

Inferring Gene Interaction Networks from ISH Images via Kernelized Graphical Models

Kriti Puniyani and Eric P. Xing

School of Computer Science, Carnegie Mellon University
{kpuniyani, epxing}@cs.cmu.edu

Abstract. New bio-technologies are being developed that allow high-throughput imaging of gene expressions, where each image captures the spatial gene expression pattern of a single gene in the tissue of interest. This paper addresses the problem of automatically inferring a gene interaction network from such images. We propose a novel kernel-based graphical model learning algorithm, that is both convex and consistent. The algorithm uses multi-instance kernels to compute similarity between the expression patterns of different genes, and then minimizes the L_1 regularized Bregman divergence to estimate a sparse gene interaction network. We apply our algorithm on a large, publicly available data set of gene expression images of *Drosophila* embryos, where our algorithm makes novel and interesting predictions.

1 Introduction

As large-scale techniques for measuring gene expressions have been developed, automatically inferring gene regulatory or interaction networks from gene expression data has emerged as a popular technique to advance our understanding of cellular systems [1,2,3]. Predictions made by automatically learned gene networks have been experimentally validated [4,5], thus increasing the credibility of the algorithms. While previous approaches for estimating gene interaction networks focus on analyzing microarray data, the goal of this paper is to utilize the spatial similarity of gene expression using ISH image data to predict such networks.

In-situ hybridization (ISH) is a newly developed technique that can measure gene expression in tissue without homogenizing it, thus capturing the spatial variations in the gene expression. For multicellular organisms such as *Drosophila* (fruit fly) and human, it is well known that gene interactions are both time and space dependent. Since ISH images capture spatial variations in the gene expression, instead of capturing the average expression like in microarrays, analyzing ISH images should allow us to estimate spatial-dependent interaction networks.

In this paper, we focus our attention on data from the Berkeley *Drosophila* Genome Project (BDGP) [6]. BDGP is an ongoing effort to determine gene expression patterns during embryonic development for *Drosophila* genes. A light microscope is used in combination with a Spot RT digital camera to capture the images. Each image has a single central embryo, stained for a known gene.

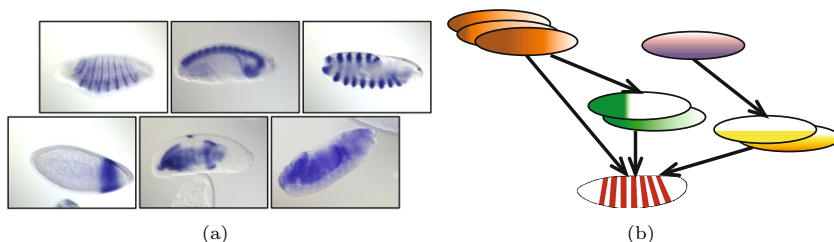


Fig. 1. (a) Examples of embryonic ISH images from the BDGP data. Presence of gene expression is indicated by the blue stain on the different regions of the embryo. (b) Our task is to infer an interaction network from bags of images per gene.

The age of the embryo has been labeled manually, classifying the embryo into one of six development stage ranges. Most images have a single embryo, however some images capture partial views of the embryo, others have overlapping or touching embryos. Some examples of the images found in the BDGP data can be viewed in Figure 1(a). The data currently contains over 100,000 ISH images capturing the expression pattern of 7200 genes.

1.1 Related Work

This paper proposes to learn an interaction network from images: each node of the network is a concept represented by a bag of images, and an edge indicates a relationship (or interaction) between the nodes. This problem is an instance of a class of problems where we wish to understand how concepts relate to each other. Similar problems already proposed in the vision literature include learning an ontology from images [7], finding related images given an image [8], etc.

Reverse engineering gene networks from microarray time course data is a well studied problem in the literature. Techniques have been proposed to reverse engineer gene networks based on Bayesian networks [9,10,1], undirected Gaussian graphical models [11,12], ordinary differential equations [13], partial correlations [2], and others. Comparisons of different methods used for reverse engineering gene networks have been performed [14,15]. However, this is the first work to reverse engineer a gene network from ISH image data.

1.2 Challenges

The underlying assumption behind our work is that if two genes have correlated expression patterns, then they may interact with each other. The main technical challenges in capturing spatial similarity to predict interaction networks (Figure 1(b)) are

Condition alignment: Images for different genes are typically taken under non-identical conditions (e.g., time, temperature, etc.), hence, signals from different images need to be 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.

Multi-variate measurement: An image records the expression of a gene at a particular time point in a d dimensional vector, where d are the number of features extracted from the image; given multiple multi-variate measurements of gene expression across time, we need to define a suitable distance metric between them.

Sparsity: The interaction network proposed must be sparse, since we expect that a small fraction of all possible interactions are actually present in a single organism.

1.3 Contributions

In this paper, we propose a system to predict gene interaction networks from spatial similarity of genes in ISH images from a single time point, leaving spatial-temporal networks for future work.

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, given bags of images per gene. The underlying graph encodes independence assumptions between the genes, thus an edge between a pair of genes suggests an interaction between them.

We propose a multi-instance kernel using different order statistics to compute similarity between bags of images. We then estimate the gene interaction network by minimizing the L_1 -penalized log-determinant Bregman divergence between the kernel and the estimated network. Under suitable conditions, we show that our graphical model learning algorithm is consistent, i.e. as the amount of available data increases, the algorithm is statistically guaranteed to predict the correct interactions between the genes. Further, our formulation is convex, hence the globally optimal estimator of the network is computed, no approximations are involved.

To the best of our knowledge, this work is the first work to predict interaction networks from images. Further, while [16] has 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. This paper is also the first work that uses Mercer kernels and Bregman divergence to predict kernelized graphical models using a convex formulation.

We apply our algorithm to the BDGP data, and find that the networks predicted are spatially coherent, modular, and have interesting new predictions of gene interactions.

2 Kernels for Image Similarity

Let g_i represent the i^{th} gene, $i \in \{1, \dots, n\}$, where n is the number of genes. For each gene g_i , the set of images measuring the expression of g_i are represented by bag \mathcal{B}_i , where each image b in \mathcal{B}_i is represented by d features, i.e. $b \in [0, 255]^d$. Each feature represents the gene expression in a spatial location of the embryo.

Any network-learning algorithm should leverage the existence of multiple images per gene per time point in improving its estimates of gene similarity, a problem which is reminiscent of multi-instance learning [17]. 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

$$\mathbf{K}(\mathcal{B}_i, \mathcal{B}_j) = k(s(\mathcal{B}_i), s(\mathcal{B}_j)) \quad (1)$$

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 $s(\mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} b$. Thus,

$$\begin{aligned} \mathbf{K}(\mathcal{B}_i, \mathcal{B}_j) &= Cov \left(\frac{1}{|\mathcal{B}_i|} \sum_{a \in \mathcal{B}_i} a, \frac{1}{|\mathcal{B}_j|} \sum_{b \in \mathcal{B}_j} b \right) \\ &= \frac{1}{|\mathcal{B}_i|} \frac{1}{|\mathcal{B}_j|} \sum_{a \in \mathcal{B}_i} \sum_{b \in \mathcal{B}_j} Cov(a, b) \end{aligned} \quad (2)$$

Thus, our choice of kernel is equivalent to comparing the 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 [18].

3 Kernel Based Graphical Models

The underlying joint distribution of a n -dimensional random vector (X_1, X_2, \dots, X_n) can be represented as an undirected graph $\mathcal{G} = (\mathcal{V}; \mathcal{E})$ with vertex set \mathcal{V} containing an element for each variable X_i , and the edges in \mathcal{E} represent relationships between the variables. Formally, an edge (i, j) is excluded from \mathcal{E} if and only if X_i is independent of X_j given all other variables.

Instead of learning \mathcal{E} by using i.i.d. samples of vector $X \in \mathbb{R}^n$, we will estimate it by directly using the kernel similarity matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$, defined in the previous section. Our goal is to estimate a graph structure which is close to \mathbf{K} , but is sparse. Thus, we wish to find a matrix Σ that minimizes the distance between the two matrices; defined as a Bregman divergence

$$D_{\mathcal{F}}(\mathbf{K}||\Sigma) = \mathcal{F}(\mathbf{K}) - \mathcal{F}(\Sigma) - \text{trace}(\nabla \mathcal{F}(\Sigma)^T(\mathbf{K} - \Sigma)) \quad (3)$$

where $\mathcal{F} : S \rightarrow \mathbb{R}$ is a differentiable, strictly convex function over the domain S of \mathbf{K} , Σ .

We consider the divergence that arises when we use $\mathcal{F}(\mathbf{X}) = -\log \det \mathbf{X}$, called the Log-Det divergence over positive definite matrices. Then,

$$D(\mathbf{K}||\Sigma) = \text{trace}(\mathbf{K}\Sigma^{-1}) - \log \det(\mathbf{K}\Sigma^{-1}) \quad (4)$$

The Log-Det divergence is of special interest, since it is scale invariant, as long as \mathbf{K} and Σ are scaled by the same linear transformation, a natural property that does not hold for the Frobenius norm distance. Further, the Log-Det divergence has been used as a measure of distance between covariance matrices, a property we will use to show the consistency of our algorithm.

In the high-dimensional scenario, when the number of features d is fewer than the number of genes n , the minimizer of $D(\mathbf{K}||\Sigma)$ is not well-defined, hence we need to add a regularizer. Since we expect the gene interaction network to have few edges, this suggests that Σ^{-1} should have few non-zeros, that is, Σ^{-1} should be sparse. To enforce sparsity, the L_0 norm of Σ^{-1} , which counts the number of non-zero elements, may be added as a regularizer to the Log-Det divergence objective. 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. Thus, our objective function is

$$\hat{\Sigma}^{-1} = \arg \min_{\Theta \succeq 0} \left\{ \text{trace}(\mathbf{K}\Theta) - \log \det \Theta + \lambda \|\Theta\|_1 \right\} \quad (5)$$

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 Θ , and hence, the graph induced by $\hat{\Sigma}^{-1}$ becomes more sparse. The edges in the graphical model are then estimated as

$$\mathcal{E} = \left\{ (i, j) \mid \hat{\Sigma}^{-1}(i, j) \neq 0; \quad i \neq j \right\} \quad (6)$$

Consistency of the estimate. Given i.i.d. data samples $X^{(1)}, X^{(2)}, \dots$, we can estimate the sparse inverse covariance matrix of the distribution of X by using the sample covariance matrix \mathbf{S} of the data as the kernel matrix in Equation 5. [19] shows that if the distribution of X is sub-Gaussian, then such an estimator is consistent, i.e. as the amount of data increases, the probability of estimating incorrect edges goes to zero.

Any kernel k can be represented as an inner product in some feature space ϕ , i.e. $k(a, b) = \phi(a)^T \phi(b)$. 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. Further, the choice of kernel we use reduces to computing the sample covariance matrix of features $s(\mathcal{B})$. Hence, for our choice of kernel, the algorithm is consistent.

Optimization. The objective function defined in Equation 5 is convex, hence it can be solved by any convex optimization algorithm. Banerjee et. al.[20] formulated an $O(n^4)$ block coordinate descent method to solve it, where n is the number of dimensions. Friedman et. al.[21] 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 [22]; and we use the *glasso* algorithm for efficient optimization of our objective function.

4 Feature Extraction and Representation

This section describe the feature extraction, representation, and normalization process used to obtain suitable features from images that can be input into our algorithm.

4.1 Gene Expression Extraction via SPEX²

The similarity of two ISH images should be dependent only on the spatial profile of the expression pattern within the embryo, and independent of the location, shape, orientation etc. of the embryo within the image. Hence, we need to extract a precise gene expression pattern from the ISH images. We use SPEX² [23], an automatic system for embryonic ISH image feature extraction for the same. SPEX² 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. 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.

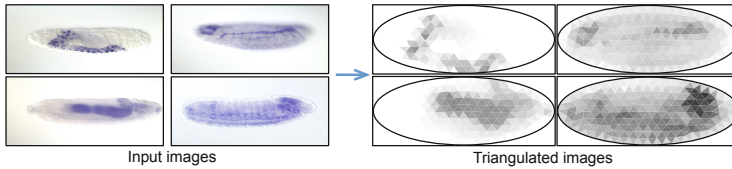


Fig. 2. Examples of how ISH images are converted into low-dimensional triangulated representations, for efficient feature representation

4.2 Delaunay Triangulation

The gene expression pattern extracted by SPEX² may be converted into features using standard methods like the SIFT feature descriptor[24], with PCA for dimensionality reduction. While this works well for supervised tasks like image annotation, it fails for unsupervised tasks, where the importance of every SIFT feature cannot be learned at training time. 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. Hence, we represent the expression pattern in a low dimensional space by triangulation, i.e. overlaying a fixed triangular mesh on top of the standardized embryo. The gene expression pattern for each image is then 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 [25], and aligning the mesh to the standardized embryo, as described in [26]. Each image is represented as a feature vector of length 311, with each feature corresponding to 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.

Figure 2 shows examples of ISH images converted into the triangulated gene expression patterns. As can be seen, triangulating the SPEX² output captures the key features of the gene expression location and strengths. Thus, triangulation enables dimensionality reduction of the feature space, while retaining explicit spatial information about the gene expression, which other dimensionality reduction techniques would not be able to capture.

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 : $\log(1 + t)$.

4.3 Feature and 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 (and their corresponding genes) must be removed from the 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 defined in Figure 3.

```

Input: triangulated features for  $n$  images :  $t_i$ , where  $i \in \{1, \dots, n\}$ ;
          variance threshold  $\epsilon$ 
Output: selected genes with their features in matrix  $\mathbf{X}$ 
for  $image\ i = 1 \dots n$  do
  |  $\mathbf{X}(i, \bullet) = t_i$ ;
end
while ( $\min(\text{var}(\mathbf{X})) < \epsilon \parallel \min(\text{var}(\mathbf{X}')) < \epsilon$ ) do
  |  $keepImages = find(\text{var}(\mathbf{X}') > \epsilon)$ ;
  |  $\mathbf{X} = \mathbf{X}(keepImages, \bullet)$ ;
  |  $keepFeatures = find(\text{var}(\mathbf{X}) > \epsilon)$ ;
  |  $\mathbf{X} = \mathbf{X}(\bullet, keepFeatures)$ ;
end
    
```

Fig. 3. Algorithm outlining feature and gene selection to reject uninformative images and features. $\text{var}(\mathbf{A})$ 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} .

5 Final Algorithm

A schematic of the system to estimate gene interaction networks from ISH images is shown in Figure 4, while the algorithm is summarized in Figure 5. The overall system has a computational complexity of $O(n^3 + n^2d + nd^2)$ (see supp. material for details). 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.

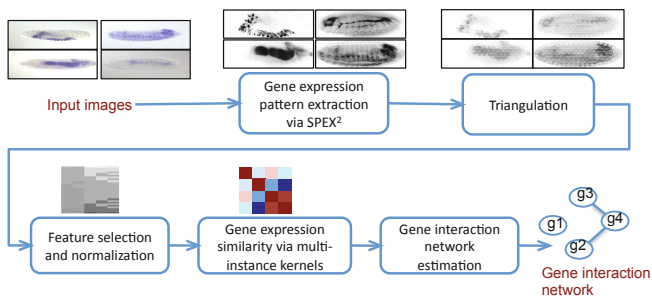


Fig. 4. The schematic shows an outline of the overall system to reverse engineer gene networks from ISH data. Sample output of each step is shown on top of the box corresponding to that step.


```

Input: Embryonic ISH images for  $n$  genes ;
 $\lambda$  - tuning parameter to control sparsity
Output: Predicted gene network for the  $n$  genes
for  $gene\ i = 1 \dots n$  do
  // feature extraction
   $\mathcal{B}_i = \{\}$ ;
  for each image  $j$  of gene  $g_i$  do
    Extract expression patterns from image  $j$  using SPEX2;
     $t_j = \text{triangulate expression pattern of image } j$ ;
    // feature normalization
     $t_j = \log(1 + 255 * \frac{t_j - \min(t_j)}{\max(t_j) - \min(t_j)})$  ;
     $b_j = \text{feature-selection}(t_j)$ ;
     $\mathcal{B}_i = \mathcal{B}_i \cup \{b_j\}$ ;
  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  $gene\ i = 1 \dots n$  do
  for  $gene\ j = 1 \dots n$  do
    |  $\mathbf{K}(i, j) = \text{Cov} \left( \frac{1}{|\mathcal{B}_i|} \sum_{a \in \mathcal{B}_i} a, \frac{1}{|\mathcal{B}_j|} \sum_{b \in \mathcal{B}_j} b \right)$ 
  end
end
// Run glasso using kernel  $\mathbf{K}$ 
 $\Sigma^{-1} = \text{glasso}(\mathbf{K}, \lambda)$  ;
Predicted edges in the network:  $\mathcal{E} = \text{non-zeros in the non-diagonal elements of } \Sigma^{-1}$  ;

```

Fig. 5. The final algorithm to obtain the gene network from ISH images

6 Results

6.1 Network Learned on Small Data Set

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 6(a), from 6 genes (each gene has 1-3 images in the data set). With $\lambda = 0.46$, 4 edges are predicted in the network, shown in Figure 6(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 our algorithm can be verified to be reasonable for the above small data set.

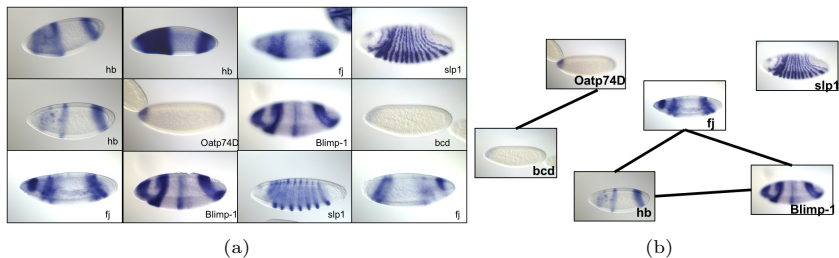


Fig. 6. (a) 12 input images and (b) network of genes learnt by it, with each gene represented by one image

6.2 BDGP Network

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, captured at key development stages of embryonic development. Each image captures embryonic gene expression of a single gene using 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. SPEX² 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. For each development stage, we ran a separate analysis.

Using a λ value of 0.775 for stage 9-10, we 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 our algorithm 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).

Some of the predicted gene interactions have already been reported in the literature. For example, in the network for stage 9-10, the algorithm predicts that CG5370 and CG12284 interact, both of which are known to be involved in apoptosis [27]. It also predicts that CG1064 interacts with 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, the algorithm predicts that the CG17158 interacts with the CG7254, and CG6608 interacts with CG2812, which has been previously reported in Rothberg et. al.[28].

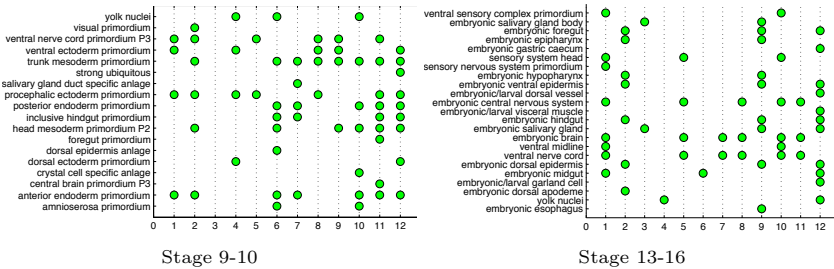


Fig. 7. Enrichment analysis for clusters in the gene interaction networks. A green dot indicates enrichment with a P-value < 0.05 .

6.3 Enrichment of Annotation Terms

Each gene in the BDGP data has been labeled manually by annotations describing the location of 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.

Enrichment analysis is a method to test if a given set of genes has a statistically significant difference in annotations from the background set. To test this, we cluster the gene network using spectral clustering [29] into 12 clusters, and analyze if each cluster has an over-representation of some annotation term, compared to the background population of all genes. We use 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 7(a)). 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 7(b).

6.4 The BDGP Networks Are Modular

We visualize 5 of the 12 clusters of each of the networks in Figure 8, and observe that the clusters in both networks are 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 reasonable prediction for real interaction networks.

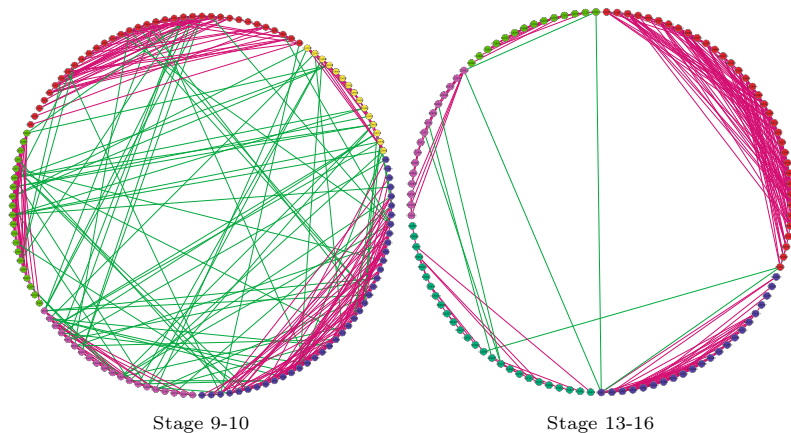


Fig. 8. A global view of the networks constructed by our algorithm for development stage 9-10, and 13-16, visualized for 5 of the 12 clusters found in section 6.3. The nodes of each cluster in the network are represented by different colors. Red edges are edges between nodes in the same cluster, while green edges are edges between nodes in different clusters.

6.5 Comparison with Microarray Network

We learn a network from microarray data from 12 time points in embryonic development [6], 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, as described in Section 6.3. Figure 9 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 enrichment of the hubs of the network: are the top 5% of the genes having the maximum connectivity with other genes functionally enriched? We find that the hubs of the ISH image networks are enriched for 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. However, the hubs of the microarray network are enriched for only a single function, where 4 of the 145 hub genes are involved in the “aromatic compound catabolic process”. The detailed enrichment analysis of the 2 networks can be found in the supplementary data.

Thus, we find that the network learned from ISH images is enriched for spatial annotation terms, as well as functional enrichment of the hubs of the network, which does not hold true for the microarray network. This suggests that analyzing ISH images could support better scientific conclusions than microarray data, which should be studied in greater detail.

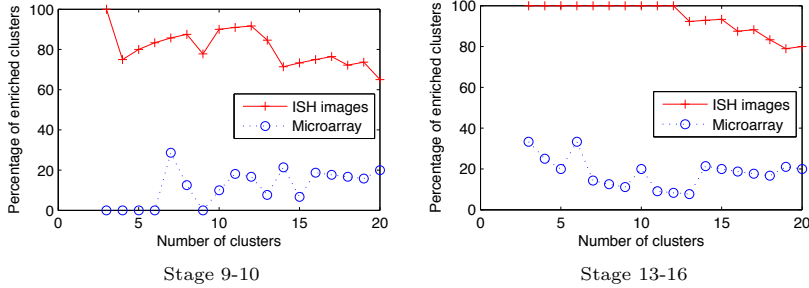


Fig. 9. The percentage of clusters that are enriched for spatial term annotations using networks learned from ISH and microarray data

7 Conclusions

We proposed an algorithm using kernelized graphical models to predict gene interaction networks by analyzing *Drosophila* embryo ISH images. While the experiments above have been reported on the ISH data from BDGP, our algorithm can be applied to all image data, by suitably modifying only the image processing SPEX² pipeline. 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 is much more significantly enriched for both spatial and functional annotations than 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.

References

1. Segal, E., Koller, D., Friedman, N.: Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics* 34, 166–176 (2003)
2. Basso, K., Magolin, A., Califano, A.: Reverse engineering of regulatory networks in human B cells. *Nature Genetics* 37, 382–390 (2005)
3. Morrissey, E.R., Juárez, M.A., Denby, K.J., Burroughs, N.J.: On reverse engineering of gene interaction networks using time course data with repeated measurements. *Bioinformatics* 26(18), 2305–2312 (2010)
4. Carro, M.S., Califano, A., Iavarone, A.: The transcriptional network for mesenchymal transformation of brain tumours. *Nature* 463, 318–325 (2010)
5. Wang, K., Saito, M., Califano, A.: Genome-wide identification of post-translational modulators of transcription factor activity in human B-cells. *Nature Biotechnology* 27(9), 829–839 (2009)

6. Tomancak, P., Beaton, A., Lewis, S.E., Richards, S., Celniker, S.E., Rubin, G.M.: Systematic determination of patterns of gene expression during drosophila embryogenesis. *Genome Biology* 3(2), 1–14 (2002)
7. Khan, L., Wang, L.: Automatic ontology derivation using clustering for image classification. In: 8th International Workshop on Multimedia Info Sys. (2002)
8. Jing, Y., Baluja, S.: Pagerank for product image search. In: WWW, pp. 307–316 (2008)
9. Friedman, N., Linial, M., Nachman, I., Pe'er, D.: Using bayesian networks to analyze expression data. *J. Comput. Biol.* 7(3-4), 601–620 (2000)
10. Pournara, I., Wernisch, L.: Reconstruction of gene networks using Bayesian learning and manipulation experiments. *Bioinformatics* 20(17), 2934–2942 (2004)
11. Ma, S., Gong, Q., Bohnert, H.J.: An arabidopsis gene network based on the graphical gaussian model. *Genome Res.* 17, 1614–1625 (2007)
12. Dobra, A., West, M.: Sparse graphical models for exploring gene expression data. *J. Multivariate Analysis* 90(1), 196–212 (2004)
13. Gardner, T., di Bernardo, D., Lorenz, D., Collins, J.J.: Inferring genetic networks and identifying compound mode of action via expression profiling. *Science* 301(5629), 102–105 (2003)
14. Bansal, M., Belcastro, V., Ambesi-Impiombato, A., di Bernardo, D.: How to infer gene networks from expression profiles. *Molecular Systems Biology* 3(78) (2007)
15. Hache, H., Lehrach, H., Herwig, R.: Reverse engineering of gene regulatory networks: A comparative study. *EURASIP Journal on Bioinformatics and Systems Biology* 2009 (2009)
16. Bach, F., Jordan, M.: Learning graphical models with mercer kernels. In: Becker, S., Thrun, S., Obermayer, K. (eds.) NIPS, vol. 15 (2002)
17. Maron, O., Ratan, A.L.: Multiple-instance learning for natural scene classification. In: ICML, pp. 341–349. Morgan Kaufmann (1998)
18. Gartner, T., Flach, P.A., Kowalczyk, A., Smola, A.J.: Multi-instance kernels. In: ICML, pp. 179–186 (2002)
19. Ravikumar, P., Wainwright, M., Raskutti, G., Yu, B.: High-dimensional covariance estimation by minimizing l_1 -penalized log-determinant divergence. *Electronic Journal of Statistics* (2011)
20. Banerjee, O., Ghaoui, L.E., d'Aspremont, A., Natsoulis, G.: Convex optimization techniques for fitting sparse gaussian graphical models. In: ICML, pp. 89–96 (2006)
21. Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* 9(3), 432–441 (2007)
22. Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B* 58(1), 267–288 (1996)
23. Puniyani, K., Faloutsos, C., Xing, E.: SPEX²: automated concise extraction of spatial gene expression patterns from fly embryo ISH images. *Bioinformatics* 26(12), 47–56 (2010)
24. Lowe, D.G.: Object recognition from local scale-invariant features. In: Seventh International Conference on Computer Vision, Kerkyra, Greece, pp. 1150–1157 (1999)
25. Persson, P., Strang, G.: A simple mesh generator in matlab. *SIAM*, 329–345 (2004)
26. Frise, E., Hammonds, A., Celniker, S.: Systematic image-driven analysis of the spatial drosophila embryonic expression landscape. *Mol. Sys. Biology* 6(345) (2010)
27. Peterson, J.S., Barkett, M., McCall, K.: Stage-specific regulation of caspase activity in drosophila oogenesis. *Dev. Biology* 260(1), 113–123 (2003)
28. Giot, L., Bader, J.S., Brouwer, C., et al.: A protein interaction map of *Drosophila melanogaster*. *Science* 302(5651), 1727–1736 (2003)
29. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Neural Information Processing Systems, pp. 849–856. MIT Press (2001)