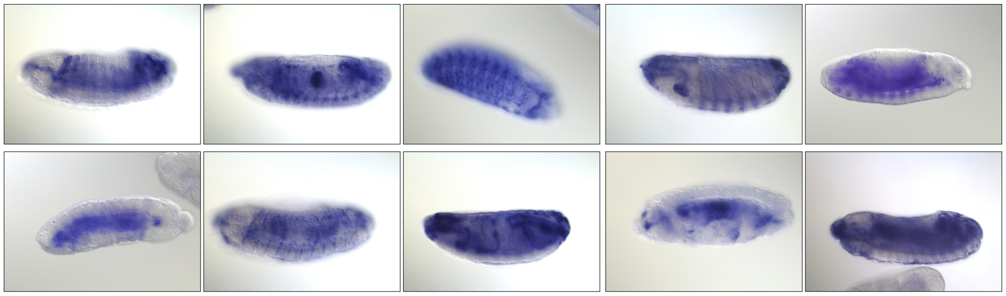


Figure 2.20. Sample images from cluster 8

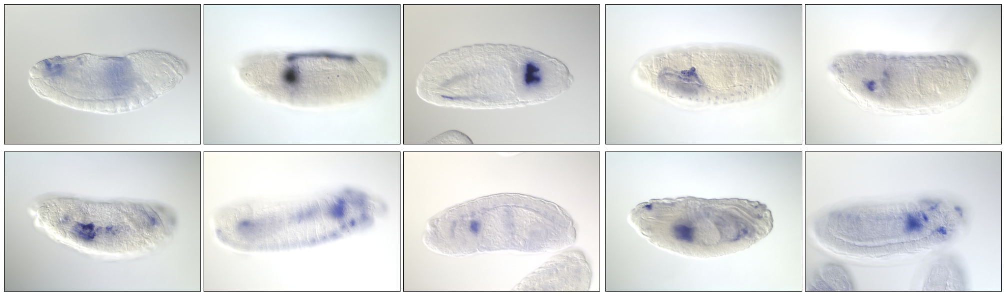


Figure 2.21. Sample images from cluster 9

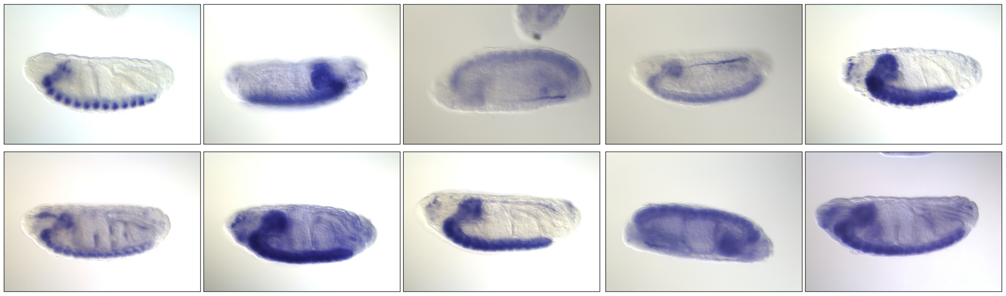


Figure 2.22. Sample images from cluster 10

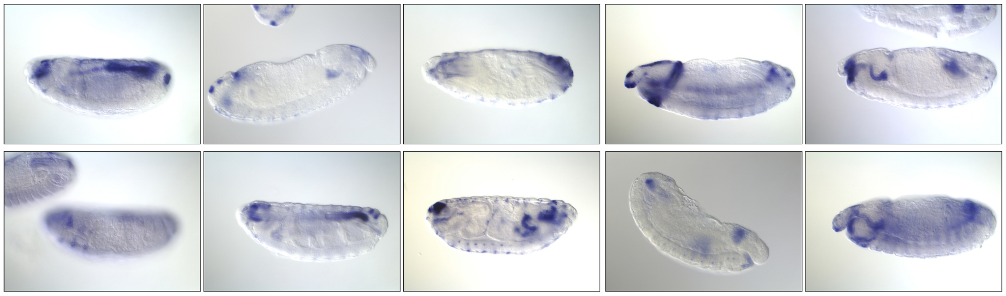


Figure 2.23. Sample images from cluster 11

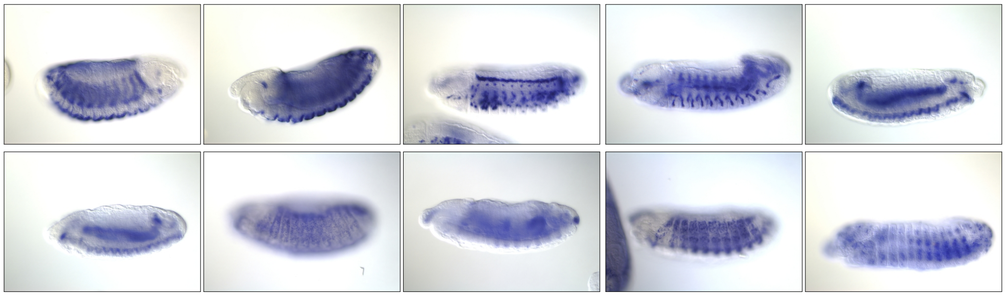


Figure 2.24. Sample images from cluster 12

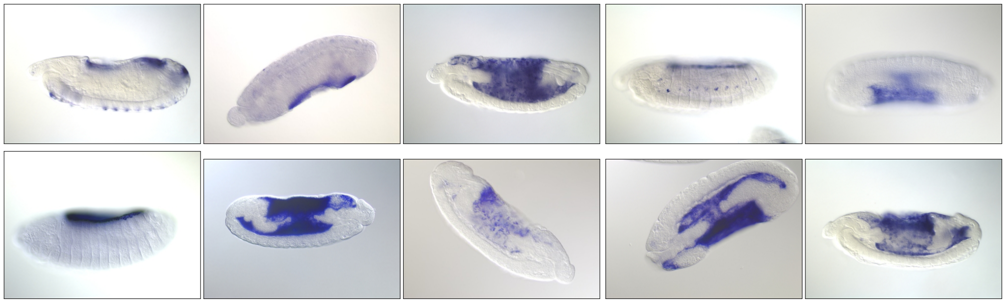


Figure 2.25. Sample images from cluster 13

2.5 Discussion

SPEX^2 represents the first step towards automatic functional analysis of ISH images of Drosophila embryos, namely concise extraction of spatial gene expression patterns. Our extraction system employs a pipeline of analytical techniques to first standardize the embryo via embryo outline extraction, orientation detection and correction, and registration; and then extracts spatial expression signal via filters and probabilistic segmenters. Finally, it converts the spatial signals

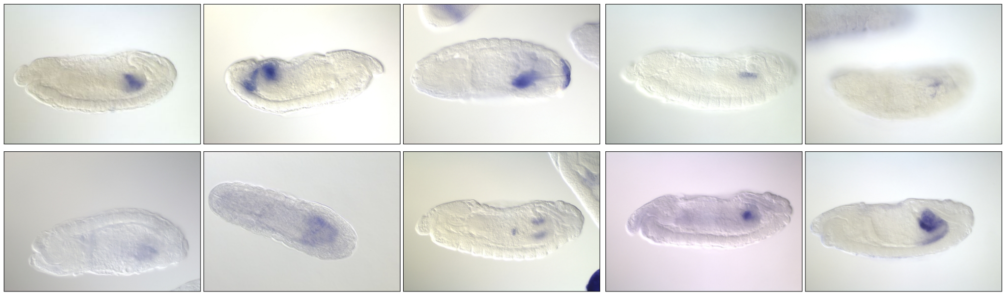


Figure 2.26. Sample images from cluster 14

into a low dimensional feature representation, suitable for advanced analysis. We evaluated our system by using the resultant features for automatic pattern annotation and clustering. Using simple classification techniques and our sophisticated feature extraction pipeline, we achieved a significant improvement in annotation accuracy over existing systems. We also clustered the Drosophila ISH images, and conducted enrichment analysis on both pattern term annotations, and GO functional annotations. We found significant enrichment in both scenarios. The next step is a more detailed analysis of ISH images using this feature representation, which we will see in the next chapter.

Acknowledgements

This work was done in collaboration with Prof. Christos Faloutsos. Fan Guo and Lei Li were also involved in helpful discussions during the initial processing of the BDGP images.

Chapter 3

GINI: From ISH images to gene interaction networks

This chapter discusses GINI (Puniyani and Xing, 2012), a machine learning system for inferring gene interaction networks from Drosophila embryonic ISH images. GINI builds on the vector-space representation of the spatial pattern of gene expression in ISH images, enabled by the SPEX^2 system; and a new multi-instance-kernel algorithm that learns a sparse Markov network model, in which, every gene (i.e., node) in the network is represented by a vector-valued spatial pattern rather than a scalar-valued gene intensity as in conventional approaches such as a Gaussian graphical model. By capturing the notion of spatial similarity of gene expression, and at the same time properly taking into account the presence of multiple images per gene via multi-instance kernels, GINI infers statistically sound, and biologically meaningful gene interaction networks from image data. Using both synthetic data and a small manually curated data set, we demonstrate the effectiveness of our approach in network building. Furthermore, we report results on a large publicly available collection of Drosophila embryonic ISH images from the Berkeley Drosophila Genome Project, where GINI makes novel and interesting predictions of gene interactions.

3.1 Introduction

In this chapter, we propose a machine learning system for image-based network estimation to infer gene interaction networks from spatial similarity of gene expressions captured via ISH images. The system is called GINI (Gene Interaction Network from Images). With such a system, we were able to systematically perform a genome scale network learning and analysis of the image data from BDGP from a single time point.

To solve this problem, we recognized the following main challenges that are unique to micro-imaging data versus the classical microarray data, which must be properly addressed before a genome-scale gene network can be derived from such data.

Representation and quantification of gene activities: Unlike microarrays, which represent gene activity with a univariate state or magnitude, images provide high-dimensional information for every gene, and it remains an open problem in computer vision research to extract meaningful features from the ISH images that are suitable for comparing activities of different genes and other genome-wide analysis (Frise et al., 2010; Puniyani et al., 2010).

Multi-variate measurement: Even after one can standardize the imagery-records of the expression of a gene at a particular time point by a d-dimensional vector, where d are the number of features extracted from the image, a proper metric must be defined to quantify distances between them.

Condition alignment: Images for different genes are typically taken under non-identical conditions (e.g., time, temperature, etc.), whereas a microarray is a snapshot of multiple genes under the same condition. This affects how signals are normalized across genes before they can be compared.

Sample imbalance: Different genes typically have different number of image records, i.e., for gene i and j, their corresponding measurements can be in the form of two bags of different sizes. It is not clear how to define distance or correlation between bags of images of different sizes. One simple solution to this problem is to randomly sample a single image from each gene. However, throwing away images fails to capture the natural variation observed in gene expression patterns for some genes. Further, if noise in the expression patterns has not been removed correctly in the feature extraction step, leveraging the existence of multiple images per gene can lead to reduced noise, and improved performance.

Sparsity and statistical interpretability: The interaction network proposed must be sparse and statistically meaningful, since we expect that a small fraction of all possible interactions are actually present in a single organism, and such interactions must reveal globally consistent conditional-independence relationships between genes, which is not possible in a simple pairwise-correlation graph.

There has been some earlier work on automatic annotation of ISH images with annotation terms (Mace and Ohler, 2010; Yuan et al., 2012), clustering of gene expressions (Tomancak et al., 2007), determination of the development stage of embryos (Gargesha et al., 2005), etc., some of which have been applied on the BDGP dataset. In this paper, we propose a machine learning system to infer gene interaction networks from spatial similarity of gene expressions captured via ISH images. The system is called GINI, for Gene Interaction Network from Images. With such a system, we were able to address satisfactorily the challenges mentioned above, and systematically performed a genome-scale network learning and analysis on the BDGP dataset.

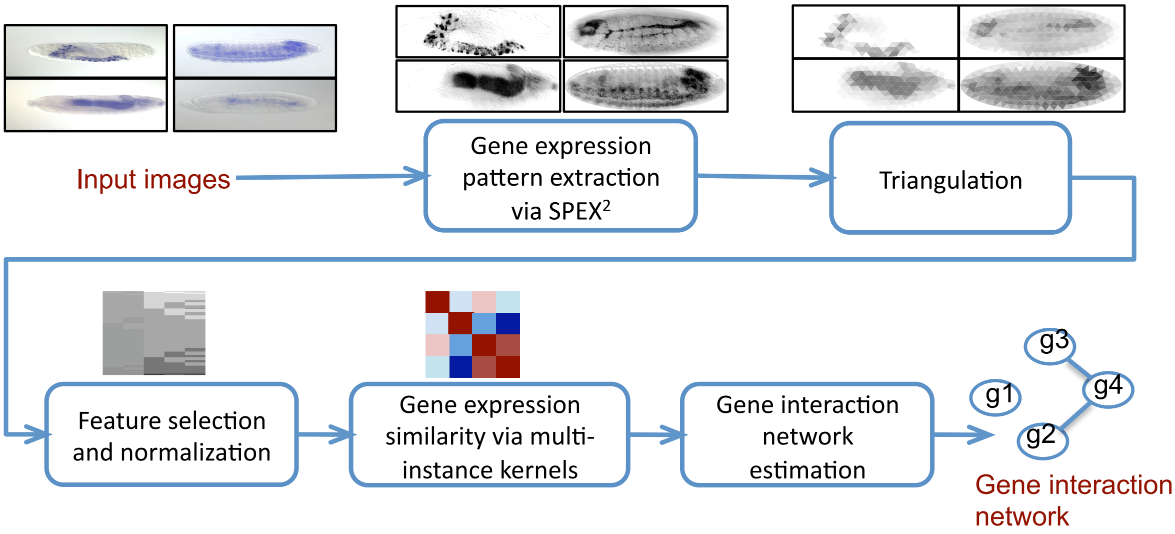

Input images
Gene expression pattern extraction via SPEX^2
Triangulation
Feature selection and normalization
Gene expression similarity via multi-instance kernels
Gene interaction network estimation
g1
g2
g3
g4
Gene interaction network

Figure 3.1. GINI schematic. The schematic shows an outline of the overall system to reverse engineer gene networks from ISH data. Sample output of each step is shown on top of the box corresponding to that step.

3.1.1 Overview of the GINI approach

GINI first extracts the gene expression pattern from each image using a computer version driven image analysis pipeline SPEX^2 (Puniyani et al., 2010). These expression patterns are spatially aligned and normalized to enable spatial comparison of gene expression across multiple images. Next, the expression patterns are represented by suitable standardized features through a process called "triangulation", followed by feature normalization and selection. Since each gene may have a different number of images in the data, each gene can now be represented by a "bag" or a set of feature vectors - one feature vector per image. Thus, our task is to estimate the gene network, given bags of images per gene (Figure **??**). We cast the problem of estimating a gene interaction network as the task of estimating the graph structure $\mathcal{G}$ of a Markov random field (MRF) over the genes. The underlying graph encodes conditional independence assumptions between the genes, that is, two genes are said to not interact in the network if their gene expressions are conditionally independent of each other, conditioned on the expression of all other genes in the network. We employ multi-instance kernel technique using different order statistics to compute similarity between bags of images. We then estimate a sparse network of gene interactions by modeling the data as a multi-variate multi-attribute Gaussian, and estimating the sparse inverse covariance matrix of the model. A schematic diagram of the system pipeline can be seen in Figure 3.1.

3.1.2 Contributions

GINI is a bioimage informatics system based on a computer vision pipeline for ISH micro-image processing and a statistical learning algorithm for network inference. The main contributions of this work are summarized below.

First, the image analytic pipeline used by GINI offers a rigorous and universal approach to extract a standardized representation of spatial patterns of gene expressions. Comparing to the popular SIFT features (Yuan et al., 2012), which is based on detecting interest points with heavy assumptions on object shape, texture, and other physical properties originally meant for natural objects, our approach is more suitable for ISH staining in Drosophila embryos which do not resemble natural objects and require preservation of overall spatial shape and overall intensity information in a canonical way for intra-gene normalization and inter-gene comparison.

Second, GINI infers a network that enjoys the Markov network property: it gives globally consistent conditional-independency interpretation for every edge, and therefore is biologically more meaningful. It is known that marginal correlation (as often used in estimating an *ad hoc* network), which is computed for every gene-pair in isolation (i.e., ignoring all other genes in the system), confounds direct and indirect dependencies, and could result in a clique-like dense graph or subgraph among genes that are not directly dependent, but have a long-distance interaction. Studying conditional independencies in a network allows us to predict interactions between a pair of genes in the context of other genes, allowing a distinction to be made between direct and indirect relationships between the genes, and reducing false positives.

Third, our formulation based on Gaussian Markov random field and multi-instance kernel for the GINI network is convex, hence the globally optimal estimator of the network is computed, no approximations are involved. Furthermore, under suitable conditions, our graphical model learning algorithm is sparsistent, i.e. as the amount of available data increases, the algorithm is statistically guaranteed to predict the correct interactions between the genes. While Bach and Jordan (2002) have previously proposed learning the structure of graphical models from data using Mercer kernels, their approach is based on a non-convex local greedy search to find edges in the graph. Our approach represents the first work that uses Mercer kernels and Gaussian Graphical Models to predict kernelized graphical models using a convex formulation.

Finally, with the GINI system, we were able to systematically perform a genome-scale network learning and analysis of the genes expressed during 2 time points of Drosophila embryogenesis recorded by ISH imaging from BDGP (Tomancak et al., 2002b). In both time points, we find that the GINI networks are modular and scale free, which are properties predicted to hold true in gene interaction networks. Further, different regions of the networks are enriched for spatial annotations, thus GINI is able to cluster spatially similar genes. The hubs of the networks, i.e., the genes with the largest number of predicted interactions are functionally enriched for important cellular functions. We demonstrate that the networks predicted by analyzing microarray data does not have either spatial or functional enrichment, thus these results could not have been obtained by analyzing microarray data.

To the best of our knowledge, GINI represents one of the first efforts to reverse engineering gene networks from ISH image data. In both extensive simulation studies and empirical biological analysis, we demonstrate the effectiveness of GINI in predicting networks, and show that the statistical assumptions behind GINI are reasonable, and the biological analysis enabled by GINI merits close examination and further exploration.

3.2 Methods

We begin by introducing the key algorithmic innovations needed to compute the gene network from the ISH images, assuming that each gene has a bag of images, with the images processed to be represented by informative and canonical feature vectors. This is followed by a discussion on the image processing procedures needed to extract informative features from the images.

3.2.1 Network inference from "one image per gene" ISH data

We first show how GINI estimates a gene network, when each gene has only one image. The next subsection extends the GINI algorithm to deal with multiple images per gene.

Let G denote the set of n genes being studied, so that g_i is the i^{th} gene, where $i \in \{1, \cdots, n\}$, and d is the number of features extracted per image. Each feature represents the gene expression in a spatial location of the embryo.

Note that algorithms that analyze microarray data typically treat samples drawn from different time points as independent samples (Jaffrezic and Tosser-Klopp, 2009), even though expressions of the same gene across time is expected to be auto correlated. We similarly assume that the different spatial features are independent of each other. The spatial independence assumption has also been implicitly made by (Janssens and Reinitz, 2006; Segal et al., 2008) while modeling transcription networks in Drosophila embryos. In the results section, we use simulated data to demonstrate that this assumption does not affect the accuracy of the algorithm significantly.

By modeling the gene interactions as invariant across the spatial locations in the embryo, we can assume that each feature is independently and identically drawn (i.i.d.) from the same distribution. Inferring gene interactions is then equivalent to modeling the dependence between the expression values of different genes at the same spatial location. Expression of the n genes in each spatial location is assumed to be drawn from some (multi-variate) distribution, independent of all other spatial locations. Each spatial feature $X^{(k)}$ ($k \in \{1, \cdots, d\}$) may be modeled as a vector of length n, with $X^{(k)}(i)$ capturing the expression value of the i^{th} gene in this location k. This gives us d independent samples with which the parameters of the underlying distribution may be learned. Formally, let each spatial location be drawn independently from a multi-variate Gaussian $\mathcal{N}(\mu, \mathbf{\Sigma}_{n\times n})$, where μ is the mean vector, and $\mathbf{\Sigma}_{n\times n}$ is the positive semi-definite covariance matrix between the genes.

In a multivariate Gaussian distribution, the $(i, j)^{th}$ entry of the inverse covariance matrix $\mathbf{\Sigma^{-1}}$ is zero if and only if the corresponding genes are conditionally independent given the rest of the graph. Thus, the non-zero entries of the inverse covariance matrix correspond to edges in the corresponding Gaussian Markov random field, giving rise to the gene interaction network. The Gaussian Markov random field is also known as a Gaussian graphical model (GGM) (Dempster, 1972). Since we expect a small number of interactions per gene, the estimated graph must be sparse, i.e. the number of non-zero entries of the inverse covariance matrix must be small.

Thus, the gene interaction network may be estimated by learning a Gaussian distribution from the observed images, such that the inverse covariance matrix is sparse. The mean μ of the Gaussian is estimated by the observed sample mean,

$$\mu = \frac{1}{d}\sum_{k=1}^{d} X^{(k)} \tag{3.1}$$

Then, the inverse covariance matrix $\widehat{\mathbf{\Sigma}}^{\mathbf{-1}}$ can be estimated by minimizing the negative log-likelihood of the data, over all possible positive semi-definite matrices. To enforce sparsity, the L_0 norm of $\mathbf{\Sigma^{-1}}$, which counts the number of non-zero elements, is added to the negative log likelihood. Since optimizing the L_0 norm is non-convex and NP hard, the L_1 norm is used as a convex relaxation to the L_0 norm. The L_1 norm of a matrix is the sum of the absolute values of the elements of the matrix, and also enforces sparsity in the solution. Adding the L_1 norm regularization also ensures that the minimizer of the objective function exists, and is well defined. Thus, our objective function is

$$\widehat{\mathbf{\Sigma}}^{\mathbf{-1}} = \arg\min_{\mathbf{\Theta}\succeq 0} \left\{ trace(\mathbf{S\Theta}) - \log\det\mathbf{\Theta} + \lambda||\mathbf{\Theta}||_1 \right\} \tag{3.2}$$

where $\mathbf{S}$ is the second moment matrix about the mean

$$\mathbf{S} = \frac{1}{d}\sum_{k=1}^{d}(X^{(k)} - \mu)(X^{(k)} - \mu)^T \tag{3.3}$$

λ is a tuning parameter, by which we determine the strength of the penalty. As we increase the value of λ, we increase the penalty on the absolute values of $\mathbf{\Theta}$, and hence, the graph induced by $\widehat{\mathbf{\Sigma}}^{\mathbf{-1}}$ becomes more sparse. The edges in the graphical model are then estimated as

$$\mathcal{E} = \left\{ \ (i,j) \mid \quad \widehat{\mathbf{\Sigma}}^{\mathbf{-1}}(i,j) \neq 0; \quad i \neq j \ \right\} \tag{3.4}$$

Optimization

The objective function defined in Equation 3.2 is convex, hence it can be solved by any convex optimization algorithm. Banerjee et al. (2006a) formulated an $O(n^4)$ block coordinate descent method to solve it, where n is the number of dimensions. Friedman et al. (2007) formulated each step of the block coordinate descent as a Lasso regression, and solved it in $O(n^3)$ - they named their technique glasso. The glasso algorithm uses a series of L_1 penalized regressions, called Lasso regressions (Tibshirani, 1996); and we use the glasso algorithm for efficient optimization of our objective function.

Note that Equation 3.2 is a function of data X only through the sample covariance matrix $\mathbf{S}$, hence, we can replace the sample covariance matrix with a suitable similarity or kernel function. This is the key idea behind GINI's algorithm to deal with multiple images per gene, which we discuss in the next section.

3.2.2 Network inference from "multiple images per gene" ISH data

Multiple images of the same gene at the same time point should have the same gene expression pattern. However, in practice, the expression patterns in different images may differ considerably, for three main reasons.

Firstly, there is a wide interval of time considered as a single time point while collecting such data. For instance, the BDGP data divides embryonic development into 6 time stages. The last stage 13-16 corresponds to development of the embryo 9.3 to 15 hours after fertilization, which represents more than a third of the time taken for embryonic development. Hence, the true gene expression pattern may be dynamic within the time period of a single development stage, and the gene expressions captured for the same gene at the same time may not look similar to each other. Secondly, we might expect that for any organism for which ISH data is collected, there will necessarily be some ambiguity in how the development stage of the organism is labeled by human annotators. Finally, noise in the expression patterns due to excessive staining, lighting conditions and similar other reasons will also be observed. For all of the above reasons, any network-learning algorithm should leverage the existence of multiple images per gene per time point in improving its estimates of gene similarity.

The problem of multiple images per gene is reminiscent of multi-instance learning (Andrews et al., 2003; Maron and Ratan, 1998). Multi-instance learning is a form of supervised learning, where instead of labeling each instance, a bag of instances is labeled. A popular solution to the multi-instance problem is to define a multi-instance kernel, that can compute the similarity between bags of instances. Let $s(\mathcal{A})$ be a collection of order statistics of the set $\mathcal{A}$, for example, mean, median, minimum, maximum etc. In d dimensions, $s(\mathcal{A})$ is computed on each dimension independently, to form a vector of order statistics. If we use m order statistics, then the length of $s(\mathcal{A})$ will be dm. The similarity between gene g_i with a set of images $\mathcal{B}_i$ and gene g_j with images $\mathcal{B}_j$ can then be computed as

$$\mathbf{K}(\mathcal{B}_i, \mathcal{B}_j) = k(s(\mathcal{B}_i), s(\mathcal{B}_j)) \tag{3.5}$$

where $k(a, b)$ is an appropriate kernel function between vectors a and b. Such a kernel is called the statistic kernel.

The choice of the order statistics used in the kernel depends on the data collection procedure of the ISH. One concern in ISH data is that images may be overstained. In such a scenario, the median may be an appropriate choice of order statistic. If over-staining is not a concern, the maximum statistic may be more appropriate to ensure that information about presence of gene expression is not lost.

For the BDGP data, we use the covariance kernel $k(a, b) = Cov(a, b)$, and the mean statistic $s(\mathcal{B}) = \frac{1}{|\mathcal{B}|}\sum_{b \in \mathcal{B}} b$. The choice of using a single statistic to represent information from multiple images was due to the presence of noisy images in the data set. Thus,

$$\begin{aligned}\mathbf{K}(\mathcal{B}_i, \mathcal{B}_j) &= Cov\left(\frac{1}{|\mathcal{B}_i|}\sum_{a \in \mathcal{B}_i} a, \frac{1}{|\mathcal{B}_j|}\sum_{b \in \mathcal{B}_j} b\right) \\ &= \frac{1}{|\mathcal{B}_i|}\frac{1}{|\mathcal{B}_j|}\sum_{a \in \mathcal{B}_i}\sum_{b \in \mathcal{B}_j} Cov(a, b)\end{aligned} \tag{3.6}$$

Thus, our choice of kernel is equivalent to computing the mean similarity of all pairs of images in bags $\mathcal{B}_i$ and $\mathcal{B}_j$. This specific kernel is also known as the normalized set kernel, and has been shown to perform very well in multi-instance classification (Gartner et al., 2002).

Any kernel function may be written as the dot product in some higher dimensional feature space, i.e. $K(a,b) = \phi(a)^T\phi(b)$ (Aronszajn, 1950). Hence, if we assume that the data is drawn from a distribution such that $\phi(a)$ is a zero-mean Gaussian, we can learn the gene interaction network by treating $\mathbf{K}$ as the sample covariance matrix. Since estimating the inverse covariance matrix by solving equation 3.2 requires only the sample covariance matrix $\mathbf{S}$ and not the data itself, we can kernelize it by using the kernel matrix $\mathbf{K}$ defined in equation 3.6 as the required sample covariance matrix. Thus, the objective function is

$$\widehat{\boldsymbol{\Sigma}}^{\boldsymbol{-1}} = \arg\min_{\boldsymbol{\Theta}\succeq 0} \Big\{ trace(\mathbf{K}\boldsymbol{\Theta}) - \log\det\boldsymbol{\Theta} + \lambda||\boldsymbol{\Theta}||_1 \Big\}, \tag{3.7}$$

which can be solved as discussed in the previous section.

Consistency of the estimate

Given samples $X^{(1)}, X^{(2)}, \cdots, X^{(n)}$ drawn from a Gaussian distribution, it can be shown that the objective function in Equation 3.2 leads to a consistent solution, with a suitable choice of λ (Banerjee et al., 2006a). That is, the estimator $\widehat{\boldsymbol{\Sigma}}^{\boldsymbol{-1}}$ converges in probability to the true inverse covariance matrix $\boldsymbol{\Sigma}$.

GINI however does not work with samples from a Gaussian distribution, but directly with a multi-instance kernel $\mathbf{K}$. By definition, any kernel $\mathbf{K}$ can be represented as an inner product in some feature space ϕ, i.e. $\mathbf{K}(a,b) = \phi(a)^T\phi(b)$. For the multi-instance statistic kernel, $\phi = s(\mathcal{B})$, that is, the feature space is defined by the order statistics computed over bag $\mathcal{B}$. Since the order statistics for image data is bounded between 0 and 255, $s(\mathcal{B})$ is a bounded random variable. Hence the distribution of $s(\mathcal{B})$ is sub-Gaussian. For sub-Gaussian distributions, Ravikumar et al. (2011)

show that the penalized maximum likelihood estimator defined in Equation 3.7 is sparsistent, i.e. as the amount of data increases, the probability of identifying incorrect edges goes to zero. Thus, the kernelized estimator defined by GINI is sparsistent.

Thus, the GINI algorithm predicts the gene interaction network in two steps: in the first step, the similarity between different genes is computed using a multi-instance kernel. In the next step, a sparse interaction network is learned from the similarity matrix by solving Equation 3.7, and predicting edges corresponding to the non-zeros of the non-diagonal entries of the estimated $\widehat{\mathbf{\Sigma}}^{-\mathbf{1}}$. The next subsections describe the feature extraction, representation, and normalization process used to obtain suitable features from images that can be input into GINI.

3.2.3 Image processing

We convert the ISH images into canonical feature vectors suitable for analysis by our algorithm described above in a three-step manner. First, the precise expression pattern found in each image is extracted and aligned spatially to make all images spatially comparable. Next, each image is represented by a feature vector using Delaunay triangulation. Finally, features are normalized and feature selection is performed to extract meaningful features, that can be then used to compute the multi-set kernels to obtain gene similarity and learn the gene network.

Feature extraction via SPEX^2

ISH images are taken under diverse lighting conditions, and may suffer from poor quality staining/washing. A good feature extraction system must remove these effects, controlling for position, orientation etc. of the embryo and extract a precise gene expression pattern from the ISH images. In previous work (Puniyani et al., 2010), we developed SPEX^2, an automatic system for embryonic ISH image feature extraction. SPEX^2 registers each Drosophila ISH image by first extracting the embryo (foreground) from the image, using edge filters and image analysis techniques. Next, the alignment, size, shape and orientation of the embryo is determined, and normalized to a standardized ellipse. SPEX^2 also does automatic error detection and correction, rejecting images where the gene expression extraction process may have introduced errors, and also rejecting partial embryos, multiple embryos physically touching each other, and excessively dried or otherwise mishandled embryos. Next, the expression stain is extracted from the standardized embryo using a novel algorithm that maximizes the contrast between the stained and unstained regions of the embryo. Finally, an image segmentation algorithm using Markov random fields is defined to extract only the regions that have gene expression. Thus, a concise and high-fidelity gene expression pattern is extracted from the ISH image.

Feature representation via Delaunay triangulation

While SPEX^2 makes the images of different genes alignable spatially, and therefore directly comparable, the expression patterns must still be converted into an appropriate feature representation. One commonly used method for feature representation is to use the SIFT feature descriptor(**?**) in either a grid of points spaced uniformly through the image, with principal component analysis(PCA) used for dimensionality reduction(Puniyani et al., 2010), or via interest point detection and codebook generation (Yuan et al., 2012). Such techniques work well for supervised tasks like image annotation where a weight can be learned for each direction computed by PCA or for each codeword in the generated codebook. However, for unsupervised tasks, where weights cannot be learned, we wish to extract features that explicitly take into account the spatial distribution of the gene expression. A pixel level feature representation on the other hand, allows us to capture spatial information, but has high redundancy due to the correlation between neighboring pixels.

To reduce redundancy while capturing spatial gene expression information, we choose to overlay a fixed triangular mesh on top of the standardized embryo. The gene expression pattern for each image may then be represented as the median gene expression present in each triangle in this mesh. A mesh of 311 equilateral triangles was produced by using the Delaunay triangulation algorithm (Persson and Strang, 2004), and aligning the mesh to the standardized embryo, as described in Frise et al. (2010). Each image can then be represented as a feature vector of length 311, with each feature representing the median gene expression expressed in a specific location on the embryo, which is fixed across all images. For example, triangle 1 may correspond to the head in all images, and so on. Modeling the spatial locations in a lower dimensional space via triangulation helps in approximating the independence assumption made in the GINI algorithm, analogous to using coarse time definitions while making microarray measurements.

Figure 3.2 shows examples of ISH images converted into the triangulated gene expression patterns. As can be seen, triangulating the SPEX^2 output captures the key features of the gene expression location and strengths. Thus, triangulation enables reducing the dimensionality of the feature space, while retaining explicit spatial information about the gene expression, which other dimensionality reduction techniques would not be able to capture.

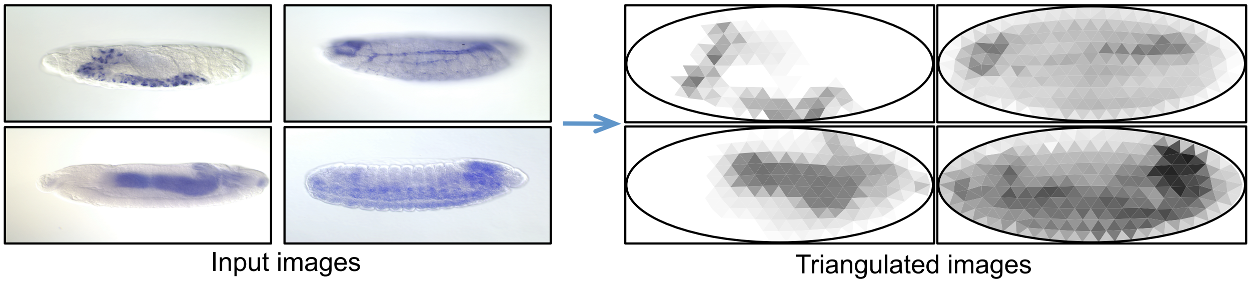


Figure 3.2. Triangulation. Examples of how ISH images are converted into low-dimensional triangulated representations, for efficient feature representation.

Feature processing

The feature vectors extracted by triangulating the expression patterns are not normalized, hence, we need to adjust the signal obtained from different images to a common scale. The set of triangulated features may also contain uninformative features that may add a bias if used directly to compute the multi-set kernel. Further, the gene network analysis should only consider genes with informative expression patterns that have non-trivial spatial expression in the data. Hence, we need to further process the features to select informative features and genes, and normalize the features in an appropriate manner.

Feature normalization:

Unlike in microarray data, the currently available ISH data does not measure the signal related to nonspecific binding of the probe for each image, hence the background correction of intensities cannot be image specific. Each gene expression pattern is normalized to have its expression values(t) lie between 0 and 255 (the minimum and maximum color value). The feature value is then computed as the logarithm of the expression value : $\log(1 + t)$.

Feature & gene selection:

A large percentage of the ISH images have no stain, or ubiquitous staining. In the BDGP data, 55% of the genes have at least one image, in at least one time point, with no stain. Since no information may be inferred from such data, these images must be removed from analysis. This can be achieved by removing expression patterns having variance below a threshold (ϵ, usually 0.1).

Additionally, features that have low variance in the data set are capturing no information about the gene expression variation across multiple genes. Hence, they must be removed from the analysis as well. Since removing images from the analysis affects the feature variance and vice-versa, we alternate removing features and images with low variance, until both feature variance and image variance is greater than the threshold. The algorithm is described formally in Figure 3.3.

```
Input: triangulated features for $n$ images : $t_i$, where $i \in \{1, ...n\}$;
       variance threshold $\epsilon$
Output: normalized and processed features in matrix $\mathbf{X}$
for image $i = 1 \cdots n$ do
    $\mathbf{X}(i, \bullet) = \log(1 + 255 * \frac{t_i - \min(t_i)}{\max(t_i) - \min(t_i)})$ ;
end
while $(\min(var(\mathbf{X})) < \epsilon \mid\mid \min(var(\mathbf{X}')) < \epsilon)$ do
    $keepImages = find(var(\mathbf{X}') > \epsilon)$;
    $\mathbf{X} = \mathbf{X}(keepImages, \bullet)$;
    $keepFeatures = find(var(\mathbf{X}) > \epsilon)$;
    $\mathbf{X} = \mathbf{X}(\bullet, keepFeatures)$;
end
```

Figure 3.3. Algorithm outlining feature normalization and processing. $var(\mathbf{A})$ for matrix $\mathbf{A}$ returns a vector containing the variance of each column of $\mathbf{A}$; $find(y)$ returns the indices of the non-zero elements of vector y, and $\mathbf{A}'$ is the transpose of matrix $\mathbf{A}$.

3.2.4 Summary of the GINI system

Putting everything together, we conclude the method section with a summary of the GINI system for network inference from ISH images. Each ISH image is converted into a standardized expression pattern using SPEX^2, and then triangulated to extract a low-dimensional spatial feature vector. Next, feature values are normalized, uninformative features are removed, and genes with insufficient information available are rejected. Finally, the multi-set kernel is used to compute the similarity between the bags of image vectors available for each gene, and the gene network is estimated using Equation 3.7. The algorithm is summarized in Figure 3.4.

```
Input: Embryonic ISH images for $n$ genes ;
$\lambda$ - tuning parameter to control sparsity
Output: Predicted gene network for the $n$ genes
for gene $i = 1 \cdots n$ do
    // feature extraction
    $\mathcal{B}_i = \{\}$;
    for each image $j$ of gene $g_i$ do
        Extract expression patterns from image $j$ using $\mathrm{SPEX}^2$;
        $t_j$ = triangulate expression pattern of image $j$;
        // feature normalization
        $t_j = \log(1 + 255 * \frac{t_j - \min(t_j)}{\max(t_j) - \min(t_j)})$ ;
        $b_j = \mathrm{feature\_selection}(t_j)$;
        $\mathcal{B}_i = \mathcal{B}_i \cup \{b_j\}$;
    end
    // $\mathcal{B}_i$ is now the set of all normalized features of the
        images of gene $g_i$
end
// Define the multi-instance kernel
for gene $i = 1 \cdots n$ do
    for gene $j = 1 \cdots n$ do
        $\mathbf{K}(i, j) = Cov\left(\frac{1}{|\mathcal{B}_i|} \sum_{a \in \mathcal{B}_i} a, \frac{1}{|\mathcal{B}_j|} \sum_{b \in \mathcal{B}_j} b\right)$
    end
end
// Run glasso using kernel $\mathbf{K}$
$\mathbf{\Sigma^{-1}} = glasso(\boldsymbol{K}, \lambda)$ ;
Predicted edges in the network: $\mathcal{E}$ = non-zeros in the non-diagonal elements of $\mathbf{\Sigma^{-1}}$ ;
```

Figure 3.4. The final GINI algorithm to obtain the gene network from ISH images.

Computational complexity

We assume that the number of images per gene is small and bounded by a constant, and hence the total number of images is $O(n)$, where n is the number of genes. Then, given the triangulated features of all images, feature and gene selection takes time $O(nd^2)$ and $O(n^2d)$ to compute the correlation matrix in feature and gene space respectively. Computing the kernel requires $O(n^2d)$ time, and finally, the computational complexity of minimizing the log-det divergence is known to be $O(n^3)$. The overall computational complexity is then $O(n^3 + nd^2 + n^2d)$. Assuming

$d << n$, the complexity may be assumed to be $O(n^3)$. The implementation is efficient, and computes a gene network for ~ 2000 genes in a few minutes on an Intel Core-2 CPU with 2 GB memory.

3.3 Results

We first demonstrate that the independence and Gaussian assumptions are reasonable for ISH data, and that GINI explains the ISH data well, with small fitting errors, and no bias in the residues. Next, we show the performance on a small subset of 12 images for 6 genes to verify that the network predicted by GINI is reasonable. We then run GINI on two datasets of ISH images from 2 time points in the BDGP data, and study the networks. We find the networks are modular and scale free as expected. Furthermore, different regions of the networks are enriched for spatial annotations, and the hubs of the networks are functionally enriched for important cellular functions. Finally, we demonstrate that these results could not have been obtained by analyzing microarray data.

3.3.1 Validation of the GINI assumptions : independent spatial data

GINI assumes that the gene expression in each triangle can be assumed to be independently drawn from a multi-variate Gaussian. However, the true gene expression in adjacent spatial locations is correlated and not independent. To verify that this dependence of adjacent samples does not affect the accuracy of the estimated network, we simulate synthetic data where the underlying network is known, but the data points are not independent of each other, and test whether GINI can recover the correct network in such a scenario. The data samples depend on each other via a parameter c that captures degree of dependence between data samples. When

$c = 0$, all data samples are drawn i.i.d. from the known distribution. As c increases, data samples are drawn from the same distribution, but they depend on the adjacent samples.

Data generation

For $p = 50$ dimensions, the true inverse covariance matrix was constructed by using the AR(1) model from Yuan and Lin (2007). That is, $\Sigma^{-1}_{i,i} = 1$, and $\Sigma^{-1}_{i,i+1} = 0.5$, with all other elements being zero. Dependent samples are generated from a zero mean Gaussian having the above known inverse covariance matrix, as explained below.

Let $c \in [0, 1)$ be the fractional overlap between adjacent samples. The first sample is sampled independently from the above specified Gaussian. The $(i + 1)^{th}$ sample is generated from the i^{th} sample as follows. Pick $c * p$ random features f, and copy the value of the previous sample for these features : $X(i + 1, f) = X(i, f) = a$. Now, X_{i+1} can be partitioned into the "known" f features and the remaining q features which still need to be sampled, conditioned on $X(i+1, f) = a$. If we partition Σ as

$$\begin{pmatrix} \Sigma_{ff} & \Sigma_{fq} \\ \Sigma_{qf} & \Sigma_{qq} \end{pmatrix}$$

then X_q conditioned on $X_f = a$ can be shown to be Gaussian with mean $\bar{\mu}$ and covariance $\bar{\Sigma}$, which can be computed as below, and $X(i + 1, q)$ can be sampled from it.

$$\bar{\mu} = \Sigma_{qf}\Sigma^{-1}_{ff}a \tag{3.8}$$

$$\bar{\Sigma} = \Sigma_{qq} - \Sigma_{qf}\Sigma_{ff}^{-1}\Sigma_{fq} \tag{3.9}$$

We ran two experiments. In the first, a fixed number of samples($n = 100$) were used to learn the network. In the second, as c increases, more samples ($n = 100\sqrt{p * c + 1}$) are available for learning the network. In both experiments, for each c value, we randomly sample data points X using the method outlined above, estimate the $\widehat{\Sigma}^{-1}$ matrix, and compare it to the known Σ^{-1} matrix, to compute precision and recall. Results are averaged over 50 runs of the experiment.

Figure 3.5(a) shows that as c increases, the precision (fraction of correct interactions among all inferred ones) and recall (fraction of correct interaction among all true interactions) stay constant for small values for c. Only when the amount of dependence increases beyond half, do we see a small reduction in accuracy. Thus, we can conclude that even if there is a large spatial dependence in gene expression, the result is equivalent to a slight reduction in performance. Futher, in Figure 3.5(b), we see that if we can increase the number of data points as we increase c, the performance remains the same as using i.i.d. data.

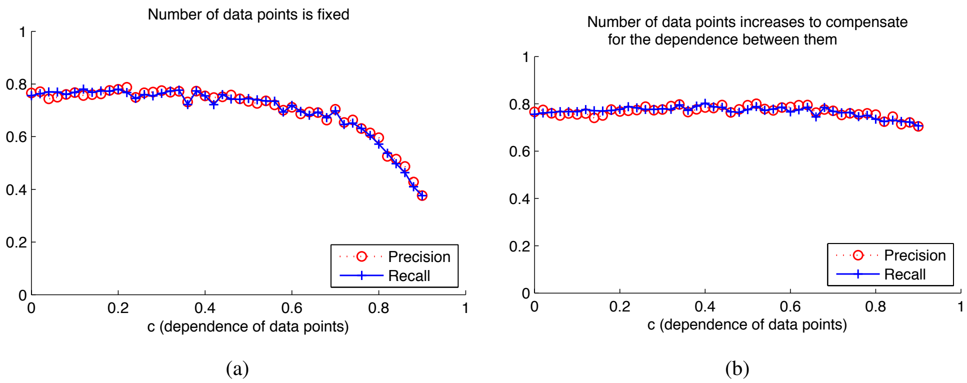


Figure 3.5. GINI assumptions are reasonable. Even if the data are not independent draws from the Gaussian, the network can still be learned with high precision and recall. (a) For a fixed number of data points, as c increases beyond 0.5, the precision and recall reduces. (b) If we allow the number of data points n to increase as c increases, the precision and accuracy of the method is not affected. The standard deviation at each point in both results is approximately 0.09.

3.3.2 GINI explains the ISH data well

For a high-dimensional distribution, it is not feasible to test if the data is truly Gaussian. However, a consequence of Gaussianity is that for each gene, the gene expression can be expressed as a weighted linear sum of the expression values of a few other genes, which form the edges of the network. To test if this assumption holds true in ISH data, for each gene, we fit a linear regression between the gene and its neighbors found by GINI and look at the absolute value of the error i.e. the mean absolute difference between the predicted and the known gene expression. When the maximum expression value is 1, for more than 90% of the genes we looked at, the absolute error was less than 0.02; 99.5% of all genes had absolute error less than 0.05, confirming that the GINI generative model explains the ISH data.

We also confirm that the prediction error is not systematic with respect to the spatial location. For each gene, we compute the prediction error (residue) when the gene is predicted by regressing it on its neighbors. For each spatial location, we plot the mean residue at that location for all genes. As can be seen in Figure 3.6, there is no systematic bias in the spatial positions that are hardest to predict for any gene.

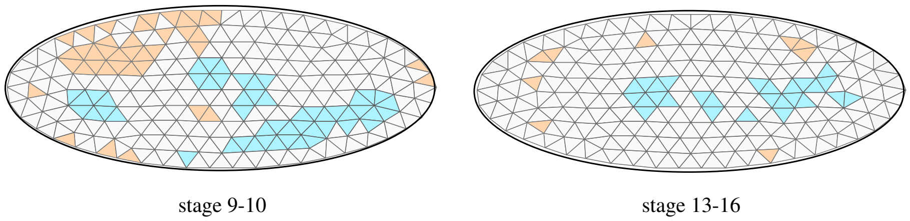


Figure 3.6. GINI error analysis. Locations where gene expression cannot be predicted easily. Red color indicates that the true gene expression was higher than predicted by the regression, while blue indicates that the true gene expression was lower than predicted by the regression. Note that since the difference in the true and predicted gene expressions is very small, the mean residue values were multiplied by 10 to improve the contrast of the image for visualization purposes. Thus, there is no systematic bias in the spatial locations where expression is hard to predict.

3.3.3 Network on limited data

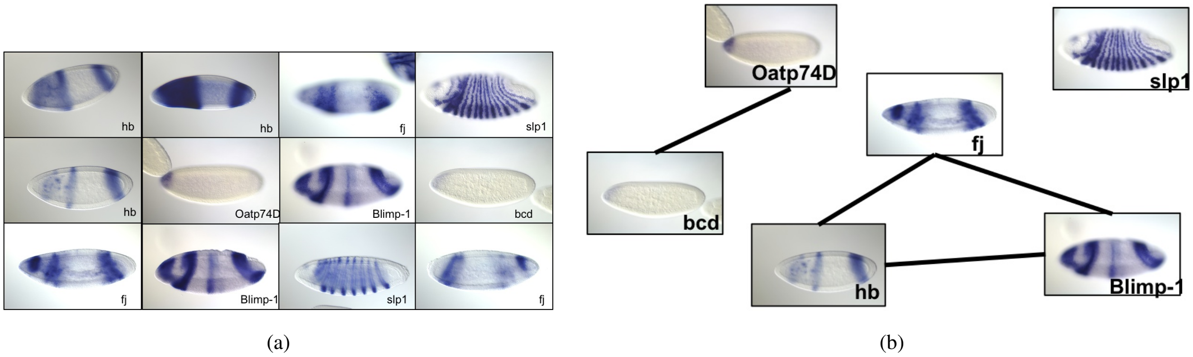


Figure 3.7. Example network on small data. (a) 12 images input to the GINI system, and (b) network of genes learnt by it, with each gene represented by one image.

Before running our algorithm on a large sized dataset, we construct an artificial small data set to verify the results. We input 12 images, shown in Figure 3.7(a) from 6 genes to the GINI algorithm (each gene has 1-3 images in the data set). With $\lambda = 0.46$, 4 edges are predicted in the network, shown in Figure 3.7(b). As can be seen, the three genes hunchback(*hb*), four-jointed(*fj*), and *Blimp-1*, which are expressed in the dorsal, ventral and procephalic ectoderm, are connected in a single cluster. Similarly, the genes organic anion transporting polypeptide 74D(*Oatp74D*) and bicoid(*bcd*) are connected by an edge, since both show expression in the foregut and the anterior endoderm. Finally, the expression of sloppy paired-1*(slp1)* was considered to be sufficiently different from the other genes, hence it is not connected to any other gene in the network.

Thus, the gene interaction network found by GINI can be verified to be reasonable for the above small data set.

3.3.4 Network on the whole BDGP data

We now turn our attention to the ISH images from the Berkeley Drosophila Genome Project data set. We have obtained around 67400 ISH images of 3509 protein-coding genes from the

BDGP data released in September 2009, captured at key development stages of embryonic development. Each image captures embryonic gene expression of a single gene using RNA in-situ hybridization. Each image was labeled manually with the age of the embryo, categorized into six distinct embryonic stages : 1-3, 4-6, 7-8, 9-10, 11-12, and 13-16. Genes are also annotated with ontology terms from a controlled vocabulary of around 295 terms, describing the unique embryonic structures in which gene expression is observed during the various stages of embryonic development. SPEX^2 analyzes these image automatically, rejecting unsuitable images, to produce 51593 expression patterns of 3347 genes.

As proof of concept, we focus on images viewed from a lateral perspective from two development stage ranges of this data : 9-10 and 13-16. For the stage 9-10, we have 2869 expression patterns of 2609 genes, and for stage 13-16, we have 6350 expression patterns of 3258 genes. We extracted features as described in the methods section. For each development stage, we ran a separate analysis.

Using a λ value of 0.775 for stage 9-10, we ran GINI and obtained a network having 258 genes, and 516 interactions (edges) between them. For the development stage 13-16, we used $\lambda = 0.875$, and obtained a network with 1202 genes and 3666 interactions between them. The λ value was selected for each network by running GINI for 21 λ values between 0.5 and 1, and picking a value such that the mean-degree for the network is reasonable (approximately 2-3) - see Supplementary Figure S1 for a plot that shows how the number of edges in the network decreases as λ increases.

Some of the interactions predicted by GINI have already been reported in the literature. For example, in the network for stage 9-10, GINI predicts that DCP-1 (CG5370), an effector caspase which is involved in apoptosis, will interact with the *thread* gene (CG12284), a known inhibitor of apoptosis protein (JS et al., 2003). GINI also predicts that *Snf5- related 1*(CG1064) interacts with *echinoid* (CG12676), both of which are known to be involved in epidermis development, muscle organ development, as well as imaginal disc-derived wing vein morphogenesis. In the 13-16 development network, GINI predicts that the *capping protein beta* gene (CG17158) interacts with the *Glycogen phosphorylase* gene (CG7254), and *Tpc1* (CG6608) interacts with CG2812, which has been previously reported in Giot et al. (2003).

The next five subsections do a detailed analysis of the 2 networks.

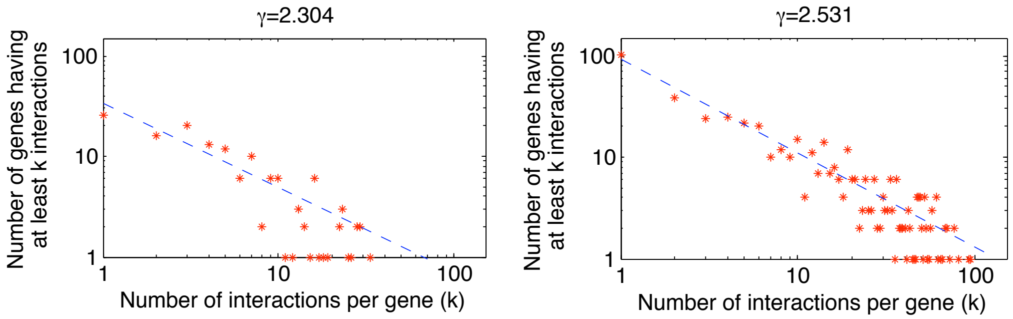
3.3.5 Scale free network

A network is said to be scale free if its degree distribution asymptotically follows a power law. That is, the fraction of genes $P(k)$ that have at least k interactions with other genes is

$$P(k) = ck^{-\gamma} \tag{3.10}$$

where γ is the scale free parameter, and c is the normalization constant. It has been hypothesized that gene regulatory networks are scale free (Basso et al., 2005). We looked at the characteristic of our interaction networks by plotting the number of interactions per gene (Figure 3.8), and found that the networks found by GINI are scale free. The γ parameter obtained is 2.3 and 2.5 for the 9-10 and 13-16 networks respectively, which corresponds well to the values observed for a large variety of power law graphs. The scale free nature of the network was found to be independent of the λ tuning parameter of the algorithm.

Unlike the gene regulatory network obtained for Human-B cells (Basso et al., 2005), we found that the scale-free nature of the gene network we obtain has a good fit, without observing a deviation from the expected at low connectivity values. However, this could be a side-effect of the larger number of genes they analyzed.



stage 9-10

stage 13-16

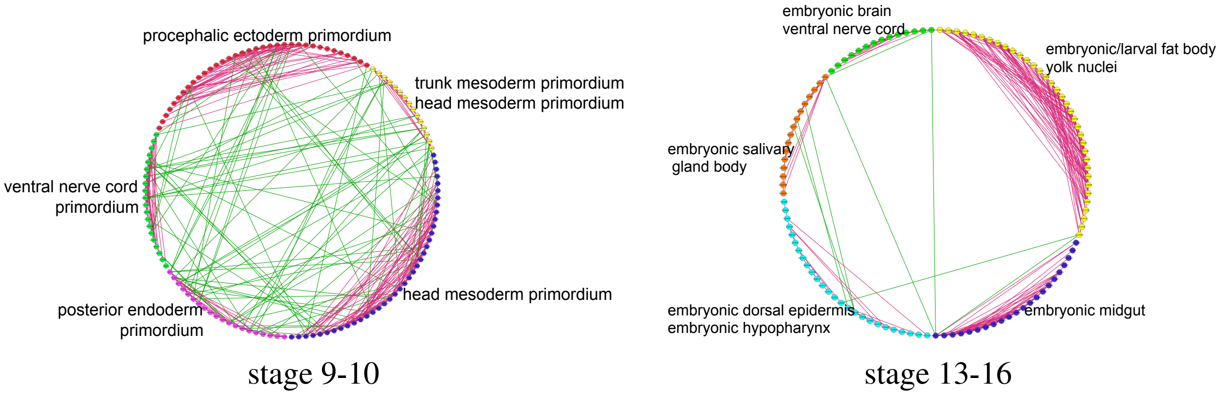
Figure 3.8. Scale-free network. Connectivity properties of the reconstructed network for time stage 9-10, and 13-16. The scale free nature of the plot can be observed for both networks. The plot for stage 9-10 has fewer points since the network constructed has fewer nodes and edges.

3.3.6 The BDGP networks are modular

Using spectral clustering, we construct 12 regions or clusters within each network, and visualize the five biggest clusters of each of the networks in Figure 3.9. All 12 clusters in both networks are very well separated. The ratio of within-cluster edges to total number of edges is 70% and 87% for the 9-10 and 13-16 development stage networks respectively, indicating that the estimated networks are highly modular. From a biological perspective, different parts of gene networks may be responsible for different pathways or biological functional components of the cell, thus modularity is a good prediction for real interaction networks.

3.3.7 Hub analysis

Given the scale-free nature of the network, a small number of the genes have a large number of interactions. We analyze the Gene Ontology functions of the genes having the largest number of interactions, i.e. the hubs of the network. The question we wish to address is : if we pick the top 5% of the genes having the maximum connectivity with other genes, what kind of functional enrichment do these genes have? Our background population is of the 2609 and 3258 genes for which we have at least one ISH image describing its expression for the 9-10 and 13-16 stages


procephalic ectoderm primordium
trunk mesoderm primordium
head mesoderm primordium
ventral nerve cord primordium
posterior endoderm primordium
head mesoderm primordium
embryonic brain
ventral nerve cord
embryonic/larval fat body
yolk nuclei
embryonic salivary gland body
embryonic dorsal epidermis
embryonic hypopharynx
embryonic midgut

stage 9-10

stage 13-16

Figure 3.9. Modular network. A global view of the networks constructed by our algorithm for development stage 9-10, and 13-16, visualized for 5 of the 12 clusters in the network. The nodes of each cluster in the network are represented by different colors. Red edges are edges between nodes in the same cluster, while green edges are edges between nodes in different clusters. Each cluster is represented by one or two spatial annotation terms enriched in the cluster.

Table 3.1. GO functional analysis for the gene hubs of the GINI network

Stage	Gene Ontology term	Hub frequency	Genome frequency	P-value
9-10	cellular macromolecule metabolic process	57 of 119 genes, 47.9%	652 of 2575 genes, 25.3%	1.79e-05
	macromolecule metabolic process	62 of 119 genes, 52.1%	772 of 2575 genes, 30.0%	8.16e-05
	cell cycle	24 of 119 genes, 20.2%	174 of 2575 genes, 6.8%	0.00029
	primary metabolic process	67 of 119 genes, 56.3%	962 of 2575 genes, 37.4%	0.00559
	cell cycle phase	18 of 119 genes, 15.1%	127 of 2575 genes, 4.9%	0.00667
	cellular metabolic process	64 of 119 genes, 53.8%	910 of 2575 genes, 35.3%	0.00827
	mitotic cell cycle	18 of 119 genes, 15.1%	131 of 2575 genes, 5.1%	0.01039
	cell cycle process	19 of 119 genes, 16.0%	146 of 2575 genes, 5.7%	0.01313
	cellular process	90 of 119 genes, 75.6%	1508 of 2575 genes, 58.6%	0.01798
	macromolecule modification	20 of 119 genes, 16.8%	163 of 2575 genes, 6.3%	0.01888
	protein modification process	19 of 119 genes, 16.0%	155 of 2575 genes, 6.0%	0.03111
	nucleobase-containing compound metabolic process	39 of 119 genes, 32.8%	476 of 2575 genes, 18.5%	0.04665
13-16	energy derivation by oxidation of organic compounds	13 of 159 genes, 8.2%	47 of 3217 genes, 1.5%	0.00011
	cellular respiration	12 of 159 genes, 7.5%	43 of 3217 genes, 1.3%	0.00031
	generation of precursor metabolites and energy	14 of 159 genes, 8.8%	60 of 3217 genes, 1.9%	0.00039
	electron transport chain	10 of 159 genes, 6.3%	32 of 3217 genes, 1.0%	0.00093
	mitochondrial ATP synthesis coupled electron transport	9 of 159 genes, 5.7%	26 of 3217 genes, 0.8%	0.00121
	ATP synthesis coupled electron transport	9 of 159 genes, 5.7%	27 of 3217 genes, 0.8%	0.00174
	cellular process	116 of 159 genes, 73.0%	1808 of 3217 genes, 56.2%	0.00211
	respiratory electron transport chain	9 of 159 genes, 5.7%	28 of 3217 genes, 0.9%	0.00246
	oxidative phosphorylation	9 of 159 genes, 5.7%	29 of 3217 genes, 0.9%	0.00342
	ribosome biogenesis	7 of 159 genes, 4.4%	20 of 3217 genes, 0.6%	0.01640
	cellular metabolic process	78 of 159 genes, 49.1%	1091 of 3217 genes, 33.9%	0.01708
	mitochondrial electron transport, NADH to ubiquinone	6 of 159 genes, 3.8%	15 of 3217 genes, 0.5%	0.02661

GO functional analysis for the gene hubs of the networks learned for the two development stages by GINI. Both networks have hubs that are enriched for multiple important cellular functions.

respectively. We use the hypergeometric test, with Bonferroni correction used to correct for multiple hypothesis tests (Boyle et al., 2004). As can be seen in Table 3.1, we observe enrichment of

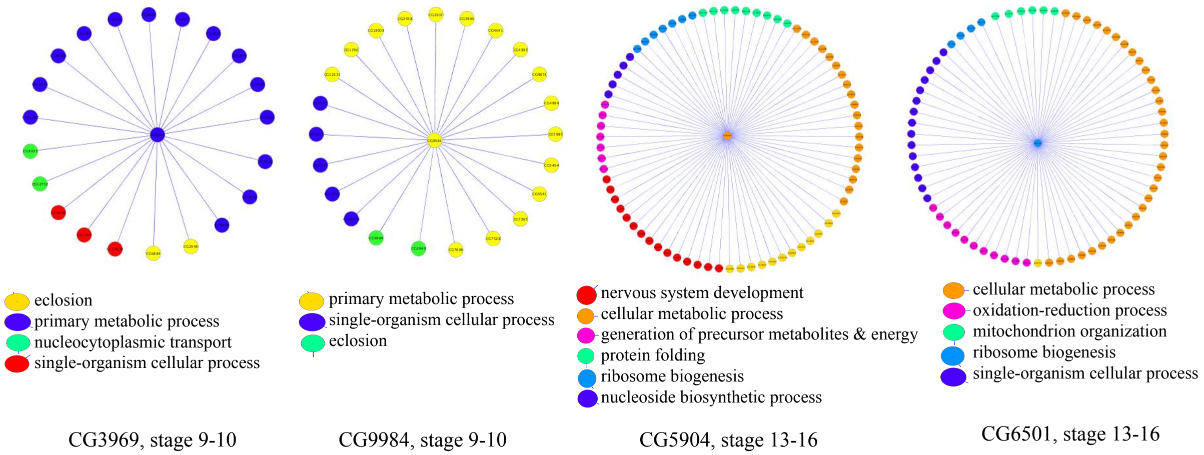

eclosion
primary metabolic process
nucleocytoplasmic transport
single-organism cellular process
CG3969, stage 9-10
primary metabolic process
single-organism cellular process
eclosion
CG9984, stage 9-10
nervous system development
cellular metabolic process
generation of precursor metabolites & energy
protein folding
ribosome biogenesis
nucleoside biosynthetic process
CG5904, stage 13-16
cellular metabolic process
oxidation-reduction process
mitochondrion organization
ribosome biogenesis
single-organism cellular process
CG6501, stage 13-16

Figure 3.10. Hubs of the GINI network. A look into the neighborhoods of a few hubs from the GINI networks for stage 9-10 and 13-16. A few enriched GO groups are highlighted in the subnetworks as shown.

a wide variety of functions that are essential to cell growth and functioning, including metabolic processes, cellular respiration, transport of electrons and ions, protein modification, ribosome biogenesis etc.

Next, we examine a few high-degree hubs in the two networks in detail, along with their neighborhood genes in the networks. Figure 3.10 shows the hub neighborhood for two genes in the 9-10 development stage network. CG3969 is a Activated Cdc42 kinase-like gene known to be involved in protein phosphorylation (Consortium, 2002) and cell death (Gorski et al., 2003), and CG9984 (*TH1*) is known to be involved in regulation of biosynthetic process (Missra and Gilmour, 2010) and nervous system development (Neumüller et al., 2011). Both genes interact with many genes having functions related to the primary metabolic process, and single-organism cellular process. In stage 13-16, we examine the hub neighborhood of CG5904 and CG6501. The mitochondrial ribosomal protein CG5904 has been previously predicted to be a structural constituent of ribosome(Koc et al., 2001), and we find that it interacts with many genes involved in the ribosome biogenesis. Gene CG6501 (*Ns2*) has been previously predicted to be involved in phagocytosis, engulfment(Stroschein-Stevenson et al., 2006), and ribosome biogenesis (Consortium, 2002);

CG6501's neighborhood has multiple genes that are also involved in ribosome biogenesis and single-organism cellular process.

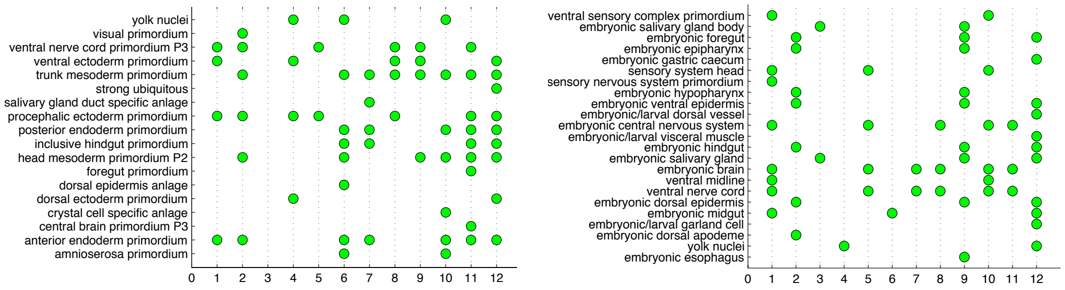
3.3.8 Enrichment of annotation terms

Each gene in the BDGP data has been labeled manually by annotations describing the spatial gene expression, using 295 annotation terms. We expect that since the gene interaction network is constructed via spatial similarity, genes that are connected to each other in the network will have similar spatial annotation terms.

To test this, we cluster the gene network using spectral clustering (NG et al., 2001) into 12 clusters, and analyze the enrichment of each cluster for annotation terms using the hypergeometric test, with Bonferroni correction used to correct for multiple hypothesis tests. In the gene network for the 9-10 stage, 11 of the 12 clusters are enriched for 63 total annotation terms (Figure 3.11). The only cluster not showing any enrichment in the 9-10 stage network is also the smallest cluster, having only 4 genes. For example, in cluster 8, 92% of the genes have expression in the ventral nerve cord primordium P3 , while only 8% of the genes in the data have expression in this region. Similarly, 73% of the genes in cluster 11 have expression in the trunk mesoderm primordium, while only 16% of the genes in the data have expression in this region. For the 13-16 stage network, all 12 clusters are enriched for a total of 81 enrichments, a part of which is visualized in Figure 3.11. Tables S1 and S2 in the supplementary material report the complete enrichment analysis.

Triangulation improves quality of result

Previous work on image processing for ISH images has focused on using SIFT features, and constructing a codebook that contain all the embryonic structures that the system is expected to



stage 9-10

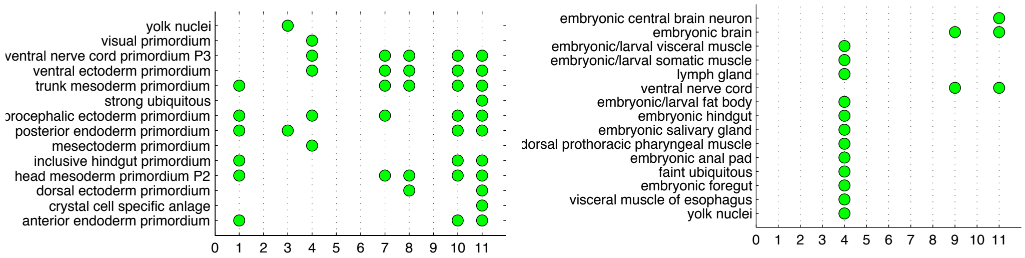
stage 13-16

Figure 3.11. Spatial annotations. Enrichment analysis for clusters in the gene interaction networks found by GINI. A green dot indicates enrichment with a P-value < 0.05.

annotate(Yuan et al., 2012). In this section, we show that triangulation produces more interesting networks over such a SIFT feature representation. We use the SPEX^2 gene expression patterns, and represent them by constructing SIFT features of the expression pattern over a grid. These grid SIFT features are then represented with a codebook of 2000 dictionary features, as described in(Yuan et al., 2012). We then use these dictionary features instead of the triangulated features to learn the GINI network. Figure 3.12 shows that the resulting networks are not as richly enriched as the ones derived from the triangulation features in Figure 3.11. The total number of enrichments in the SIFT codebook network is 42 and 21 for the 9-10 and 13-16 development stage networks respectively. In contrast, the triangulated GINI networks had 63 and 81 enrichments for the 9-10 and 13-16 stage networks. Figure 3.13 shows that this result is independent of the number of clusters selected for the analysis, for both triangulated networks as well as SIFT codebook networks.

Sensitivity to the tuning parameter λ

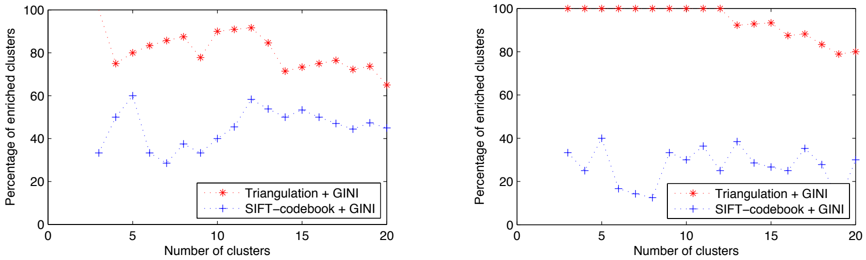
Supplementary Figure S1 shows how the number of edges in the network decreases as the tuning parameter(λ) of the GINI algorithm increases. To confirm that the enrichment results are not sensitive to the choice of λ, we obtained 21 predicted networks by varying the λ value uniformly from 0.5 to 1. For each network, we repeated the clustering and enrichment analysis, and found



stage 9-10

stage 13-16

Figure 3.12. Enrichment analysis on networks learned from SIFT dictionary features instead of triangulation features. The network for development stage 9-10 has only 7 enriched clusters of the 12 clusters in the network. For the stage 13-16 network, only 3 of the 12 clusters are enriched for spatial annotations.

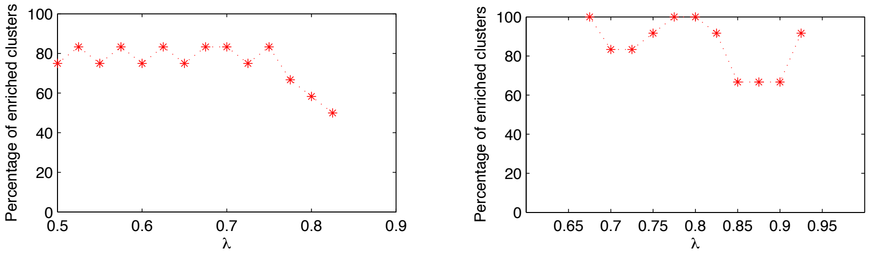


stage 9-10

stage 13-16

Figure 3.13. SIFT codebook features do not perform as well as triangulated features on ISH data. Percentage of clusters enriched for spatial annotations in networks predicted by GINI as a function of number of clusters for data from development stage 9-10 and 13-16. As can be seen, using triangulated features produces networks with more enriched clusters than using SIFT-codeword features, independent of the number of clusters selected for the analysis. Further, the enrichment of the GINI network clusters does not significantly vary as the number of clusters are varied.

that the enrichment for term annotations is not highly sensitive to choice of λ (Figure 3.14). The enrichment results are also not dependent on the number of clusters - we get high enrichment, independent of the number of clusters chosen while running the clustering algorithm (Figure 3.13).



stage 9-10

stage 13-16

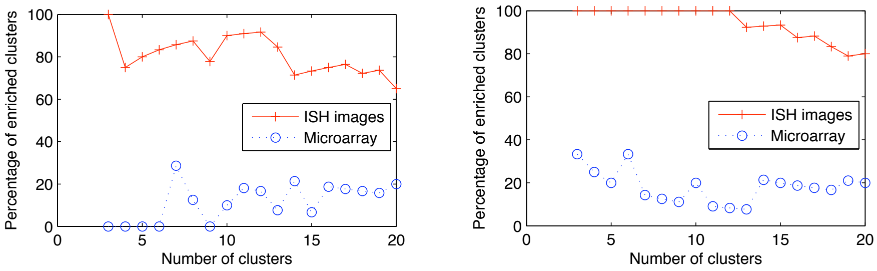
Figure 3.14. λ **tuning.** Percentage of clusters enriched for spatial annotations in networks predicted by GINI as a function of tuning parameter λ for data from development stage 9-10 and 13-16. As we increase λ, the number of edges predicted in the network decrease, however, the enrichment of the different clusters stays almost constant. Thus, qualitative analysis of the network does not seem to be sensitive to choice of λ.

3.3.9 Comparison with microarray network

We learn a network from microarray data collected by the BDGP project over 12 time points in embryonic development (Tomancak et al., 2002b), over the same genes that are being studied in the 9-10 and 13-16 networks, using covariance between the microarray expression as the kernel. We find that the overlap in edges between the 2 networks is very small, only 1% of the edges are common to both networks. If we assume that spatial expression annotations are a proxy for functional enrichment, then we can check if the microarray network is enriched for the spatial annotation terms. Figure 3.15 shows that the percentage of enriched clusters in the microarray network is small, independent of the number of clusters analyzed.

We can also test functional GO enrichment of the hubs of the network. Table 3.2 shows that the hubs of the microarray network for stage 13-16 are enriched for only a single function, where 4 of the 145 hub genes are involved in the "aromatic compound catabolic process", while the microarray data network for stage 9-10 has no enrichments.

Thus, we find that the network learned from ISH images is clearly different from a network learned from microarray data. The ISH image network is enriched for spatial annotation terms, as well as functional enrichment of the hubs of the network, which does not hold true for the



stage 9-10

stage 13-16

Figure 3.15. Microarray v/s ISH data. The percentage of clusters that are enriched for spatial term annotations using networks learned from ISH and microarray data.

microarray network. This suggests that analyzing ISH images could support different scientific conclusions, which should be studied in greater detail.

Table 3.2. GO functional analysis for the gene hubs of the microarray network

Stage	Gene Ontology term	Hub frequency	Genome frequency	P-value
13-16	aromatic compound catabolic process	4 of 145 genes, 2.8%	6 of 3213 genes, 0.2%	0.01841

GO functional analysis for the gene hubs of the microarray networks learned for genes with images in the 13-16 development stage. No enriched terms were found for the microarray network constructed on genes from the 9-10 stage.

3.4 Discussion

GINI predicts gene interaction networks by analyzing Drosophila embryo ISH images. While the experiments above have been reported on the ISH data from BDGP, the GINI algorithm can be applied to all image data, by suitably modifying only the image processing SPEX^2 pipeline. Using synthetic and image data, we establish that GINI fits the ISH data well, with low error residues, and that it can learn the true network correctly even if the data is not completely i.i.d. The analysis of the BDGP data shows that the hubs of the predicted gene interaction network are enriched for essential cellular functions, and that different regions of the interaction network are enriched for different combinations of annotation terms describing the gene expression. Thus, the predicted gene interaction network is capturing essential spatial and functional information about the expression pattern of the genes. We found that the gene interaction network learned from ISH images differs significantly from a network learned from microarray data.

The current work focuses on extracting gene networks from spatial data. The next step is combining information from multiple time stages to improve predictions, thus learning spatial-temporal gene networks. The problem of time-varying networks has been studied extensively for microarray data, by using different statistical penalties to estimate the network. For example, Ahmed and Xing (2009a) construct time varying networks by using a temporally smoothed L_1-regularized logistic regression formulation, while Danaher et al. (2013) propose a fused lasso and group lasso based approach to combine information across time. Extensions of such algorithms for image data require stronger assumptions on data quality, such as having the same number of genes and image quality across time. Further, certain development stages may be less informative than others; for example, very few genes are active at development stage 1-3, and expression data from this stage is not as informative as expression data from development stage 13-16, when the embryo is much more mature. Developing algorithms that can account for such variations in data quality, while combining information across time, remains an interesting future direction to explore.

Chapter 4

NP-MuScL: Unsupervised global prediction of interaction networks from multiple data sources

Prediction of gene interaction networks from expression data usually focuses on global network estimation from a single data source. However, in many real world applications, multiple data sources are available that give information about the same set of genes. We propose NP-MuScL (nonparanormal multi source learning) to estimate a gene interaction network that is consistent with multiple sources of data, having the same underlying relationships between the nodes **?**. NP-MuScL casts the network estimation problem as estimating the structure of a sparse undirected graphical model. We use the semiparametric Gaussian copula to model the distribution of the different data sources, with the different copulas sharing the same covariance matrix, and show how to estimate such a model in the high dimensional scenario. Results are reported on synthetic data, where NP-MuScL outperforms baseline algorithms significantly, even in the presence of noisy data sources. Experiments are also run on two real-world scenarios: two yeast microarray data sets, and three Drosophila embryonic gene expression data sets, where NP-MuScL predicts a higher number of known gene interactions than existing techniques.

4.1 Introduction

There have been two popular approaches to reverse engineering gene networks. The first approach is to build a generative model of the data, and learn a graphical model that captures the conditional independencies in the data. Learning the structure of a graphical model under a multivariate Gaussian assumption of the data has received wide attention in recent years (Banerjee et al., 2006b; Friedman et al., 2008; Meinshausen and Bühlmann, 2006); various algorithms have been proposed (Banerjee et al., 2006b; Friedman et al., 2008; Meinshausen and Bühlmann, 2006), many with theoretical analysis offering asymptotic guarantee of consistent estimation of the interactions between genes in the network. Empirically, these algorithms are computationally efficient and the results obtained have been encouraging.

However, a limitation of this class of network inference approach is that, it assumes data are identically and independently distributed (i.e., *iid*), which implicitly means that they are from a single experimental source. In reality, many real world biological problems sit on multiple sources of information that can be used to predict interactions between genes. For example, there can be multiple microarray data sets from different laboratories available for the same organism, sometimes measured at the same conditions where the main differences lie in the data sampling strategy or measurement technologies. Biologically, it is often plausible to assume that multiple experimental means resulting in the different datasets may have captured the same information from different viewpoints, e.g., both microarray and *in-situ* hybridization can capture gene expression information, even though the technology used to measure mRNA abundances is different. It remains unclear how to integrate such multiple sources of data in a statistically valid and computational efficient way to infer the underlying network. One may imagine inferring independently a network from each data source, and then averaging across multiple resultant networks, but such an *ad hoc* method is not only un-robust (e.g., each view may have only a small amount of samples), but also lacks statistically justification and consistence guarantee (e.g., on the "average" operator). In this paper, we address the question of inferring a network by analyzing multiple sources of information simultaneously.

An alternative approach to tackle this problem is via supervised learning methods, where a classifier (e.g., SVM) is trained by using examples of known gene interactions (edges in the network) as training data to learn the importance of each data source in predicting unknown interactions between other gene-pairs (Ben-Hur and Noble, 2005). This approach suffers from some intrinsic limitations which prevent it from being widely applicable. First, while such an approach works well for problems where there are sufficient examples of known edges in the network, e.g., in the form of a *reference network* or *reference interactions* obtained from reliable sources, it fails for problems where few or no examples of known edges are available. Gene networks for humans or yeast may be learned by supervised methods where reference interactions are available from extensive prior studies; but for organisms where prior research is limited, this approach cannot be used. Furthermore, one can argue that predicting gene networks is of high importance for such organisms with few known edges, to help biologists who are starting research for regulatory mechanisms of these organisms.

Secondly, using a classifier to predict edges implicitly utilizes the notion of *marginal independence* between nodes. To classify an edge as "positive", i.e., to predict an edge between a given pair of nodes, the correlation between the data for these nodes must be high. Gene networks usually have pathways in which genes interact with each other in a sequential order, which results in high marginal correlation between all pairs of genes in the same pathway. Predicting each edge locally and independently of all other edges will often result in an non-stringent prediction of a clique for all genes in the same pathway, leading to high false positive rates. To reduce such false positives and increase accuracy, we wish to analyze *conditional independence* between the genes instead, which must be done by building a global graphical model that captures simultaneously all the conditional independencies among genes. Thus, conditional independence predicts an interaction between two genes in the context of all the other genes in the network, unlike marginal independence, which takes into account only the localized interaction between the genes, without taking into account the other genes that may also be interacting with the given pair of genes. Each edge resultant from such an estimator enjoys *global* statistical interpretability and consistency guarantee, and such an estimator does not require supervised training, although prior knowledge of interactions on the "reference gene pairs" can still be utilized via introducing a prior over the model, if desired. Thus, it is desirable to develop an *unsupervised* and global inference method which can incorporate multiple data sources to predict a consensus graphical model that explains all the data sources, without using any examples of known edges for training the model.

This paper proposes NP-MuScL (NonParanormal Multi-Source Learning), a machine learning technique for estimating the structure of a sparse undirected graphical model that is consistent with multiple sources of data. The multiple data sources are all defined over the same feature space, and it is assumed that they share the same underlying relationships between the genes (nodes). We use the semiparametric Gaussian copula to model the distribution of the different data sources, where the copula for each data source has its own mean and transformation functions, but all data sources share the same precision matrix (i.e., the inverse covariance matrix, which captures the topological structure of the network). We propose an efficient algorithm to estimate such a model in the high dimensional scenario. The likelihood-related objective function used in NP-MuScL is convex, and results in a globally optimal estimator. Furthermore, the implementation of our algorithm is simple and efficient, computing a network over 2000 nodes using 3 data sources in a matter of minutes. Results are reported on synthetic data, where NP-MuScL outperforms baseline algorithms significantly, even in the presence of noisy data sources.

We also use NP-MuScL to estimate a gene network for yeast using two microarray data sets: one over time series expression, and the other over knockout mutants. Finally, we run NP-MuScL on three data sets of Drosophila embryonic gene expression using ISH images and microarray. In both yeast and Drosophila, we find that NP-MuScL predicts a higher number of gene interactions that are known to interact in the literature, than existing techniques.

4.1.1 Related work

Previous work on analyzing multiple data sources for network prediction has either specifically taken time into account (Ahmed and Xing, 2009b; Wang et al., 2006), or has different source and target organisms via transfer learning (Xu et al., 2012). Katenka and Kolaczyk (2012) and Kolar et al. (2013) propose a strategy to learn a network from multi-attribute data, where aligned vector observations are made for each node. The NP-MuScL algorithm on the other hand works for data sources which are not aligned, hence each data source may have a different number of observations. Honorio and Samaras (2011) and Danaher et al. (2013) proposed techniques for multi-task structure learning of Gaussian Graphical Models, to share knowledge across multiple problems, using multi-task learning and fused lasso or group lasso penalties respectively. However, their method estimates a separate graphical model for each data source, unlike our problem which requires a consensus network common to all data sources. To the best of our knowledge, the NP-MuScL algorithm is the first work that builds a consensus graphical model to explain the relationship between genes by combining information from multiple data sources without explicitly constraining the data to be time-series, or about different organisms.

4.2 Nonparanormal Multi-Source Learning (NP-MuScL)

Let the k input data sources be defined as $\mathbf{X}^{(1)} \in \mathbb{R}^{n_1 \times d}$, $\mathbf{X}^{(2)} \in \mathbb{R}^{n_2 \times d}$, $\ldots$, $\mathbf{X}^{(k)} \in \mathbb{R}^{n_k \times d}$ with total number of data samples $n = \sum_{i=1}^{k} n_i$. Each data source i may have a different number of measurements or samples n_i, but they all measure information about the same feature space of d genes. The goal of NP-MuScL is to learn the structure of a graphical model over the feature space, such that the graphical model will encapsulate global conditional independencies between the genes.

4.2.1 Glasso

Given a single source of data $\mathbf{X} \in \mathbb{R}^{n \times d}$ drawn from a Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, a Gaussian graphical model (GGM) may be estimated by computing the inverse covariance matrix $\mathbf{\Sigma}^{-1}$ of the Gaussian. Zeros in the inverse covariance matrix imply conditional independence between the features, and thus the absence of an edge between them in the corresponding GGM. Given the empirical covariance matrix $\mathbf{S}$ of the data, the inverse covariance matrix may be computed by maximizing the log likelihood of the data, with an L_1 regularizer to encourage sparsity.

$$\widehat{\mathbf{\Sigma}}^{-1} = \arg\max_{\mathbf{\Theta} \succ 0} \{\log\det\mathbf{\Theta} - \text{tr}(\mathbf{S}\mathbf{\Theta}) - \lambda||\mathbf{\Theta}||_1\} \tag{4.1}$$

where λ is a tuning parameter that controls the sparsity of the solution; as λ increases, fewer edges are predicted in the GGM. Rothman et al. (2008) showed the consistence of such estimators in Frobenius and Operator norms in high dimensions when $d >> n$; Friedman et al. (2008) proposed a block coordinate descent algorithm for this objective - they named their technique glasso. The glasso algorithm uses a series of L_1 penalized regressions, called Lasso regressions

(Tibshirani, 1996), that can be solved in time $O(d^3)$.

4.2.2 Joint estimation of the GGM

Given k data sources $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots \mathbf{X}^{(k)}$ with corresponding sample covariances $\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, \cdots, \mathbf{S}^{(k)}$, a joint estimator of the underlying GGM may be computed as

$$\widehat{\mathbf{\Sigma}}^{-1} = \arg\max_{\mathbf{\Theta} \succ 0} \sum_{i=1}^{k} w_i \left\{ \log\det \mathbf{\Theta} - \text{tr}(\mathbf{S}^{(i)}\mathbf{\Theta}) \right\} - \lambda ||\mathbf{\Theta}||_1 \tag{4.2}$$

where w_i defines the relative importance of each data source, and must be defined by the user such that $\sum_{i=1}^{k} w_i = 1$. Assuming the data in each data source is drawn i.i.d., an appropriate choice for the weights may be $w_i = \frac{n_i}{n}$. It can be seen that if each data source is assumed to have mean 0, then for this choice of w_i

$$\begin{aligned} \widehat{\mathbf{\Sigma}}^{-1} &= \arg\max_{\mathbf{\Theta} \succ 0} \ \log\det \mathbf{\Theta} - \sum_{i=1}^{k} \frac{n_i}{n} \text{tr}\left(\mathbf{S}^{(i)}\mathbf{\Theta}\right) - \lambda ||\mathbf{\Theta}||_1 \\ &= \arg\max_{\mathbf{\Theta} \succ 0} \ \log\det \mathbf{\Theta} - \text{tr}\left(\frac{1}{n} \sum_{i=1}^{k} \sum_{l=1}^{n_i} \mathbf{X}^{(i)}(l,\cdot)^T \mathbf{X}^{(i)}(l,\cdot) \ \mathbf{\Theta} \right) - \lambda ||\mathbf{\Theta}||_1 \end{aligned} \tag{4.3}$$

Thus, our objective function is equivalent to calling glasso with covariance matrix computed as $\frac{1}{n} \sum_{i=1}^{k} \sum_{l=1}^{n_i} \mathbf{X}^{(i)}(l,\cdot)^T \mathbf{X}^{(i)}(l,\cdot)$. We call this method “glasso-bag of data”. With an appropriate choice of weights, this model concatenates the data from all data sources into a single matrix, and uses the second moment of the data to estimate the inverse covariance matrix.

Such a procedure highlights the underlying assumption of Gaussianity of the data. If we assume that all data is being drawn from the same Gaussian distribution, then it is reasonable to construct a single sample covariance matrix from the data to estimate the network. However, real data is not always Gaussian; and such an assumption can be limiting, especially when analyzing multiple data sources simultaneously, since non-Gaussianity in a single data source will result in the non-Gaussianity of the combined data. A lot of previous work has been done to drop the Gaussianity assumption in the solution to classic problems like sparse regression (Ravikumar et al., 2007), estimating GGMs (Liu et al., 2009), sparse CCA (Balakrishnan et al., 2012) etc., and propose non-parametric solutions to the same. We will also drop the assumption that the data is drawn from the same Gaussian distribution in the next section.

However, if the data is not drawn from the same Gaussian distribution, then how can we characterize the underlying network that generated the data? We propose a generative model where we assume that each data source is drawn from a semi-parametric Gaussian copula, where the copulas for the different data sources share the same covariance matrix, but have different functional transformations. To justify this model, we assume that for each data source, the data is sampled from a multi-variate Gaussian, but this sample is not directly observed. Instead, due to non-linearities introduced during data measurement, a transformed version of the data is measured. Each data source will have its own transformation, hence, the observed distribution of each data source will be different. The key idea of NP-MuScL is then to estimate the non-linear transformation, so that all data can be assumed Gaussian, and the network can be estimated using Equation 4.2.

4.2.3 Dropping the Gaussianity assumption

We model that each data source is drawn from an underlying Gaussian distribution with mean 0, and covariance matrix $\mathbf{\Sigma}$, where the variance of each feature $\sigma_{jj} = 1, \ \forall \ j \in \{1, \cdots, d\}$. However, the observed data may be some unknown transformation of the Gaussian data; thus, if $y \sim \mathcal{N}(0, \mathbf{\Sigma})$, then the observed data is $X^{(i)}(j) = \mu_{ij} + \rho_{ij} g_{ij}(y(j))$ where μ_{ij} and ρ_{ij} is the mean and standard deviation respectively of feature j in data source i.

The function g_{ij} is some (unknown) transformation that depends on the data source, our task is to estimate $f_{ij} = g_{ij}^{-1}$ from the data, so that $f_{ij}(X_j^{(i)})$ is Gaussian. The data generation process is then described in Figure 4.1.

```
Input: True covariance matrix $\mathbf{\Sigma}$ with $\sigma_{jj} = 1 \quad \forall\, j \in \{1, \cdots, d\}$
Input: Transformation function $g_{ij}$, mean $\mu_{ij}$ and variance $\rho_{ij}$ for each feature $j$ for each
       data source $i$.
for $i = 1$ to $k$ do
    for $l = 1$ to $n_i$ do
        $y \sim N(0, \mathbf{\Sigma})$ ;
        for $j = 1$ to $d$ do
            $\mathbf{X}^{(i)}(l, j) = \mu_{ij} + \rho_{ij} g_{ij}(y(j))$ ;
        end
    end
end
Output: Observed data $\mathbf{X}^{(i)}$ from $k$ data sources.
```

Figure 4.1. Data generation model for NP-MuScL

4.2.4 NP-MuScL algorithm

A random vector X has a nonparanormal distribution $NPN(\mu, \mathbf{\Sigma}, f)$ if there exists a function $f(X) = (f_1(X_1), f_2(X_2), \cdots, f_d(X_d))$ such that $f(X)$ has a multi-variate Gaussian distribution $\mathcal{N}(\mu, \mathbf{\Sigma})$ (Liu et al., 2009). To preserve identifiability, we constrain each f_j to have mean 0 and standard deviation 1. The nonparanormal distribution is a Gaussian copula when the fs are monotone and differentiable. For our model, we assume that each data source $X^{(i)} \sim NPN(\mathbf{0}, \mathbf{\Sigma}, f_i)$, that is, while each data source has its own functional transformation, they all share the same underlying relationship between the nodes, represented by $\mathbf{\Sigma}$. The mean of each copula is zero, since we constrain the estimated functions f_j to have zero means. Then, for non-paranormal data, it can be shown that conditional independence in the corresponding graph is

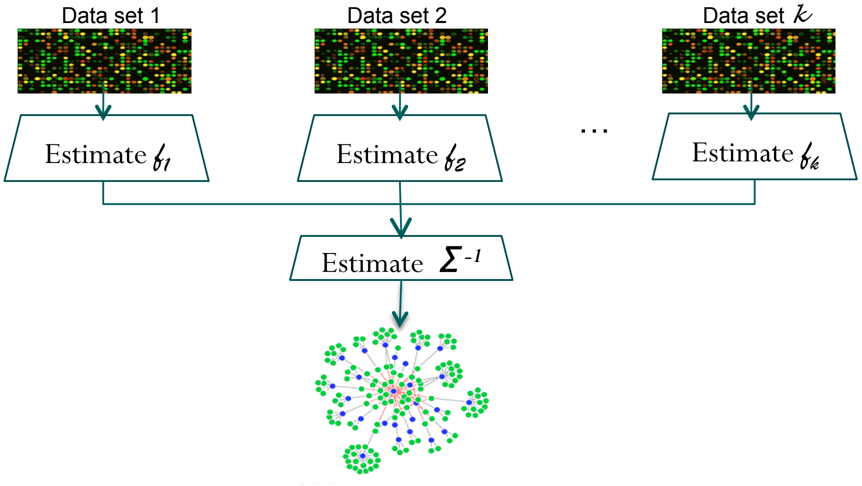


Figure 4.2. The overall algorithm for NP-MuScL. Each data source is transformed into a Gaussian, using a nonparanormal, and the Gaussian data is then used to jointly estimate a inverse covariance matrix, giving the structure of the Gaussian Graphical Model, underlying the data.

equivalent to zeros in the inverse covariance matrix $\mathbf{\Sigma}^{-1}$ (Liu et al., 2009).

This suggests the following two step algorithm. For each data source i and each feature j, we first estimate the sample mean μ_{ij} and sample variance ρ_{ij}.

$$\widehat{\mu}_{ij} = \frac{1}{n_i}\sum_{l=1}^{n_i} \mathbf{X}^{(i)}(l,j); \qquad \widehat{\rho}^2_{ij} = \frac{1}{n_i}\sum_{l=1}^{n_i} \left(\mathbf{X}^{(i)}(l,j) - \widehat{\mu}_{ij}\right)^2 \tag{4.4}$$

The data in each data source is normalized by the appropriate μ and ρ to have mean 0 and standard deviation 1. Non-parametric functions f_{ij} are estimated for each data source i and feature j, so that $f_{ij} \sim \mathcal{N}(0,1)$. The details of estimating f are discussed in Sec. 4.2.6.

In the second step, the inverse covariance matrix is estimated jointly from the transformed f_is. We can define $\mathbf{Y}^{(i)} \in \mathbb{R}^{n_i \times d}$ as

$$\mathbf{Y}^{(i)}(\cdot, j) = \widehat{f_{ij}}\left(\mathbf{X}^{(i)}(\cdot, j)\right) \forall j \in \{1, 2, \cdots d\} \tag{4.5}$$

The distribution of $\mathbf{Y}^{(i)}$ is then Gaussian with covariance matrix $\mathbf{\Sigma}$. The graphical model corresponding to all data sources can be jointly estimated as

$$\widehat{\mathbf{\Sigma}}^{\mathbf{-1}} = \arg\max_{\mathbf{\Theta}\succeq 0} \sum_{i=1}^{k} w_i \left\{ \log\det \mathbf{\Theta} - \text{tr}(\mathbf{\Theta}\ \widehat{\mathbf{S}_\mathbf{f}}^{(\mathbf{i})}) \right\} - \lambda||\mathbf{\Theta}||_1 \tag{4.6}$$

where

$$\widehat{\mathbf{S}_\mathbf{f}}^{(\mathbf{i})} = \frac{1}{n_i} \sum_{l=1}^{n_i} \mathbf{Y}^{(i)}(l, \cdot)^T \mathbf{Y}^{(i)}(l, \cdot) \tag{4.7}$$

Setting the weights $w_i = \frac{n_i}{n}$ is equivalent to the data in each data source being drawn i.i.d. from the corresponding Gaussian copula; while setting different weights suggests that the effective sample size of a data source is not the observed sample size.

4.2.5 Optimization

The objective function in Equation 4.6 can be rewritten as

$$\widehat{\mathbf{\Sigma}}^{\mathbf{-1}} = \arg\max_{\mathbf{\Theta}\succeq 0}\ \log\det \mathbf{\Theta} - \text{tr}(\mathbf{\Theta}\ \sum_{i=1}^{k} w_i \widehat{\mathbf{S}_\mathbf{f}}^{(\mathbf{i})}) - \lambda||\mathbf{\Theta}||_1 \tag{4.8}$$

Thus, by using $\sum_{i=1}^{k} w_i \widehat{\mathbf{S}_\mathbf{f}}^{(\mathbf{i})}$ as the covariance matrix, we can optimize the above objective by using efficient, known algorithms like glasso. The overall NP-MuScL algorithm is summarized in Figure 4.2.

4.2.6 Estimating $\widehat{f}$

For each feature j in data source i, we can compute the empirical distribution function as (where $\mathbb{I}$ is the indicator function)

$$\widehat{F}_{ij}(t) = \frac{1}{n_i}\sum_{l=1}^{n_i}\mathbb{I}(X^{(i)}(l,j) \le t) \tag{4.9}$$

The variance of such an estimate may be very large, when computed in the high dimensional scenario $d >> n$. Liu et al. (2009) propose using a Winsorized estimator, for the same, where very small and large values of $\widehat{F}_{ij}(t)$ are bounded away from 0 and 1 respectively. Thus,

$$\widetilde{F}_{ij}(t) = \begin{cases} \delta_n & \widehat{F}_{ij}(t) < \delta_n \\ \widehat{F}_{ij}(t) & \delta_n \le \widehat{F}_{ij}(t) \le 1-\delta_n \\ 1-\delta_n & \widehat{F}_{ij}(t) \ge 1-\delta_n \end{cases} \tag{4.10}$$

where δ_n is a truncation parameter. A value of δ_n chosen to be $\delta_n = \frac{1}{4n^{1/4}\sqrt{\pi \log n_i}}$ is found to give good convergence properties for estimating the network for a single data source (Liu et al., 2009); and we use the same estimate for NP-MuScL.

Now, for any continuous pdf f, the distribution of the cdf $F(x) = P(X \le x)$ is uniform. Then, the distribution of $\Phi^{-1}(F(x))$ is Gaussian with mean zero, and standard deviation one, as required (where Φ is the cdf of the standard Gaussian). Thus, we can estimate the required function by using the marginal empirical distribution function defined above: $\widehat{f}_{ij}(x) = \Phi^{-1}(\widetilde{F}_{ij}(x))$.

4.3 Results

We first demonstrate that when multiple data sources have different distributions, NP-MuScL can extract the underlying network more accurately than other methods. Next, we show that NP-MuScL can identify the correct network, even when one of the data sources is noise. To analyze NP-MuScL on real data, we run NP-MuScL on two microarray yeast data sets, and find that the network obtained by NP-MuScL predicts more known edges of the yeast interaction network than other methods. Finally, we analyze NP-MuScL on Drosophila embryonic gene expression data from 3 data sets of ISH images and microarray.

4.3.1 Multiple data sources with different distributions

Data generation We construct an inverse covariance matrix with an equivalent random sparse Gaussian graphical model with a known structure and parameters. Data is sampled from the Gaussian, and then transformed into non-Gaussian distribution using different transformations.

Synthetic network We generate synthetic data by first sampling a sparse inverse covariance matrix, as described in Meinshausen and Bühlmann (2006).

The key idea behind generating a network is: associate each node in the network with a random point in space. Then, add edges so that "nodes that are closer to each other are more likely to have an edge between them". In this simulation, we pick random points in 2D space (as suggested by Meinshausen and Bühlmann (2006)), but in general, each node can be associated with a random point in any n-dimensional space.

Each node is associated with a point uniformly at random in the two dimensional square $[0, 1]^2$, and an edge is included with probability $\Phi(y/\sqrt{d})$, where y is the Euclidean distance between the two nodes. The maximum degree of a node is set to 4; if more than 4 edges are generated by the above procedure, they are discarded to maintain the maximum degree. The sparse inverse covariance matrix is generated by setting the diagonal elements to 1; the non-diagonal elements to 0.245 if an edge is present, and 0 otherwise (this value guarantees than the matrix is diagonal-dominant).

Transformations We use two transformations to generate non-Gaussian data (Liu et al., 2009):

1. Gaussian cdf: Let g_0 be the one-dimensional Gaussian cdf with mean μ_0 and std dev σ_0. The Gaussian cdf transformation function for the j^{th} dimension $g_j = f_j^{-1}$ is then defined as

$$g_j(z_j) = \sigma_j \left(\frac{g_0(z_j) - \int g_0(t)\Phi(\frac{t-\mu_j}{\sigma_j})dt}{\sqrt{\int \left(g_0(y) - \int g_0(t)\Phi\left(\frac{t-\mu_j}{\sigma_j}\right)dt \right)^2 \Phi\left(\frac{y-\mu_j}{\sigma_j}\right)dy}} \right) + \mu_j \tag{4.11}$$

2. Power transform: Let $g_0 = sign(t)|t|^\alpha$ be the symmetric transform. Then the power transformation for the j^{th} dimension can be defined as

$$g_j(z_j) = \sigma_j \left(\frac{g_0(z_j - \mu_j)}{\sqrt{\int g_0^2(t-\mu_j)\Phi(\frac{t-\mu_j}{\sigma_j})dt}} \right) + \mu_j \tag{4.12}$$

For $d = 50$ and $d = 200$ with $k = 2$ data sources, we use the Gaussian cdf ($\mu_0 = 0.05, \sigma_0 = 0.4$) and power transform ($\alpha = 3$) for the two data sources respectively. The task then is to jointly use the data from the two sources to extract the network. For $k = 3$ data sources, we use the identity transform for the third data source, so that the data sampled from the third source is truly Gaussian. For $k = 4$ data sources, the fourth data source is Gaussian noise, to test the performance of the algorithms in the presence of noise. We generate the same amount of data in each source (n), and run the experiment as n varies. Each result is reported as the average of 10 randomized runs of the experiment.

Metrics We report the $F1$ measure, which is the harmonic mean of precision and recall, as a measure of the accuracy of predicting the edges in the network. The maximum and minimum value are 1 and 0 respectively, with higher values representing better performance (Goutte and Gaussier, 2005).

Baselines We report three baselines. The first baseline is to report the best accuracy found by a single data source (Best Single Network). We assume that an oracle tells us which data source is most predictive. In our data experiments, we found that it was not possible to predict the most informative data source without using an oracle. Even when $k = 3$, the identity transformed source was not always the most informative. The second baseline is the glasso-bag of data, described in Section 4.2.2. The third baseline is to compute a separate network for each data source using glasso, and combine the networks to predict a single network (glasso-combine networks). An edge in the final network is present if it is present in m out of the k networks from the k data sources. We assume an oracle defines the best value of m for a given data set, the best value of m varied with different data sets.

As can be seen in Figure 4.3, NP-MuScL outperforms all three baselines significantly in all three scenarios. Interestingly, using the best single source outperforms estimating separate networks, and combining them in a second step. Note that an oracle is used for identifying the best source, as well as the optimal m used to combine networks. Hence, in a real world scenario, we may expect combining different data sources to perform as well as using only the best single data source for network prediction. When $k = 4$ (Figure 4.3(c) and 4.3(f)), one of the data sources is Gaussian noise, however, the use of the oracle in the "Glasso-combine networks" and the "single best source" baselines allows these baselines to ignore the noise source completely. However, NP-MuScL is still able to identify more correct edges in the network. Using a paired t-test, we found that the difference in $F1$ scores between NP-MuScL and “glasso-bag of data” is significant in all conditions, with P-value $p = 10^{-4}$.

4.3.2 Effect of varying size of data sources

In the previous experiment, we assumed that each data source has equal amount of data, i.e. n_i is the same for each data source i. Next, we vary the amount of data available for different data sources, and study the effect of the same on our results. For $d = 200$ dimensional data, we again generate simulated data for 3 data sources, with a total of 300 data samples across all 3 sources. Data source 1 has Gaussian data (identity transform) with 100 i.i.d. data samples, data source 2 has CDF transformed data with m samples, and data source 3 has power transformed data with $(200 - m)$ samples. We vary m to study the effect of data sources having different amounts of data. Figure 4.4 shows that even when the three data sources are of different sizes, NP-MuScL performs significantly better than the other algorithms. When the amount of data in data source 2 is very small, a network learned from this data source alone has poor predictive performance, and the “Glasso-combine networks” method performs poorly since it incorporates this low precision network in its predictions. “Glasso-bag of data” also drops its performance as m reduces, since this results in increasing importance to be placed on predictions from data source 3, which now has more data and hence a higher weight $w_i = \frac{(200-m)}{300}$.

4.3.3 Yeast data

In this experiment, we look at two different yeast microarray data sets, and make joint predictions via NP-MuScL. Data source 1 is a set of 18 expression profiles from Cho et al. (1998), where each expression corresponds to a different stage in the cell cycle of the the yeast. Data source 2 is a set of 300 expression profiles from Hughes et al. (2000), where each expression corresponds to a different knockout mutant of the yeast. Both data sets are processed using standard microarray processing algorithms (Hibbs et al., 2007).

We use a list of known interactions from BioGrid (Stark et al., 2011) to test how well do the different algorithms predict the known edges. Note that since the known gene interactions is an incomplete set, predicted gene interactions may be interactions that have not been observed yet, and thus, have not been added to the BioGrid data base. Hence, measuring recall is no longer appropriate, and we report the improvement in accuracy over random prediction of edges, as suggested by Liben-Nowell and Kleinberg (2003). "Improvement over random guessing" is a dimensionless quantity, a value of one means that the algorithm is doing as poorly as randomly guessing the edges in the network, a value of m means that the algorithm does m times better in accuracy than randomly guessing edges.

The total data is over 6120 genes, we sample 1000 genes at a time, and run the algorithms for them. Results are reported for 10 random sub-samples of the genes. Figure 4.5 shows the improvement over random prediction for edges predicted by each method. Due to the amount of data available, the knockout mutant expression profiles capture more information (and hence more known edges) than the time series expression. Surprisingly, both methods of combining information without taking non-Gaussianity into account, perform worse than using only data source 2. NP-MuScL is the only method where using both data sets into account increases the number of correctly predicted edges. The same results were found to hold true when the network is predicted over the entire set of 6120 genes - NP-MuScL did significantly better than all other methods, and both glasso bag-of-data and glasso-OR did worse than using only data set 2.

To test the effect of varying tuning parameter λ, Figure 4.6 plots the number of known edges predicted by each method, versus the total number of edges predicted, as λ is varied. For very large values of λ when few edges are predicted, NP-MuScL and "glasso-Bag of data" perform equally well, however, as the amount of predictions increase, NP-MuScL outperforms other methods significantly.

Figure 4.7 shows the transformations learned for the two data sets by NP-MuScL for 4 random genes. A straight line corresponds to Gaussian data, non-linearities are clearly detected by the NP-MuScL algorithm. The transformations also seem to be damping extremely large values observed in the features.

4.3.4 Drosophila embryonic data

We study three data sets of Drosophila embryonic gene expression for 146 genes (Tomancak et al., 2002a). The first data set measures spatial gene expression in embryonic stage 9-10 of Drosophila development via in-situ hybridization (ISH) images (4.3 to 5.3 hours after fertilization), when germ band elongation of the embryo is observed. The second data set also studies ISH images measuring spatial gene expression in the 13-16 stage of embryonic development (9.3 to 15 hours after fertilization), when segmentation has already been established. The last data set is of microarray expression at 12 time points spaced evenly in embryonic development.

The ISH images were processed to extract 311 data points for each data set, as described in Puniyani and Xing (2012). The microarray data was processed using standard microarray processing algorithms (Hibbs et al., 2007). Since the number of data points extracted from the ISH data is dependent on the image processing algorithm used, using weights proportional to the number of data points is no longer suitable. We expect the microarray data to be as informative as the ISH data, hence we use $w_i = 0.25$ for each of the two ISH data sources, and $w_i = 0.5$ for the microarray data. The results in Table 4.1 show that NP-MuScL outperforms using the data separately, and glasso bag-of-data and glasso-combine networks (m=1, called glasso-OR).

We visualized the differences in edge prediction between the NP-MuScL network and the networks predicted by analyzing only one single data source at a time. The orange ellipse in Fig-

NP-MuScL	Glasso Bag-of-data	Glasso OR	ISH 13-16	ISH 9-10	Microarray
7.29	4.88	4.06	5.98	2.35	3.66

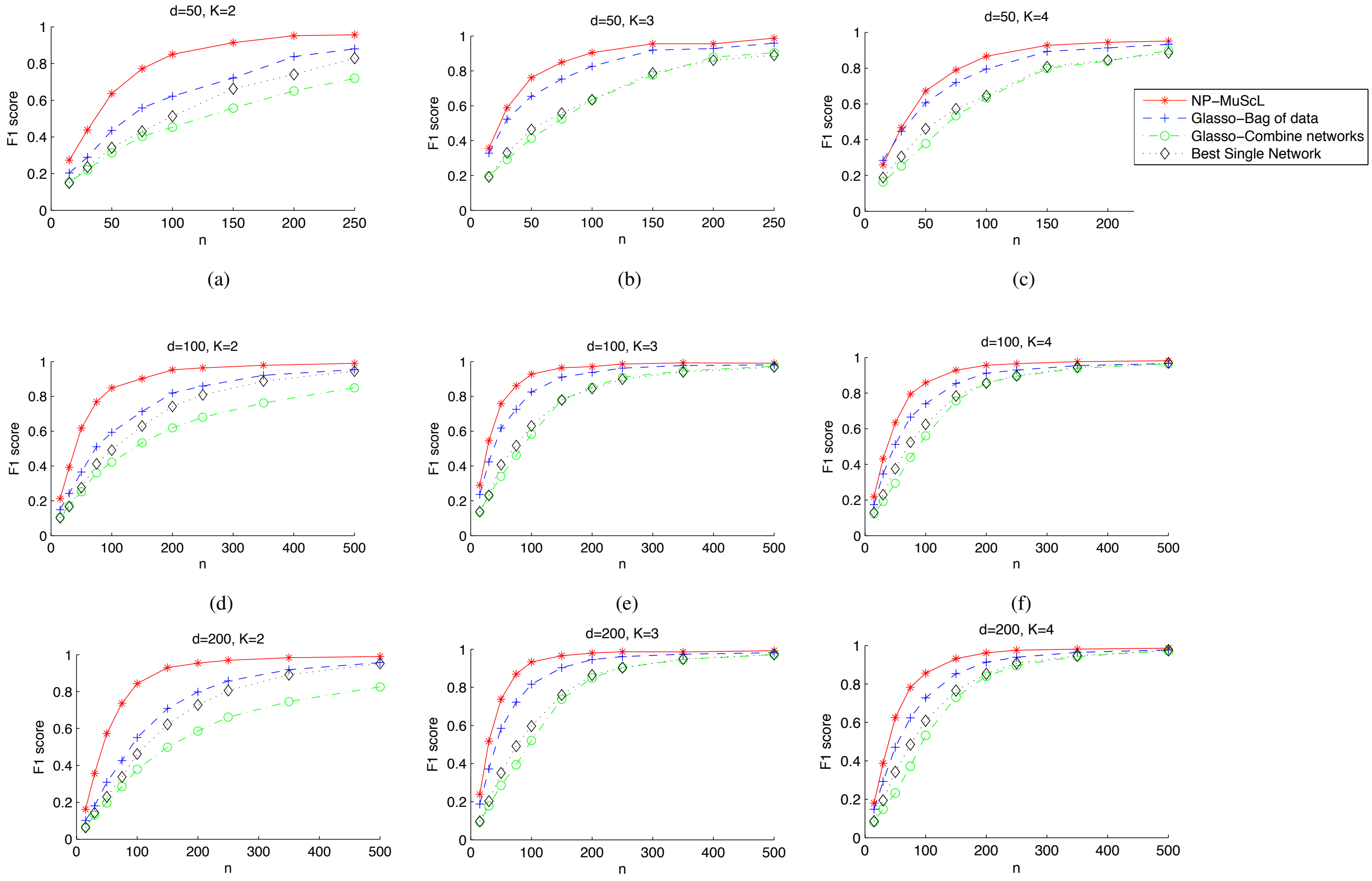
Table 4.1. Improvement in prediction over random guessing for predicting gene interactions using Drosophila embryonic data.

ure 4.8(a) highlights gene interactions predicted by NP-MuScL by analyzing all 3 data sources, which were not predicted by any single data source. Figure 4.8(b) highlights interactions predicted by the microarray data that were not predicted either by the ISH data or the NP-MuScL network. The 9-10 ISH network is similar to the 13-16 ISH network, and hence, is not shown. A detailed analysis of the specific differences in the gene interactions predicted by the different methods is ongoing.

4.4 Discussion

We proposed NP-MuScL, an algorithm that predicts gene interaction networks in a global, unsupervised fashion by jointly analyzing multiple data sources to capture the conditional independencies observed in the data. NP-MuScL models each data source as a non-parametric Gaussian copula, with all data sources having different mean and transformation functions, but sharing the covariance matrix across the underlying copulas. The network can then be efficiently estimated in a two step process, of transforming each data source into Gaussian, and then estimating the inverse covariance matrix of the Gaussian using all data sources jointly. We found that NP-MuScL significantly outperforms baseline methods in both synthetic data, and two experiments predicting a gene interaction network from two yeast microarray data sets, and three Drosophila ISH images and microarray data sets.

One limitation of NP-MuScL is that the weights giving the importance of each data source must be assigned by the user. While a good estimate of the weights may be obtained if all data sources are truly drawn i.i.d. from their nonparanormal distributions, and have similar noise levels; in practice, some data sources may be known to be noisier than others, or known to not be i.i.d. (e.g.. microarray experiments over time are not truly independent draws from the distribution). The question of automatically learning the weights from data remains an open challenge.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.3. ***$F1$* score for predicting edges in simulated data.** The results are plotted as n is varied, for $k = 2$, $k = 3$, and $k = 4$ data sources for $d = 50$ ((a) - (c)), $d = 100$ ((d) - (f)), and $d = 200$ ((g) - (i)) dimensions. The standard deviation in the results is small and almost constant across the different experiments; it ranges from (0.01-0.03), and is hence not displayed on the plot. Note that the X-axis values for $d = 50$ stops at a lower value of n than for $d = 100$ and $d = 200$.

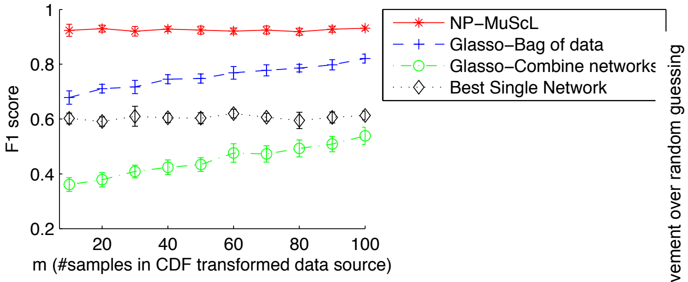


Figure 4.4. $F1$ **score as the amount of data(m) in the data source with CDF transform is varied.** The total amount of data is kept constant by using $(200 - m)$ samples in the data source with power transform. The data source with an identity transform has fixed 100 samples in all experiments.

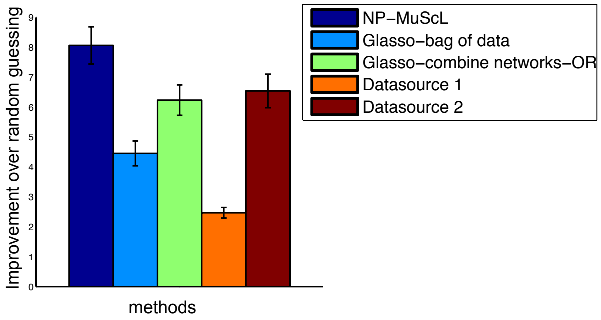


Figure 4.5. Performance of different methods on predicting edges in the yeast network. Only NP-MuScL is able to jointly use the two data sources to obtain better performance than using a single data source alone.

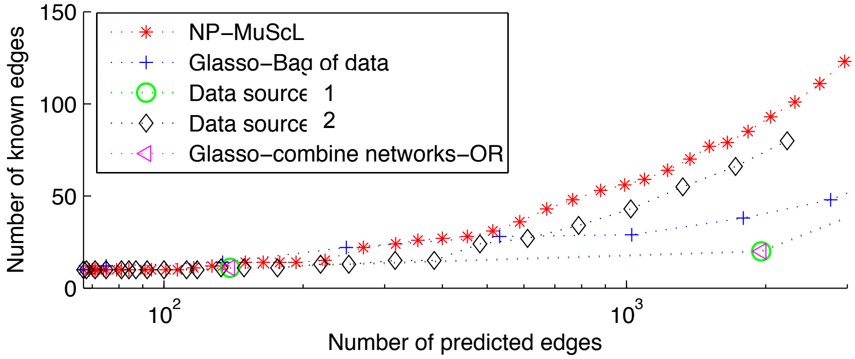


Figure 4.6. Effect of varying tuning parameter on different methods. For a fixed number of predicted edges, the NP-MuScL method predicts more known edges than the other methods.

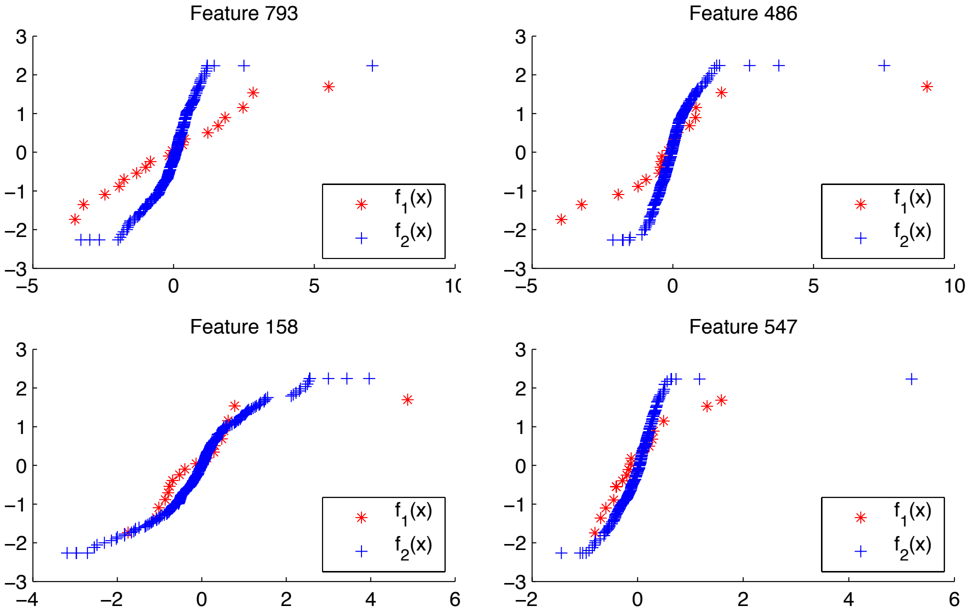


Figure 4.7. Examples of the transformations made for data in source 1 (red) and source 2 (blue) for different features.

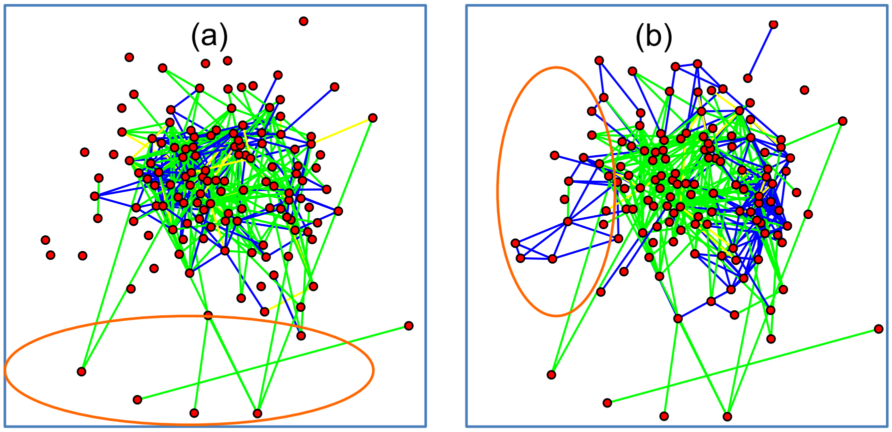


Figure 4.8. Difference between the NP-MuScL network and (a) the 13-16 ISH network alone and (b) microarray network alone. Green edges are only predicted in the NP-MuScL network. Blue edges are only present in the (a) 13-16 ISH network and (b) microarray network.

Chapter 5

Analysis of the entire BDGP data set

The techniques developed in this thesis were applied on smaller subsets of the BDGP data, due to availability issues (a smaller subset was accessible to us when we started working on the problem in 2008). To verify that the algorithms generalize well to the larger data set, we repeat some of the key experiments on the entire data set, and report these results in this chapter. The data set we work with was obtained in late 2011, and contained 108423 images of 7382 genes. Figure 5.1 shows a histogram of the number of images available per gene, over all 6 time points. We notice that most genes have very few images - 198 of these genes have exactly 1 image over the 6 time points, and 2455 of them have exactly 2 images over the 6 time points.

The data collection process for BDGP, which was started in 2000, is still on-going. These numbers suggest that a large scale analysis of the variation in gene expression, and gene networks over time will be possible only after more data has been collected. However, we will still be able to construct gene networks over smaller subsets of the genes, for which more data is available.

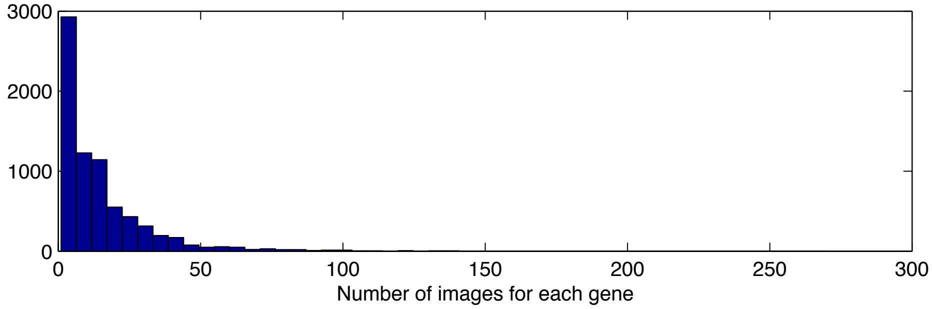


Figure 5.1. Histogram of number of images per gene. The counts on the Y-axis represent the number of genes that have X images available, over all 6 time points in the BDGP data. The maximum number of images available for a single gene is 270 for gene CG16738 (*sloppy paired 1*), and 248 images for gene CG17390 (*O/E-associated zinc finger protein*).

5.1 Feature extraction

We ran the SPEX^2 feature extraction pipeline on the images, and extracted 69420 feature patterns out of them. Effectively, we extracted patterns from 64% of the total images. To test the reasons for rejection of images, we looked at the rejected images, and categorized the reason for rejection

1. No outline of any embryo could be found for 333 images.
2. The outline obtained was too small and too big for 1865 and 4887 images respectively.
3. The extracted ellipse was too circular, suggesting errors in extracting the correct boundary (often, this means that the extract boundary encapsulated two touching embryos).
4. The extracted embryo boundary differs dramatically from the convex ellipse outline fit to it, suggesting highly non-convex regions in the embryo.
5. The fitted convex outline to the embryo was much larger than the segmented embryo, suggesting that the embryo extraction extracted an outline that has regions that were highly non-convex.

Figure 5.2 shows the number of images rejected for each of these reasons. Given that the original SPEX^2 pipeline was designed and tested using only $\sim$ 2000 images, we observe that many of the

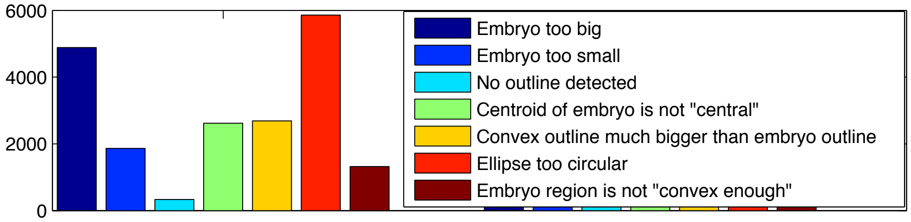


Figure 5.2. Reasons why an image is rejected by SPEX^2. The counts on the Y-axis represent the number of images that were rejected for various reasons by the SPEX^2 pipeline, over all 6 time points in the BDGP data.

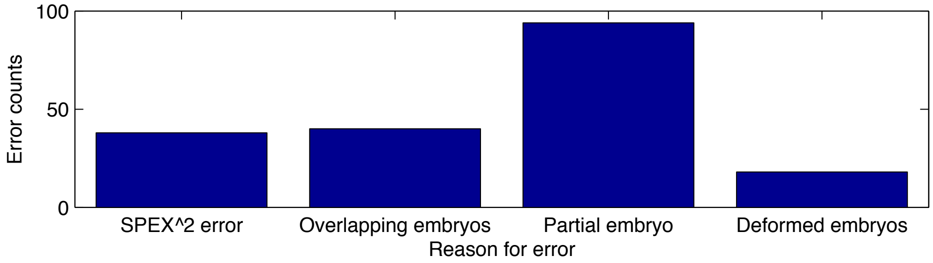


Figure 5.3. Manual error analysis. Categorization of the reason why SPEX^2 did not extract any expression pattern from a given image.

images in the large data set have the same problems encountered in the small data set. To test if images were rejected correctly, we manually looked at 200 random images that were rejected, and assign the validity of rejection by SPEX^2 (Figure 5.3).

In the initial analysis, we assigned 4 reasons for rejecting images

1. Rejected since the image had only a partial embryo, which cannot be handled in downstream analysis.
2. Rejected due to overlapping or touching embryos in the image, that the SPEX^2 pipeline could not extract a single embryo from, and hence rejected the image.
3. Rejected due to surprising shape of embryo, suggesting damage to embryo, multiple embryos in the image that don't capture details,
4. One single embryo in the image, that is not deformed, hence should not have been rejected by SPEX^2.

The first three reasons are inherent to the data, (though further work into extracting 1 embryo from multiple touching embryos may help), while embryos rejected for category 4 would be errors made by SPEX^2. Figure 5.3 shows us that most of the rejected embryos are partial embryos, while the number of errors made by SPEX^2 are quite small. Further, in further analysis of the types of images rejected by SPEX^2, we found that out of the 38 embryos classified as "should not have been rejected", 11 did not have any gene expression pattern, hence, the outline detection did not extract correct boundaries of the embryo, even though it did detect the correct region. The distorted boundaries led to failures of the tests in SPEX^2, that check not just whether we found the embryo, but also, if the extracted boundary is precise. Further, 23 of the embryos were not imaged at the correct focus, leading to blurry images, that again confused the outline detection algorithm. Rejecting incorrectly focused images is an unexpected, but good side-effect of the SPEX^2 pipeline. Figure 5.4 shows example images rejected by SPEX^2.

5.2 Classification

To test the SPEX^2 features, we again test the annotation accuracy for spatial annotations for the images, as discussed in Section 2.3.1. The original results were reported on 2689 images from a single development stage, and the accuracy reported was around 80%. We now test the same classification task on the larger data set of $\sim$ 70000 images, for all 6 development stages. Since the annotations are different for each development stage, we report results separately on each of the 6 development stages. The task, evaluation method, etc. are the same as described in Section 2.3.1. Figure 5.5 shows that we get high prediction accuracy in both classification accuracy and mean AUC (area under the curve) for all 6 time stages. This verifies that SPEX^2 extracts meaningful features on a larger data set, and the system was not over fit to the smaller data set we originally tested on. As an aside, the results are reported by using the SVM classifier. The Local Regularization classifier (Ji et al., 2009) that was designed for this task, and had very good

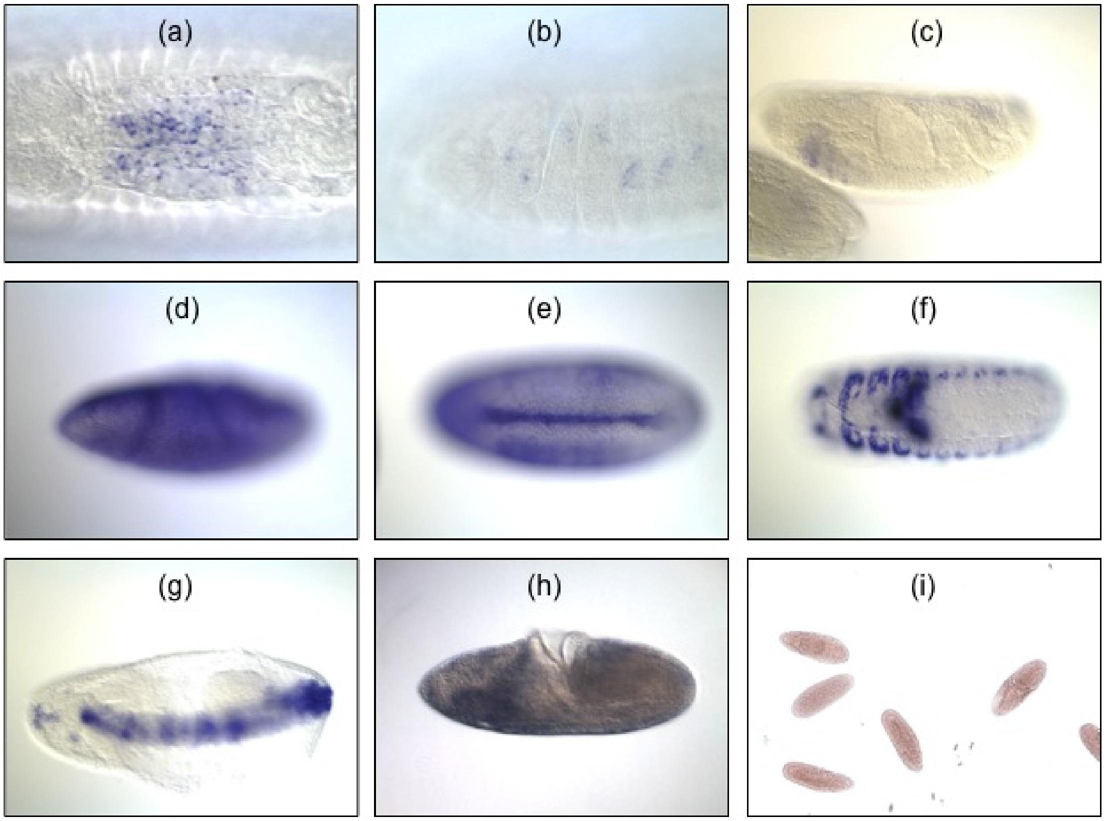


Figure 5.4. Examples of images rejected by SPEX^2. (a)-(b) partial embryos, (c) multiple overlapping embryos, (d)-(f) images rejected due to incorrectly focused images, (g)-(h) deformed embryos and (i) multiple embryos without sufficient detail.

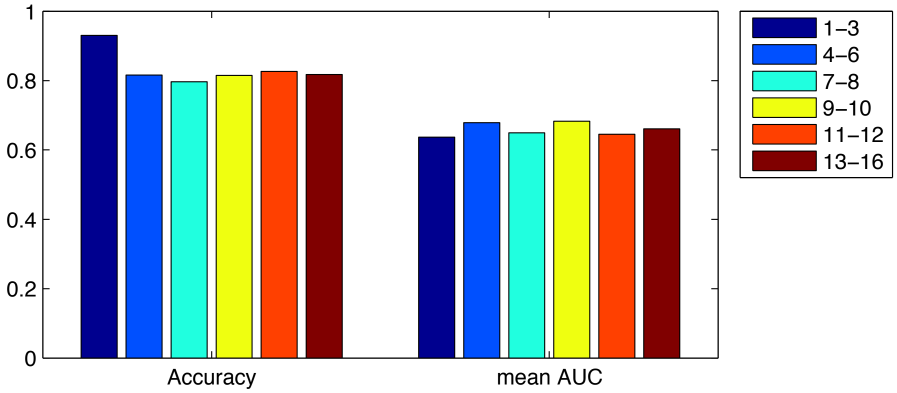


Figure 5.5. Annotation accuracy and mean Area Under the Curve (AUC) score for each of the 6 development stages in the BDGP data

results on the smaller data set did not perform as well on the larger data set. While the AUC scores were as high as using an SVM classifier, the accuracy scores were much smaller (< 0.4), suggesting that the classifier is unable to deal with rare classes in the larger data set.

5.3 Importance of each step in the pipeline

Since SPEX^2 is a complex image processing pipeline, we wish to ascertain the role of each step in the pipeline in obtaining out results. To test this, we extracted features by replacing specific modules in SPEX^2 by simpler modules as follows.

1. **No image segmentation** In this baseline, we do not remove the background noise from the expression pattern by using image segmentation. Instead, the stain extracted in Section 2.2.2 is directly converted into features.

2. **Grey scale color** In Section 2.2.2, we discussed a specific algorithm proposed by us to convert the RGB image into a representation of the stain in the image. In this baseline, instead of using this algorithm, the RGB image is converted into greyscale, and the greyscale values are used as a representation of the stain, which is then segmented to extract the ex-

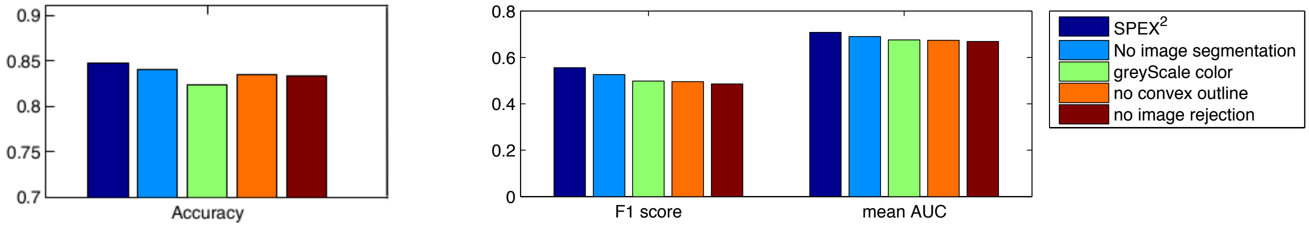


Figure 5.6. Effect of each step in the pipeline. We show annotation accuracy, F1 score, and mean AUC score for stage 13-16 for SPEX^2, and 4 baselines derived by omitting various steps in the SPEX^2 pipeline. The difference in accuracy, and F1 score is statistically significant at P-value = 0.01, except for the difference in SPEX^2, and the "no image segmentation baseline".

pression pattern.

3. **No convex outline** In Section 2.2.1, we discussed how a convex outline of the embryo is registered to a standard elliptical shape. In this baseline, we directly use the extracted embryo outline, instead of fitting a convex boundary to the embryo shape. Not computing the convex outline means that small errors in boundary detection (a hard problem in vision) will result in distortions of the extracted pattern.

4. **No image rejection** In Section 2.2.1, we discussed various tests that SPEX^2 runs to automatically reject some images to reduce errors. In this baseline, such tests are not performed, and all images are converted into expression patterns.

Figure 5.6 shows the reduction in accuracy as each step in the pipeline is replaced by a simpler algorithm. Each result is the average of 10 annotation tasks. The difference in accuracy and F1 score is statistically significant (using a paired t-test) at P-value = 0.01, for each step in the pipeline, except the "no image segmentation" baseline. The image segmentation module in the SPEX^2 pipeline removes noise, hence, one reason for the similar performance with and without the image segmentation could be because the SVM classifier is able to handle the added noise when we do not segment the image to remove the background. We would still argue to keep the image segmentation component of the pipeline, since (a) it may lead to improved performance in the unsupervised task of network prediction, and (b) segmented image patterns may be easier to analyze and visualize on a larger scaler. For example, if we use the non-segmented patterns to cluster the image, and plot the mean image of each cluster (as in Figure 2.11), we find that using the non-segmented images produces visibly noisier image patterns as the mean of each cluster, than if we used the SPEX^2 expression pattern output.

5.4 Learning a network from multiple data sources

Finally, we turn our attention to applying NP-MuScL to our bigger data set. From our smaller experiments, we observed that the image data from development stage 13-16 is more informative than the image data from development stage 9-10. One possible reason for this could be that the more developed embryos in later stages of embryonic development have more informative gene expression patterns. Due to the large amount of missing data (more than half the genes have 2 or less images across the 6 time stages), we can only analyze a smaller subset of genes and development stages of Drosophila. In this experiment, we choose to analyze ISH data from the development stage 11-12 and 13-16, along with microarray data, over $n = 600$ genes for which data is available for these data sources.

5.4.1 Known gene interactions

We predict networks with 2000 edges, and using a list of known interactions from BioGrid (Stark et al., 2011), we test how well do the different algorithms predict the known edges. To test performance of the different methods, we report the improvement in accuracy over random prediction of edges, as suggested by Liben-Nowell and Kleinberg (2003). We find that NP-MuScL is capable of predicting more known edges than other baseline methods. However, due to the difficulty of the problem itself, and the small number of known genetic interactions for Drosophila, the

NP-MuScL	Glasso Bag-of-data	Glasso OR	ISH 11-12	ISH 13-16	Microarray
6.066	2.574	2.225	1.658	2.574	2.386

Table 5.1. Improvement in prediction accuracy over random guessing for predicting gene interactions using Drosophila embryonic data.

absolute improvement over random guessing is small for all methods.

5.4.2 Clustering annotation terms

We expect genes having similar spatial annotations to be clustered within the network. We test this hypothesis by clustering the NP-MuScL network, and running an enrichment analysis for each cluster, testing if specific annotations appear in a cluster with frequency higher than would be expected by chance. The experimental setup is the same as described in Section 3.3.8, with 15 clusters found using spectral clustering, and the enrichment analysis done using the hypergeometric test(with Bonferroni correction used to correct for multiple hypothesis tests). We observed that spectral clustering produced 9 small clusters with only 1-2 genes, and only 6 clusters with 5 or more genes in them, and restricted our analysis only the clusters that have at least 5 genes.

Unlike in Section 3.3.8 that analyzed a network learned from data at a single time point, we now have annotations from 2 development stages, both of which were used to learn the network. We run a separate enrichment analysis for annotations from both development stages. The overall process is: the NP-MuScL network was clustered, small clusters were rejected, and for each cluster with at least 5 genes, we run enrichment analysis for annotations made at stage 11-12 and at stage 13-16. We found that all 6 clusters were enriched for terms from both development stages (Table 5.2 and 5.3). In contrast, when the network was learnt by using the “Glasso-Bag of data” algorithm, only 4 clusters were enriched.

Cluster	Gene Ontology term	Cluster frequency	Genome frequency	P-value
1	anterior midgut primordium	52 of 112 genes, 46.43%	1051 of 4237 genes, 24.81%	9.09098e-06
	brain primordium	44 of 112 genes, 39.29%	767 of 4237 genes, 18.10%	3.05368e-06
	head mesoderm primordium	24 of 112 genes, 21.43%	315 of 4237 genes, 7.43%	2.75141e-05
	posterior midgut primordium	52 of 112 genes, 46.43%	1104 of 4237 genes, 26.06%	3.35944e-05
	somatic muscle primordium	17 of 112 genes, 15.18%	297 of 4237 genes, 7.01%	0.0233243
	strong ubiquitous	10 of 112 genes, 8.93%	36 of 4237 genes, 0.85%	9.29689e-07
	trunk mesoderm primordium	38 of 112 genes, 33.93%	631 of 4237 genes, 14.89%	7.91628e-06
	ubiquitous	66 of 112 genes, 58.93%	812 of 4237 genes, 19.16%	4.83252e-19
	ventral nerve cord primordium	36 of 112 genes, 32.14%	712 of 4237 genes, 16.80%	0.000591857
2	amnioserosa	18 of 85 genes, 21.18%	197 of 4237 genes, 4.65%	9.82781e-07
	anterior midgut primordium	44 of 85 genes, 51.76%	1051 of 4237 genes, 24.81%	1.20414e-06
	dorsal epidermis primordium	18 of 85 genes, 21.18%	351 of 4237 genes, 8.28%	0.00146043
	dorsal pharyngeal muscle primordium	14 of 85 genes, 16.47%	180 of 4237 genes, 4.25%	0.000118361
	fat body/gonad primordium	10 of 85 genes, 11.76%	125 of 4237 genes, 2.95%	0.00150384
	foregut primordium	28 of 85 genes, 32.94%	315 of 4237 genes, 7.43%	2.28828e-10
	hindgut proper primordium	30 of 85 genes, 35.29%	554 of 4237 genes, 13.08%	1.60408e-06
	longitudinal visceral mesoderm primordium	7 of 85 genes, 8.24%	108 of 4237 genes, 2.55%	0.0324033
	posterior midgut primordium	45 of 85 genes, 52.94%	1104 of 4237 genes, 26.06%	1.48091e-06
	salivary gland body primordium	26 of 85 genes, 30.59%	269 of 4237 genes, 6.35%	2.28828e-10
	tracheal primordium	16 of 85 genes, 18.82%	181 of 4237 genes, 4.27%	4.88977e-06
	trunk mesoderm primordium	36 of 85 genes, 42.35%	631 of 4237 genes, 14.89%	2.35953e-08
	visceral muscle primordium	17 of 85 genes, 20.00%	272 of 4237 genes, 6.42%	0.000206696
	yolk nuclei	19 of 85 genes, 22.35%	222 of 4237 genes, 5.24%	9.82781e-07
3	trunk mesoderm primordium	26 of 57 genes, 45.61%	631 of 4237 genes, 14.89%	1.42576e-06
	somatic muscle primordium	16 of 57 genes, 28.07%	297 of 4237 genes, 7.01%	3.69239e-05
	brain primordium	22 of 57 genes, 38.60%	767 of 4237 genes, 18.10%	0.00293397
	dorsal epidermis primordium	16 of 57 genes, 28.07%	351 of 4237 genes, 8.28%	0.00017718
	head epidermis primordium P1	6 of 57 genes, 10.53%	73 of 4237 genes, 1.72%	0.00500391
	hindgut proper primordium	19 of 57 genes, 33.33%	554 of 4237 genes, 13.08%	0.0010242
	strong ubiquitous	4 of 57 genes, 7.02%	36 of 4237 genes, 0.85%	0.0122736
	ubiquitous	30 of 57 genes, 52.63%	812 of 4237 genes, 19.16%	1.42576e-06
	ventral epidermis primordium	15 of 57 genes, 26.32%	294 of 4237 genes, 6.94%	0.00013097
	visceral muscle primordium	14 of 57 genes, 24.56%	272 of 4237 genes, 6.42%	0.00017718
4	salivary gland body primordium	3 of 3 genes, 100.00%	269 of 4237 genes, 6.35%	0.029629
5	embryonic central brain glia	3 of 5 genes, 60.00%	113 of 4237 genes, 2.67%	0.00936417
	neuroblasts of ventral nervous system	3 of 5 genes, 60.00%	125 of 4237 genes, 2.95%	0.00936417
	procephalic neuroblasts	3 of 5 genes, 60.00%	109 of 4237 genes, 2.57%	0.00936417
6	anterior midgut primordium	99 of 143 genes, 69.23%	1051 of 4237 genes, 24.81%	6.112e-28
	posterior midgut primordium	100 of 143 genes, 69.93%	1104 of 4237 genes, 26.06%	3.45418e-27
	Malpighian tubule primordium	17 of 143 genes, 11.89%	190 of 4237 genes, 4.48%	0.0019426
	dorsal pharyngeal muscle primordium	31 of 143 genes, 21.68%	180 of 4237 genes, 4.25%	2.28314e-13
	head mesoderm primordium	33 of 143 genes, 23.08%	315 of 4237 genes, 7.43%	2.32371e-08
	hindgut proper primordium	56 of 143 genes, 39.16%	554 of 4237 genes, 13.08%	3.66025e-14
	muscle system primordium	33 of 143 genes, 23.08%	224 of 4237 genes, 5.29%	2.52148e-12
	somatic muscle primordium	36 of 143 genes, 25.17%	297 of 4237 genes, 7.01%	5.19369e-11
	trunk mesoderm primordium	67 of 143 genes, 46.85%	631 of 4237 genes, 14.89%	6.60219e-19
	ubiquitous	64 of 143 genes, 44.76%	812 of 4237 genes, 19.16%	1.66609e-11
	visceral muscle primordium	24 of 143 genes, 16.78%	272 of 4237 genes, 6.42%	0.000110534

Table 5.2. Enrichment analysis for the NP-MuScL network, using the spatial annotations made for the genes in the network in development stage 11-12.

Cluster	Gene Ontology term	Cluster frequency	Genome frequency	P-value
1	embryonic brain	57 of 112 genes, 50.89%	1092 of 4237 genes, 25.77%	4.40035e-07
	embryonic midgut	57 of 112 genes, 50.89%	1401 of 4237 genes, 33.07%	0.00144194
	embryonic/larval somatic muscle	20 of 112 genes, 17.86%	362 of 4237 genes, 8.54%	0.0217912
	embryonic/larval visceral muscle	25 of 112 genes, 22.32%	375 of 4237 genes, 8.85%	0.000290122
	ubiquitous	54 of 112 genes, 48.21%	547 of 4237 genes, 12.91%	3.14719e-18
	ventral nerve cord	53 of 112 genes, 47.32%	1102 of 4237 genes, 26.01%	2.93825e-05
2	embryonic midgut	50 of 85 genes, 58.82%	1401 of 4237 genes, 33.07%	1.87279e-05
	embryonic proventriculus	19 of 85 genes, 22.35%	336 of 4237 genes, 7.93%	0.000331347
	embryonic salivary gland	19 of 85 genes, 22.35%	234 of 4237 genes, 5.52%	4.05001e-06
	embryonic ventral epidermis	25 of 85 genes, 29.41%	629 of 4237 genes, 14.85%	0.00438576
	embryonic/larval fat body	21 of 85 genes, 24.71%	278 of 4237 genes, 6.56%	3.96323e-06
	amnioserosa	9 of 85 genes, 10.59%	131 of 4237 genes, 3.09%	0.0113089
	embryonic dorsal epidermis	31 of 85 genes, 36.47%	702 of 4237 genes, 16.57%	0.000107587
	embryonic foregut	27 of 85 genes, 31.76%	410 of 4237 genes, 9.68%	1.21157e-06
	embryonic hindgut	36 of 85 genes, 42.35%	808 of 4237 genes, 19.07%	1.5881e-05
	plasmatocytes	11 of 85 genes, 12.94%	93 of 4237 genes, 2.19%	3.6021e-05
	ubiquitous	24 of 85 genes, 28.24%	547 of 4237 genes, 12.91%	0.00142254
	yolk nuclei	15 of 85 genes, 17.65%	216 of 4237 genes, 5.10%	0.000287568
3	embryonic brain	28 of 57 genes, 49.12%	1092 of 4237 genes, 25.77%	0.00478187
	embryonic hindgut	21 of 57 genes, 36.84%	808 of 4237 genes, 19.07%	0.0366295
	embryonic/larval muscle system	18 of 57 genes, 31.58%	655 of 4237 genes, 15.46%	0.0403494
	ubiquitous	19 of 57 genes, 33.33%	547 of 4237 genes, 12.91%	0.00315244
	ventral nerve cord	30 of 57 genes, 52.63%	1102 of 4237 genes, 26.01%	0.00178349
4	embryonic salivary gland	3 of 3 genes, 100.00%	234 of 4237 genes, 5.52%	0.020469
5	embryonic brain	5 of 5 genes, 100.00%	1092 of 4237 genes, 25.77%	0.0484706
	embryonic central brain neuron	2 of 5 genes, 40.00%	38 of 4237 genes, 0.90%	0.0484706
	ventral nerve cord	5 of 5 genes, 100.00%	1102 of 4237 genes, 26.01%	0.0484706
6	dorsal prothoracic pharyngeal muscle	58 of 143 genes, 40.56%	442 of 4237 genes, 10.43%	3.65634e-20
	embryonic Malpighian tubule	30 of 143 genes, 20.98%	314 of 4237 genes, 7.41%	1.33598e-06
	embryonic anal pad	44 of 143 genes, 30.77%	322 of 4237 genes, 7.60%	1.28059e-15
	embryonic gastric caecum	15 of 143 genes, 10.49%	111 of 4237 genes, 2.62%	4.17235e-05
	embryonic hindgut	76 of 143 genes, 53.15%	808 of 4237 genes, 19.07%	6.17686e-19
	embryonic midgut	108 of 143 genes, 75.52%	1401 of 4237 genes, 33.07%	5.26809e-24
	embryonic/larval muscle system	66 of 143 genes, 46.15%	655 of 4237 genes, 15.46%	2.30025e-17
	embryonic/larval somatic muscle	40 of 143 genes, 27.97%	362 of 4237 genes, 8.54%	5.33029e-11
	embryonic/larval visceral muscle	41 of 143 genes, 28.67%	375 of 4237 genes, 8.85%	4.27454e-11
	ubiquitous	59 of 143 genes, 41.26%	547 of 4237 genes, 12.91%	1.36952e-16

Table 5.3. Enrichment analysis for the NP-MuScL network, using the spatial annotations made for the genes in the network in development stage 13-16.

Chapter 6

Conclusions

This thesis has explored the question of whether we can learn gene interaction networks from ISH images in a systematic and statistically sound manner. Our thesis presents a combination of (a) computer vision algorithms and pipelines to extract meaningful features, (b) design of algorithmic and machine learning tools to learn networks from ISH images, and finally (c) extensive empirical analysis of the results predicted by our algorithms to verify how well they perform with real data. This naturally closes the loop between designing algorithms on one hand, and empirical measurement and modeling on the other hand.

6.1 Summary of contributions

The techniques described in this thesis advance the state-of-the-art in the following ways:

- We proposed a computer vision pipeline SPEX^2 to extract features from Drosophila embryonic images. We found that using SPEX^2 features results in a significant improvement in both annotation accuracy, and clustering enrichment, over using state of the art computer vision features like SIFT, or previous feature extraction methods proposed for this problem.

- We proposed statistical machine learning based algorithms to learn gene interaction networks from ISH images collected at a single time point (GINI), as well as from multiple time points (NP-MuScL). GINI and NP-MuScL both extend Gaussian Graphical Model learning algorithms to work with kernel data and multiple data sources respectively. We verified that our algorithms work well on synthetic data, as well as the image data that is the focus of this thesis.
- Using extensive evaluations on a large data set of more than 100,000 images, we showed that our system can predict accurate gene interaction networks from Drosophila embryonic ISH data, using the notion of both spatial and temporal similarities of gene expression.

6.2 Future research

This thesis incorporates research from various fields including machine learning, computational biology, imaging and computer vision. We summarize ongoing and future work in each of these fields below.

6.2.1 Machine learning

NP-MuScL combines information from multiple data sources in a completely unsupervised manner to predict gene interactions. One future work problem would be to allow it some supervision in the form of examples of known gene interactions. Preliminary experiments show that given some known interactions, a GGM that does not penalize the known edges has improved precision and recall in predicting the unknown edges, over an unsupervised method that does not use the known edges in prediction. This suggests that a method may be devised to learn the importance of each data source, using the examples of known gene interactions as training data. Such a weighted NP-MuScL method will allow us to compare to methods like SVM that predict each edge independently of the rest, with a generative GGM-based model that predicts each edge in the context of all other edges.

One of the challenges in working with ISH data is the large amount of missing data - many genes have images for only 2-3 out of the 6 development stages. Preliminary work has been done on such estimation for GGMs (Kolar and Xing, 2012), but generalizing such results for GINI and NP-MuScL would be very useful in predicting a much larger gene interaction network.

Given n nodes, learning the structure of a Gaussian Graphical Model takes computational $O(n^3)$ time (for dense problems). While we can learn models over $10,000 - 20,000$ nodes, when the nodes are genes, for other problems, where the number of nodes is much larger, eg. $\sim 500,000$ SNPs, ~ 4 million Wikipedia articles, or ~ 1 billion users on a social network, we need to build more scalable algorithms, that can learn the structure of a Gaussian graphical model in a distributed fashion. This remains an open challenge.

6.2.2 Computational biology and bioimaging

This thesis implicitly assumes that the relationship between genes can be captured by a linear function; i.e. the expression of a single gene may be expressed as a linear weighted sum of the expression of its neighbors in the interaction network. However, the relationship between gene expressions may not be linear at all, but is a complex function that may also involve other data sources like the cis-regulatroy module(CRM) sequence of each gene, the type of interaction, etc. Janssens and Reinitz (2006); Segal et al. (2008) model the mechanistic forces that may control gene interaction using Arrhenius kinetics. However, such modeling is valid only when trying to capture interactions between transcription factors binding to the CRM of a gene. Generalizing the linear model to non-linear relationships that may capture more complex interactions between genes remains an open problem.

Recent advances in bio-imaging now allow us to develop robotic pipelines to collect 2D ISH data in a completely automated manner, on live embryos (Delubac, 2012). Further, the Berkeley Drosophila Transcription Network Project (Hendriks et al., 2006) has started imaging 3D images of the expression patterns for Drosophila embryos. For 2D automated images, SPEX^2, GINI and NP-MuScL can be used to predict gene interactions. In addition, since the imaging is done on live embryos, the amount of noise will be much smaller. However, for such data, we might wish to predict time-varying networks, as suggested in Ahmed and Xing (2009a). Extending such algorithms to work with vector-valued image data instead of classical scalar microarray data will need further work.

6.2.3 Computer vision

The SPEX^2 pipeline extracts high- fidelity spatial patterns of gene expression in a developing Drosophila embryo. Similar systems have also been proposed on a limited scale for mouse brain (Jagalur et al., 2007) and C. elegans (Peng et al., 2008). Further, different Drosophila data sets are now being collected with minor variations in color expression profile, and amount of noise in the data (Delubac, 2012; Lécuyer et al., 2007). How do we design the SPEX^2 pipeline to make it easily usable by the researchers who collect such data sets, to allow them to extract features from their own data set? An initial approach into this problem has been done by Shamir et al. (2010), but note that the image processing part of their pipeline is limited to only region-of-interest detection, which we have seen is not sufficient to get good results (Section 2.3.1). Open problems in this area are (a) how do you define expected shape of objects in the data set to be extracted, to make it easy to extract the correct objects, and register them to the correct shape, and (b) how do you define color profiles in the data set to easily extract the correct foreground and background from the data set.

6.3 Related peer-reviewed publications

The following is a list of publications that I have published during the course of my Ph.D.

1. Kriti Puniyani, Eric Xing. *GINI: From ISH images to gene interaction networks.* PLOS Computational Biology, 2013 (*in press*).
2. Kriti Puniyani, Eric Xing, *NP-MuScL: Unsupervised Global Prediction of Interaction Networks from Multiple Data Sources.* Journal of Computational Biology, 2013 (*in press*).
3. Eric P. Xing, Ross Curtis, Seunghak Lee, Junming Yin, Kriti Puniyani, Wei Wu, Peter Kinnaird, *GWAS in a Box: Statistical and Visual Analytics of Structured Associations via GenAMap* (*under review*).
4. Kriti Puniyani, Eric Xing, *NP-MuScL: Unsupervised Global Prediction of Interaction Networks from Multiple Data Sources.* Proceedings of the 17th annual international conference on RECOMB (Research in Computational Molecular Biology), LNCS Vol. 7821, pp 173-185, 2013.
5. Kriti Puniyani, Eric Xing, *Inferring gene interaction networks from ISH images via kernelized graphical models.* Proceedings of the 12th ECCV (European Conference on Computer Vision), LNCS Vol. 7577, pp 72-85, 2012.
6. Sivaraman Balakrishnan, Kriti Puniyani, John Lafferty, *Sparse additive functional and kernel CCA.* Proceedings of the 29th ICML (International Conference on Machine Learning) 2012, Edinburgh, Scotland.
7. Kriti Puniyani, Christos Faloutsos, Eric P. Xing, *SPEX^2: Automated Concise Extraction*

of Spatial Gene Expression Patterns from Fly Embryo ISH Images. ISMB (Intelligent Systems for Molecular Biology), Bioinformatics Vol. 26, Issue 12, pp i47-i56, 2010.

8. Kriti Puniyani, Seyoung Kim, Eric P. Xing, *Multi-Population GWA Mapping via Multi-Task Regularized Regression.* ISMB, Bioinformatics Vol. 26, Issue 12, pp i208-i216, 2010.

9. Kriti Puniyani, Jacob Eisenstein, Shay Cohen and Eric P. Xing, *Social Links from Latent Topics in Microblogs.* Social Media Workshop, NAACL 2010.

Bibliography

Ahmed, A. and Xing, E. P. (2009a). Tesla: Recovering time-varying networks of dependencies in social and biological studies. *Proc. Natl. Acad. Sci*, 106(29):11878–11883.

Ahmed, A. and Xing, E. P. (2009b). Tesla: Recovering time-varying networks of dependencies in social and biological studies. *Proc. Natl. Acad. Sci.*, 106:11878–11883.

Almstrup, K. and Leffers, H. (2004). Embryonic stem cell-like features of testicular carcinoma in situ revealed by genome-wide gene expression profiling. *Cancer Research*, 64(4736):4736–43.

Andrews, S., Tsochantaridis, I., and Hofmann, T. (2003). Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems*, pages 561–568. MIT Press.

Arava, Y., Wang, Y., Storey, J., Brown, P., and Herschlag, D. (2003). Genome-wide analysis of mrna translation profiles in saccharomyces cerevisiae. *Proceedings of the National Academy of Sciences*, 100:3889–3894.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404.

Bach, F. and Jordan, M. (2002). Learning graphical models with mercer kernels. In Becker, S., Thrun, S., and Obermayer, K., editors, *NIPS*, volume 15, pages 1009–1016.

Balakrishnan, S., Puniyani, K., and Lafferty, J. (2012). Sparse additive functional and kernel cca.

In *ICML*.

Banerjee, O., Ghaoui, L. E., d'Aspremont, A., and Natsoulis, G. (2006a). Convex optimization techniques for fitting sparse Gaussian graphical models. In *ICML*, pages 89–96.

Banerjee, O., Ghaoui, L. E., d'Aspremont, A., and Natsoulis, G. (2006b). Convex optimization techniques for fitting sparse gaussian graphical models. In *ICML*.

Bansal, M., Belcastro, V., Ambesi-Impiombato, A., and di Bernardo, D. (2007). How to infer gene networks from expression profiles. *Mol Sys Biology*, 3(78):doi:10.1038/msb4100158.

Bar-Joseph, Z., Gerber, G., Lee, T., Rinaldi, N., Yoo, J., Robert, F., Gordon, D., Fraenkel, E., Jaakkola, T., Young, R., and Gifford, D. (2003). Computational discovery of gene module and regulatory networks. *Nature Biotechnology*, 21(11).

Basso, K., Magolin, A., and Califano, A. (2005). Reverse engineering of regulatory networks in human B cells. *Nature Genetics*, 37:382–390.

BDGP (2005). Patterns of gene expression in Drosophila embryogenesis.

Ben-Hur, A. and Noble, W. S. (2005). Kernel methods for predicting protein–protein interactions. In *ISMB*, volume 21, pages i38–i46.

Boyle, E., Weng, S., and Sherlock, G. (2004). GO::TermFinder—open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. *Bioinformatics*, 20:3710–3715.

Carro, M. S., Califano, A., and Iavarone, A. (2010). The transcriptional network for mesenchymal transformation of brain tumours. *Nature*, 463:318–325.

Causier, B. (2004). Studying the interactome with the yeast two-hybrid system and mass spectrometry. *Mass Spectrom Rev*, 23(5):350–367.

Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*.

Chen, W.-Y., Song, Y., Bai, H., Lin, C.-J., and Chang, E. Y. (2008). *Parallel Spectral Clustering in Distributed Systems*.

Cho, R., Campbell, M., Winzeler, E., and Davis, R. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol Cell*, 2(1):65–73.

Consortium, T. F. (2002). The FlyBase database of the Drosophila genome projects and community literature. *Nucleic Acids Res*, 30:106–108.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., and Spiegelhalter, D. J. (1999). *Probabilistic Networks and Expert Systems*. Springer.

Danaher, P., Wang, P., and Witten, D. M. (2013). The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society, Series B.*, page In press.

Davidson, E. H. (2001). *Genomic Regulatory Systems*. Academic Press.

Delubac, D. (2012). *Automated Drosophila Embryo Injection, Imaging and IMAGE ANALYSIS TECHNOLOGIES FOR HIGH-THROUGHPUT SCREENS*. PhD Thesis.

Dempster, A. P. (1972). Covariance selection. *Biometrics-Special Multivariate Issue*, 28:157–175.

Dobra, A., Hans, C., Jones, B., Nevins, J., Yao, G., and West, M. (2004). Sparse graphical models for exploring gene expression data. *J. Mult. Analysis*, 90(1).

Dworak, H. and Sink, H. (2002). Myoblast fusion in drosophila. *BioEssays*.

Friedman, J., Hastie, T., and Tibshirani, R. (2007). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). glasso: Graphical lasso- estimation of Gaussian graphical models. http://cran.r-project.org/web/packages/glasso/index.html.

Frise, E., Hammonds, A., and Celniker, S. (2010). Systematic image-driven analysis of the spatial Drosophila embryonic expression landscape. *Mol Sys Biology*, 6(345):doi:10.1038/msb.2009.102.

Gargesha, M., Yang, J., Van Emden, B., Panchanathan, S., and Kumar, S. (2005). Automatic annotation techniques for gene expression images of the fruit fly embryo. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 5960, pages 576–583.

Gartner, T., Flach, P. A., Kowalczyk, A., and Smola, A. J. (2002). Multi-instance kernels. In *ICML*, pages 179–186.

Gilbert, S. F. (2003). *Developmental Biology, Seventh Edition*. Sinauer Associates.

Giot, L., Bader, J. S., Brouwer, C., et al. (2003). A protein interaction map of Drosophila melanogaster. *Science*, 302(5651):1727–1736.

Gorski, S. M., Chittaranjan, S., Pleasance, E. D., Freeman, J. D., Anderson, C. L., Varhol, R. J., Coughlin, S. M., Zuyderduyn, S. D., Jones, S. J. M., and Marra, M. A. (2003). A SAGE approach to discovery of genes involved in autophagic cell death. *Current Biology*, 13(4):358–363.

Goutte, C. and Gaussier, E. (2005). *Advances in Information Retrieval*, chapter A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. Springer Berlin / Heidelberg.

Hache, H., Lehrach, H., and Herwig, R. (2009). Reverse engineering of gene regulatory networks: A comparative study. *EURASIP Journal on Bioinformatics and Systems Biology*, 2009:doi:10.1155/2009/617281.

Harbison, C. T., Gordon, D. B., Lee, T. I., Rinaldi, N. J., Macisaac, K. D., Danford, T. W., Hannett, N. M., Tagne, J., Reynolds, D. B., Yoo, J., Jennings, E. G., Zeitlinger, J., Pokholok, D. K., Kellis, M., Rolfe, P. A., Takusagawa, K. T., Lander, E. S., Gifford, D. K., Fraenkel, E., and Young, R. A. (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104.

Heffel, A., Stadlerand, P. F., Prohaska, S. J., Kauer, G., and Kuska, J. P. (2008). Process flow for classification and clustering of fruit fly gene expression patterns. In *15th IEEE International Conference on Image Processing*, pages 721–724.

Hendriks, C. L., Keränen, S., Fowlkes, C., Simirenko, L., Weber, G. H., DePace, A. H., Henriquez, C., Hamann, D. W. K. B., Eisen, M. B., Malik, J., Sudar, D., Biggin, M., and Knowles, D. W. (2006). Three-dimensional morphology and gene expression in the drosophila blastoderm at cellular resolution i: data acquisition pipeline. *Genome Biology*, 7.

Hibbs, M., Hess, D., Myers, C., and Troyanskaya, O. (2007). Exploring the functional landscape of gene expression: directed search of large microarray compendia. *Bioinformatics*.

Honorio, J. and Samaras, D. (2011). Multi-task learning of gaussian graphical models. In *ICML*.

Hughes, T., Marton, M., Jones, A., Roberts, C., and Friend, S. (2000). Functional discovery via a compendium of expression profiles. *Cell*, 102(1).

Jaffrezic, F. and Tosser-Klopp, G. (2009). Gene network reconstruction from microarray data. *BMC Proceedings*, 3(Suppl 4):S12.

Jagalur, M., Pal, C., Learned-Miller, E., Zoeller, R., and Kulp, D. (2007). Analyzing in situ gene expression in the mouse brain with image registration, feature extraction and block clustering. *BMC Bioinformatics*, 8(10:S5).

Janssens, H. and Reinitz, J. (2006). Quantitative and predictive model of transcriptional control of the Drosophila melanogaster even skipped gene. *Nat Gen*, 38:1159–1165.

Ji, S., Yuan, L., Li, Y.-X., Zhou, Z.-H., Kumar, S., and J, Y. (2009). Drosophila gene expression pattern annotation using sparse features and term-term interactions. In *KDD*, pages 407–416.

JS, P., M, B., and K, M. (2003). Stage-specific regulation of caspase activity in Drosophila oogenesis. *Dev Biology*, 260(1):113–123.

Katenka, N. and Kolaczyk, E. D. (2012). Inference and characterization of multi-attribute networks with application to computational biology. *Arxiv*.

Kelley, B. P., Yuan, B., Lewitter, F., Sharan, R., Stockwell, B. R., and Ideker, T. (2004). PathBLAST: a tool for alignment of protein interaction networks. *Nucleic Acids Res*, 32(Web Server issue):83–88.

Koc, E. C., Burkhart, W., Blackburn, K., Moseley, A., and Spremulli, L. L. (2001). The small subunit of the mammalian mitochondrial ribosome: Identification of the full complement of ribosomal proteins present. *Journal of Biological Chemistry*, 276(22):19363–19374.

Kolar, M., Liu, H., and Xing, E. (2013). Markov network estimation from multi-attribute data. In *30th International Conference on Machine Learning*.

Kolar, M. and Xing, E. P. (2012). Estimating sparse precision matrices from data with missing values. In *ICML*.

Kumar, S., Jayaraman, K., Panchanathan, S., Gurunathan, R., Marti-Subirana, A., and Newfeld, S. (2002). BEST: a novel computational approach for comparing gene expression patterns from early stages of Drosophila melanogaster development. *Genetics*, 162(4):2037–47.

Lécuyer, E., Yoshida, H., Parthasarathy, N., Alm, C., Babak, T., Cerovina, T., Hughes, T., Tomancak, P., and Krause, H. (2007). Global analysis of mrna localization reveals a prominent role in organizing cellular architecture and function. *Cell*, 131(1):174–187.

Lein, E. S., Hawrylycz, M., Ao, N., and Jones, A. R. (2007). Genome-wide atlas of gene expression in the adult mouse brain. *Nature*, 445:168–176.

Li, Y.-X., Ji, S., Kumar, S., Ye, J., and Zhou, Z.-H. (2009). Drosophila gene expression pattern annotation through multi-instance multi-label learning. In *he Twenty-first International Joint Conference on Artificial Intelligence*.

Liben-Nowell, D. and Kleinberg, J. (2003). The link prediction problem for social networks. In *CIKM*.

Liu, H., Lafferty, J., and Wasserman, L. (2009). The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, 10:2295–2328.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Seventh International Conference on Computer Vision*, pages 1150–1157, Kerkyra , Greece.

Mace, D. L. and Ohler, U. (2010). Extraction and comparison of gene expression patterns from 2D RNA in situ hybridization images. *Bioinformatics*, 26(6):761–769.

Maron, O. and Ratan, A. L. (1998). Multiple-instance learning for natural scene classification. In *The Fifteenth International Conference on Machine Learning*, pages 341–349. Morgan Kaufmann.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*.

Missra, A. and Gilmour, D. S. (2010). Interactions between DSIF (DRB sensitivity inducing factor), NELF (negative elongation factor), and the Drosophila RNA polymerase II transcription elongation complex. *Proceedings of the National Academy of Sciences*, 107(25):11301–11306.

Montalta-He, H. and Reichert, H. (2003). Impressive expressions: developing a systematic database of gene-expression patterns in Drosophila embryogenesis. *Genome Biol*, 4(2):205.

Neumüller, R. A., Richter, C., Fischer, A., Novatchkova, M., Neumüller, K. G., and Knoblich, J. A. (2011). Genome-wide analysis of self-renewal in Drosophila neural stem cells by transgenic RNAi. *Cell Stem Cell*, 8(5):580–593.

NG, A., Jordan, M., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Neural Information Processing Systems*, pages 849–856. MIT Press.

Ong, I. M. (2002). Modelling regulatory pathways in E.coli from time series expression profiles. *Bioinformatics*, 18:241S–248S.

Pan, J.-Y., Balan, A. G. R., Xing, E. P., Traina, A. J. M., and Faloutsos, C. (2006). Automatic mining of fruit fly embryo images. In *KDD*.

Peng, H., Long, F., Liu, X., Kim, S., and Myers, E. W. (2008). Straightening C. elegans images. *Bioinformatics*, 24(2):234–242.

Peng, H. and Myers, E. (2004). Comparing in situ mrna expressions of Drosophila embryos.

In *Proc. 8th Annual Int. Conf. on Research in Computational Molecular Biology (RECOMB 2004)*, pages 157–166.

Pentland, A. and Turk, M. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*.

Persson, P. and Strang, G. (2004). A simple mesh generator in Matlab. *SIAM*, 46:329–345.

Puniyani, K., Faloutsos, C., and Xing, E. (2010). SPEX^2: automated concise extraction of spatial gene expression patterns from fly embryo ISH images. *Bioinformatics*, 26(12):47–56.

Puniyani, K. and Xing, E. P. (2012). Inferring gene interaction networks from ish images via kernelized graphical models. In *13th ECCV*.

Ravikumar, P., Liu, H., Lafferty, J., and Wasserman, L. (2007). Spam: Sparse additive models. In *NIPS*.

Ravikumar, P., Wainwright, M., Raskutti, G., and Yu, B. (2011). High-dimensional covariance estimation by minimizing L1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980.

Ren, X. and Malik, J. (2003). Learning a classification model for segmentation. In *the Ninth IEEE International Conference on Computer Vision*.

Rothman, A. J., Bickel, P. J., Levina, E., and Zhu, J. (2008). Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2.

Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., and Friedman, N. (2003). Module networks: Identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34(2):166–76.

Segal, E., Unnerstall, U., and Gaul, U. (2008). Predicting expression patterns from regulatory sequence in Drosophila segmentation. *Nature*, 451:535–540.

Shamir, L., Delaney, J. D., Orlov, N., Eckley, D. M., and Goldberg, I. G. (2010). Pattern recognition software and techniques for biological image analysis. *PloS Comp Bio*, 6(11).

Stark, C., Breitkreutz, B., Chatr-Aryamontri, A., Boucher, L., and Tyers, M. (2011). The biogrid interaction database: 2011 update. *Nucleic Acids Res.*, 39(D):698–704.

Storey, J. D. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society, Series B*, 64:479–498.

Stroschein-Stevenson, S. L., Foley, E., O'Farrell, P. H., and Johnson, A. D. (2006). Identification of Drosophila gene products required for phagocytosis of candida albicans. *PLoS Biology*, 4(1):e4.

Stuart, J. M., Segal, E., Koller, D., and Kim, S. K. (2003). A gene-coexpression network for global discovery of conserved genetic modules. *Science*, 302(5643):249–255.

Tanay, A., Sharan, R., Kupiec, M., and Shamir, R. (2004). Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc. Natl. Acad. Sci. USA*, 101(9):2981–6.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J R Statist. Soc B*, 58(1):267–288.

Tomancak, P., Beaton, A., Weiszmann, R., Kwan, E., Shu, S., Lewis, S., Richards, S., Ashburner, M., Hartenstein, V., Celniker, S., and Rubin, G. (2002a). Systematic determination of patterns of gene expression during Drosophila embryogenesis. *Genome Biol*, 3(2):14.

Tomancak, P., Beaton, A., Weiszmann, R., Kwan, E., Shu, S., Lewis, S. E., Richards, S., Ashburner, M., Hartenstein, V., Celniker, S. E., and Rubin, G. M. (2002b). Systematic determination of patterns of gene expression during Drosophila embryogenesis. *Genome Biology*, 3(2):research0088.1–research0088.14.

Tomancak, P., Berman, B. P., A, B., Weiszmann, R., Kwan, E., Hartenstein, V., Celniker, S. E., and Rubin, G. M. (2007). Global analysis of patterns of gene expression during drosophila embryogenesis. *Genome Biology*, 8(7):doi:10.1186/gb–2007–8–7–r145.

Uetz, P., Giot, L., Cagney, G., Mansfield, T., and et. al. (2000). A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, 403(6670):601–3.

Wang, K., Saito, M., and Califano, A. (2009). Genome-wide identification of post-translational modulators of transcription factor activity in human B-cells. *Nature Biotechnology*, 27(9):829–839.

Wang, Y., Joshi, T., Zhang, X.-S., Xu, D., and Chen, L. (2006). Inferring gene regulatory networks from multiple microarray datasets. *Bioinformatics*, 22(19):2413–2420.

Wellner, F. and Meyerowitz, E. M. (2004). Genome-wide analysis of spatial gene expression in Arabidopsis flowers. *Plant Cell*, 16:1314–1326.

Xu, Q., Hu, D. H., Yang, Q., and Xue, H. (2012). Simpletrppi: A simple method for transferring knowledge between interaction networks for ppi prediction. In *Bioinformatics and Biomedicine Workshops*.

Ye, J., Chen, J., Li, Q., and Kumar, S. (2006). Classification of drosophila embryonic developmental stage range based on gene expression pattern images. In *Computational Systems Bioinformatics conference*.

Yook, S.-H., Oltvai, Z. N., and Barabási, A.-L. (2004). Cell–cell interaction networks regulate blood stem and progenitor cell fate functional and topological characterization of protein interaction networks. *Proteomics*, 4(4):928–942.

Yuan, L., Woodard, A., Ji, S., Jiang, Y., Zhou, Z.-H., Kumar, S., and Ye, J. (2012). Learning sparse representations for fruit-fly gene expression pattern image annotation and retrieval. *BMC Bioinformatics*, 13(107):doi:10.1186/1471–2105–13–107.

Yuan, M. and Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35.

Yuji, K. (2001). Systematic analysis of gene expression of the C. elegans genome. *Protein, Nucleic Acid and Enzyme*, 46(16):2425–2431.

Zhou, J. and Peng, H. (2007). Automatic recognition and annotation of gene expression patterns of fly embryos. *Bioinformatics*, 23(5):589–596.