Gene expression

Automated annotation of *Drosophila* gene expression patterns using a controlled vocabulary

Shuixiang Ji, Liang Sun, Rong Jin, Sudhir Kumar and Jieping Ye

1Department of Computer Science and Engineering, 2Center for Evolutionary Functional Genomics, The Biodesign Institute, Arizona State University, Tempe, AZ, 3Department of Computer Science and Engineering, Michigan State University, East Lansing, MI and 4School of Life Sciences, Arizona State University, Tempe, AZ, USA

Received on March 24, 2008; revised on July 2, 2008; accepted on July 6, 2008

Advance Access publication July 16, 2008

Associate Editor: David Rocke

ABSTRACT

Motivation: Regulation of gene expression in space and time directs its localization to a specific subset of cells during development. Systematic determination of the spatiotemporal dynamics of gene expression plays an important role in understanding the regulatory networks driving development. An atlas for the gene expression patterns of fruit fly *Drosophila melanogaster* has been created by whole-mount *in situ* hybridization, and it documents the dynamic changes of gene expression pattern during *Drosophila* embryogenesis. The spatial and temporal patterns of gene expression are integrated by anatomical terms from a controlled vocabulary linking together intermediate tissues developed from one another. Currently, the terms are assigned to patterns manually. However, the number of patterns generated by high-throughput *in situ* hybridization is rapidly increasing. It is, therefore, tempting to approach this problem by employing computational methods.

Results: In this article, we present a novel computational framework for annotating gene expression patterns using a controlled vocabulary. In the currently available high-throughput data, annotation terms are assigned to groups of patterns rather than to individual images. We propose to extract invariant features from images, and construct pyramid match kernels to measure the similarity between sets of patterns. To exploit the complementary information conveyed by different features and incorporate the correlation among patterns sharing common structures, we propose efficient convex formulations to integrate the kernels derived from various features. The proposed framework is evaluated by comparing its annotation with that of human curators, and promising performance in terms of F1 score has been reported.

Contact: jieping.ye@asu.edu

Supplementary information: Supplementary data are available at Bioinformatics online.

1 INTRODUCTION

Detailed knowledge of the expression and interaction of genes is crucial to deciphering the mechanisms underlying cell-fate specification and tissue differentiation. DNA microarrays and RNA *in situ* hybridization are two primary methods for monitoring gene expression levels on a large scale. Microarrays provide a quantitative overview of the relative changes of expression levels of a large number of genes, but they do not often document the spatial information on individual genes. In contrast, RNA *in situ* hybridization uses gene-specific probes and can determine the spatial patterns of gene expression precisely. Recent high-throughput investigations have yielded spatiotemporal information for thousands of genes in organisms such as *Drosophila* (Lécuyer et al., 2007; Tomancak et al., 2002) and mouse (Carson et al., 2005; Lein et al., 2006). These data have the potential to provide significant insights into the functions and interactions of genes (Kumar et al., 2002; Samsonova et al., 2007).

The fruit fly *Drosophila melanogaster* is one of the model organisms in developmental biology, and its patterns of gene expression have been studied extensively (Arbeitman et al., 2002; Campos-Ortega and Hartenstein, 1997; Lécuyer et al., 2007; Tomancak et al., 2002). The comprehensive atlas of spatial patterns of gene expression during *Drosophila* embryogenesis has been created by *in situ* hybridization techniques, and the patterns are documented in the form of digital images (Grumbling et al., 2006; Harmon et al., 2007; Tomancak et al., 2002; Van Emde et al., 2006). Comparative analysis of gene expression pattern images can potentially reveal new genetic interactions and yield insights into the complex regulatory networks governing embryonic development (Estrada et al., 2006; Kumar et al., 2002; Peng and Myers, 2004; Tomancak et al., 2002).

To facilitate pattern comparison and searching, the images of *Drosophila* gene expression patterns are annotated with anatomical and developmental ontology terms using a controlled vocabulary (Grumbling et al., 2006; Tomancak et al., 2002). The basic requirement for annotation is to assign a unique term, not only for each terminally differentiated embryonic structure, but also for the developmental intermediates that correspond to it. Four general classes of terms, called anlage *in statu nascendi*, anlage, primordium and organ (ordered in terms of developmental time), are used in the annotation. Such an elaborate naming scheme describes a developing ‘path’, starting from the cellular blastoderm stage until organs are formed, which documents the dynamic process of *Drosophila* embryogenesis. Due to the overwhelming complexity of this task, the images are currently annotated manually by human experts. However, the number of available images produced by high-throughput *in situ* hybridization is now...
rapidly increasing (Gurunathan et al., 2004; Kumar et al., 2002; Peng and Myers, 2004; Tomancak et al., 2007; Ye et al., 2006). It is, therefore, tempting to design computational methods for the automated annotation of gene expression patterns.

The automated annotation of Drosophila gene expression patterns was originally considered difficult due to the lack of a large reference dataset from which to learn. Moreover, the ‘variation in morphology and incomplete knowledge of the shape and position of various embryonic structures’ have made this task more elusive (Tomancak et al., 2002). We attempt to address this problem by resorting to advanced tools developed recently in the computer vision and machine learning research communities and on the large set of annotated data available from the Berkeley Drosophila Genome Project (BDGP) (Tomancak et al., 2002). There are several challenging questions that need to be addressed when approaching this problem by computational methods. As has been stated in Tomancak et al. (2002), the first challenge is to deal with the issue that the same embryonic structure can appear in different shapes and positions due to the distortions caused by the image acquisition process. Fortunately, recent advances in object recognition research have led to robust methods that can detect interest regions and extract features that are invariant to a class of local transformations from these regions. These two correlated lines of research have reached some maturity now [see Mikolajczyk et al. (2005) and Mikolajczyk and Schmid (2005) for an overview].

The second challenge of this task lies in the data representation. The embryogenesis of Drosophila has been divided into six discrete stage ranges (1–3, 4–6, 7–8, 9–10, 11–12 and 13–16) in the BDGP high-throughput study (Tomancak et al., 2002). Gene expression patterns are documented collectively by a group of images in a specific stage range. Similarly, annotation terms are also associated with a group of patterns sharing a subset of the named structures (Fig. 1). These attributes of the existing biological data pose challenges, because traditional machine learning tools require that each object in question be represented by a feature vector of fixed length. It is challenging to encode the variable number of images in a group into a fixed-length vector. The existing approach (Zhou and Peng, 2007) is based on the simplifying assumption that the terms are associated with individual images instead of image groups. Kernel methods developed in machine learning are a class of versatile tools for learning from unconventional data types, since they only require that the similarity between objects be abstracted into the so-called kernel matrix (Schölkopf and Smola, 2002). Kernels between various data types, e.g., strings, trees, graphs, and sets of vectors, have been proposed in the literature (Grauman and Darrell, 2005; Kondor and Jebara, 2003; Schölkopf et al., 2004).

We propose to extract a number of locally invariant features from each gene expression pattern image, and to compute kernels between sets of images based on the pyramid match algorithm (Grauman and Darrell, 2007a).

A recent comprehensive study shows that when local features are used to compute kernels between images, a combination of multiple feature types tends to yield better results than even the most discriminative individual feature type (Zhang et al., 2007). This motivates us to extract multiple feature types from each image and obtain multiple kernel matrices, one for each feature type. Thus, the final challenge for automated gene expression pattern annotation is to develop methods that can combine the multiple kernel matrices effectively. Automated methods for combining multiple kernel matrices, called multiple kernel learning (MKL), have been studied in machine learning recently. In such a framework, the optimal kernel matrix is obtained as a convex combination of a set of predefined candidate kernel matrices, and the coefficients for the combination can be computed by optimizing certain criteria. Methods for MKL have been proposed in the contexts of binary-class (Lanckriet et al., 2004a) and multi-class classification (Zien and Ong, 2007), and they have been applied successfully to various biological applications (De Bie et al., 2007; Lanckriet et al., 2004b).

For the problem of gene expression pattern annotation, a variable number of terms from the controlled vocabulary can be assigned to a group of patterns. Hence, this problem belongs to the more general framework of multi-label learning. We propose methods based on hypergraph (Agarwal et al., 2006; Zhou et al., 2007) to project and combine the multiple kernel matrices for multi-label data. The proposed formulation can capture the correlation among patterns sharing a common embryonic structure by including them in a common edge in hypergraph. We also show that kernel canonical correlation analysis (Hardoon et al., 2004) is a special case of the proposed formulation. The overall flowchart of the proposed framework is depicted in Figure 2.

We discuss feature generation and kernel construction in Section 2. The proposed formulation for multi-label multiple kernel learning is presented in Section 3. We report the results on gene expression pattern annotation in Section 4 and conclude this article with future work in Section 5.

2 FEATURE GENERATION AND KERNEL CONSTRUCTION

In this section, we present our methods for extracting features from gene expression pattern images and constructing kernels between sets of patterns.

2.1 Feature generation

There are two primary methods for extracting features. When the images are not well-aligned, the covariant region detector is first applied on the images to detect interest regions. Then, local descriptor is used to extract features from the detected regions. An alternative approach is to apply local descriptor on a dense regular grid, instead of interest regions (Grauman and Darrell, 2007b; Lazebnik et al., 2006). Such an approach is motivated from the bag-of-words model from the text-modeling literature, and competitive performance has been achieved on image applications (Fei-Fei and Perona, 2005). Since the images in our FlyExpress (Van Emden et al., 2006) database are already well-aligned, we take the second approach in this article (Fig. 2). Instead of tuning the local descriptor and grid size manually, we apply several popular local descriptors on regular grids of different sizes, and rely on the MKL

![Fig. 1. Sample image sets and the associated terms in the BDGP database in two stage ranges. Only images taken from lateral view with the anterior to the left are shown.](https://example.com/f1.png)
local descriptors. Results in Section 4 show that the integrated kernels yield better performance than that of the best individual kernel.

### 3.1 Hypergraph spectral learning

Hypergraph generalizes traditional graph by allowing edges, known as ‘hyperedges’, to connect more than two vertices, thus capturing the joint relationship among multiple vertices. We propose to construct a hypergraph (for the collection of gene expression patterns in question) in which each pattern is represented as a vertex. To document the joint similarity among patterns annotated with a common term, we propose to construct a hyperedge for each term in the vocabulary, and include all patterns annotated with a common term into one hyperedge. Hence, the number of hyperedges in this hypergraph equals the number of terms in the vocabulary.

Laplacian is commonly used to learn from a graph (Chang, 1997). To learn from a hypergraph, one can either define hypergraph Laplacian directly, or expand it into a traditional graph for which Laplacian is constructed. Since it has been shown that the Laplacians defined in both ways are similar (Agarwal et al., 2006), we use the expansion-based approaches in this article. The star and clique expansions are two commonly used schemes for expanding hypergraphs. Following the spectral graph embedding theory (Chung, 1997), we propose to project the patterns into a low-dimensional space in which patterns sharing a common term are close to each other. When formulated in the kernel-induced feature space, this can be achieved by solving the following optimization problem:

$$\max_B \text{trace}(B^T(KCK)B)$$

subject to $B^T(K^2 + \lambda K)B = I$, where $K \in \mathbb{R}^{n \times n}$ is the kernel matrix, $n$ is the number of image sets, $C = I - \mathcal{L}$ in which $\mathcal{L}$ is the normalized Laplacian matrix derived from the hypergraph, $B$ is the coefficient matrix for reconstructing the projection in feature space and $\lambda > 0$ is the regularization parameter.

Kernel canonical correlation analysis (kCCA) (Hardoon et al., 2004) is a widely used method for dimensionality reduction. It can be shown that kCCA involves the following optimization problem:

$$\max_B \text{trace}(B^T(K^2 + \beta K^2)B)$$

subject to $B^T(K^2 + \lambda K)B = I$,

where $X$ is the label matrix. Thus, kCCA is a special case of our proposed formulation based on hypergraph.

### 3.2 A convex formulation

It follows from the theory of reproducing kernels (Schölkopf and Smola, 2002) that the kernel $K$ in Equation (1) uniquely determines a mapping of the patterns to some feature space. Thus, kernel selection (learning) is one of the central issues in kernel methods. Following the multiple kernel learning framework (Lanckriet et al., 2004a), we propose to obtain an optimal kernel matrix by integrating multiple kernel matrices constructed from various features, that is, $K = \sum_{j=1}^{p} \theta_j K_j$ where $\{K_j\}_{j=1}^p$ are the $p$ kernels constructed from various local descriptors and $\{\theta_j\}_{j=1}^p$ are the weights satisfying $\sum_{j=1}^{p} \theta_j \text{trace}(K_j) = 1$. We show that the optimal weights that maximize the objective function in Equation (1) can be obtained by solving a semi-infinite linear program (SILP) (Hettich and Kortanek, 1993) in which a linear objective is optimized subject to an infinite number of linear constraints. This is summarized in the following theorem (Proof given in the Supplementary Material):

**Theorem 3.1.** Given a set of $p$ kernel matrices $K_1, \ldots, K_p$, the optimal kernel matrix, in the form of a linear combination of the given $p$ kernel matrices, is
matrices that maximizes the objective function in Equation (1), can be obtained by solving the following SILP problem:

\[
\max_{\theta, \gamma} \quad \gamma \quad \text{subject to} \quad \theta \geq 0, \theta^T r = 1,
\]

\[
\sum_{j=1}^{p} \theta_j S_j(\Psi) \geq \gamma, \quad \text{for all } \Psi \in \mathbb{R}^{n \times k},
\]

where \( S_j(\Psi), \) for \( j = 1, \ldots, p, \) is defined as

\[
S_j(\Psi) = \sum_{i=1}^{k} \left( \frac{1}{4} \psi_i^T \psi_j + \frac{1}{4k} \psi_i^T K_j \psi_j - r_j \psi_i^T h_i \right).
\]

\( \Psi = [\psi_1, \ldots, \psi_k], r = [r_1, \ldots, r_p]^T, \) and \( r_j = \text{trace}(K_j). \)

The SILP formulation proposed in Theorem 3.1 can be solved by the so-called ‘column generation’ technique, and more details can be found in the Supplementary Material.

4 RESULTS

In this section, we apply the proposed framework for annotating gene expression patterns. We use a collection of images obtained from the FlyExpress database (Van Emden et al., 2006), which contains standardized and aligned images. All the images used are taken from lateral view with the anterior to the left. The size of each raw image is 128 × 320.

4.1 Experimental setup

We apply nine local descriptors on regular grids of two different sizes on each image. The nine local descriptors are SIFT, shape context, PCA-SIFT, spin image, steerable filters, differential invariants, complex filters, moment invariants and cross-correlation. These local descriptors are commonly used for object recognition (more details can be found in Mikolajczyk and Schmid, 2005). The sizes of the grids we used are 16 and 32 pixels in radius and spacing (Fig. 2), and 133 and 27 local features are produced for each image, respectively.

It is known that local textures are important discriminative features of gene expression pattern images, and features constructed from filter banks and raw pixel intensities are effective in capturing such information (Varma and Zisserman, 2003). We, therefore, apply Gabor filters with different wavelet scales and filter orientations on each image to obtain global features of 384 and 2592 dimensions. We also sample the pixel values of each image using a bilinear technique, and obtain features of 10240, 2560 and 640 dimensions. The resulting features are called ‘global features’.

After generating the features, we apply the vocabulary-guided pyramid match algorithm (Grauman and Darrell, 2007a) to construct kernels between image sets. A total of 23 kernel matrices (2 grid sizes × 9 local descriptors + 2 Gabor + 3 pixel) are obtained. Then, the proposed MKL formulation is employed to obtain the optimal integrated kernel matrix. The performance of kernel matrices (either single or integrated) is evaluated by applying the support vector machine (SVM) for each term and treating image sets annotated with the selected terms as positive, and all other image sets as negative. We extract different numbers of terms from the FlyExpress database and use various numbers of image sets annotated with the selected terms for the experiments.

4.2 Annotation results

We apply the proposed formulations (star, clique and kCCA) to combine the various kernel matrices derived from different local descriptors. The performance of multiple kernel learning based on the soft margin 1-norm SVM (SVM1) criterion proposed in Lanckriet et al. (2004a) is also reported. Since the SVM1 formulation is only applicable to binary-class problems, we apply the formulation for each term by treating image sets annotated with this term as positive, and all other image sets as negative. To demonstrate the effectiveness of the proposed formulation for integrating kernels, we also report results obtained by combining the candidate kernels with uniform weight, along with the performance of the best individual kernel (among the 23 kernels) for each dataset. To compare with the existing method proposed in Zhou and Peng (2007), we extract wavelet features from images and apply the min-redundancy max-relevance feature selection algorithm to select a subset of features. As was done in Zhou and Peng (2007), we assign terms to individual images and apply linear discriminant analysis to annotate each image. Note that this setup does not consider the image group information and is the same as the one proposed in Zhou and Peng (2007). The annotation results measured by F1 score and precision and recall are summarized in Tables 1–4.

It can be observed from the results that in terms of both macro and micro F1 scores, the kernels integrated by either star or clique expansions achieve the highest performance on all but one of the datasets. This shows that the proposed formulation is effective in combining multiple kernels and potentially exploiting the complementary information contained in different kernels. For all datasets, the integrated kernels outperform the best individual kernel. In terms of precision and recall, our results indicate that SVM1 and Uniform achieve higher precision than the proposed formulations, while they both yield significantly lower recall. On the other hand, the best individual kernel produces slightly higher recall than the proposed formulations, while it yields significantly lower precision. Note that precision and recall are two competing criteria, and one can always...
We first select a number of terms and then extract certain number of image sets annotated with at least one of the selected terms. The number of terms used are 10, 20, 30, 40, 50 and 60, and the number of image sets used are 1000, 1500 and 2000 in each case. The first three rows report the F1 scores obtained by kernels combined with star expansion, clique expansion and CCA, respectively. The fourth row presents the F1 scores achieved by kernels combined with the soft margin 1-norm SVM (SV1M) formulation in which an optimal kernel is learned for each term separately. The fifth row shows the F1 scores achieved by kernels combined from the candidate kernels with uniform weights. The performance of the best individual kernel (BIK) over all local descriptors and grid sizes on the same dataset is reported in the sixth row. The results obtained by the method proposed in Zhou and Peng (2007) are reported in the last row. The performance shown in this table is the averaged scores over 10 random partitions of the entire dataset into training and test sets with ratio 1:1.

### Table 1. Performance of integrated kernels on gene expression pattern annotation in terms of macro F1 score

<table>
<thead>
<tr>
<th>No. of terms</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of sets</td>
<td>1000</td>
<td>1500</td>
<td>2000</td>
<td>1000</td>
<td>1500</td>
<td>2000</td>
</tr>
<tr>
<td>Star</td>
<td>0.5661</td>
<td>0.5741</td>
<td>0.5434</td>
<td>0.4396</td>
<td>0.4903</td>
<td>0.4575</td>
</tr>
<tr>
<td>Clique</td>
<td>0.5251</td>
<td>0.5220</td>
<td>0.4876</td>
<td>0.4536</td>
<td>0.5125</td>
<td>0.4926</td>
</tr>
<tr>
<td>kCCA</td>
<td>0.5487</td>
<td>0.5608</td>
<td>0.3532</td>
<td>0.3987</td>
<td>0.4635</td>
<td>0.4477</td>
</tr>
<tr>
<td>SVM1</td>
<td>0.4924</td>
<td>0.5413</td>
<td>0.3553</td>
<td>0.3780</td>
<td>0.4640</td>
<td>0.4356</td>
</tr>
<tr>
<td>Uniform</td>
<td>0.4947</td>
<td>0.5498</td>
<td>0.3517</td>
<td>0.3727</td>
<td>0.4703</td>
<td>0.4480</td>
</tr>
<tr>
<td>BIK</td>
<td>0.5418</td>
<td>0.5430</td>
<td>0.3518</td>
<td>0.4241</td>
<td>0.4515</td>
<td>0.4344</td>
</tr>
<tr>
<td>Z&amp;P</td>
<td>0.3756</td>
<td>0.3810</td>
<td>0.3775</td>
<td>0.2695</td>
<td>0.2759</td>
<td>0.2804</td>
</tr>
</tbody>
</table>

We first select a number of terms and then extract certain number of image sets annotated with at least one of the selected terms. The number of terms used are 10, 20, 30, 40, 50 and 60, and the number of image sets used are 1000, 1500 and 2000 in each case. The first three rows report the F1 scores obtained by kernels combined with star expansion, clique expansion and CCA, respectively. The fourth row presents the F1 scores achieved by kernels combined with the soft margin 1-norm SVM (SV1M) formulation in which an optimal kernel is learned for each term separately. The fifth row shows the F1 scores achieved by kernels combined from the candidate kernels with uniform weights. The performance of the best individual kernel (BIK) over all local descriptors and grid sizes on the same dataset is reported in the sixth row. The results obtained by the method proposed in Zhou and Peng (2007) are reported in the last row. The performance shown in this table is the averaged scores over 10 random partitions of the entire dataset into training and test sets with ratio 1:1.

### Table 2. Performance of integrated kernels on gene expression pattern annotation in terms of micro F1 score

<table>
<thead>
<tr>
<th>No. of terms</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of sets</td>
<td>1000</td>
<td>1500</td>
<td>2000</td>
<td>1000</td>
<td>1500</td>
<td>2000</td>
</tr>
<tr>
<td>Star</td>
<td>0.5841</td>
<td>0.6011</td>
<td>0.5728</td>
<td>0.4861</td>
<td>0.5199</td>
<td>0.4847</td>
</tr>
<tr>
<td>Clique</td>
<td>0.5424</td>
<td>0.5429</td>
<td>0.5079</td>
<td>0.5039</td>
<td>0.5422</td>
<td>0.5257</td>
</tr>
<tr>
<td>kCCA</td>
<td>0.5727</td>
<td>0.5922</td>
<td>0.5643</td>
<td>0.4581</td>
<td>0.4994</td>
<td>0.4887</td>
</tr>
<tr>
<td>SVM1</td>
<td>0.5290</td>
<td>0.5781</td>
<td>0.5786</td>
<td>0.4361</td>
<td>0.5024</td>
<td>0.4844</td>
</tr>
<tr>
<td>Uniform</td>
<td>0.5341</td>
<td>0.5870</td>
<td>0.5837</td>
<td>0.4390</td>
<td>0.5096</td>
<td>0.4975</td>
</tr>
<tr>
<td>BIK</td>
<td>0.5585</td>
<td>0.5650</td>
<td>0.5677</td>
<td>0.4614</td>
<td>0.4735</td>
<td>0.4562</td>
</tr>
<tr>
<td>Z&amp;P</td>
<td>0.4031</td>
<td>0.4032</td>
<td>0.3796</td>
<td>0.3034</td>
<td>0.2985</td>
<td>0.2827</td>
</tr>
</tbody>
</table>

This table shows the performance of each method in terms of micro F1 score. See the footnotes of Table 1 for detailed explanations.

### Table 3. Performance of integrated kernels on gene expression pattern annotation in terms of precision

<table>
<thead>
<tr>
<th>No. of terms</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of sets</td>
<td>1000</td>
<td>1500</td>
<td>2000</td>
<td>1000</td>
<td>1500</td>
<td>2000</td>
</tr>
<tr>
<td>Star</td>
<td>0.5246</td>
<td>0.5141</td>
<td>0.4861</td>
<td>0.4629</td>
<td>0.5349</td>
<td>0.4842</td>
</tr>
<tr>
<td>Clique</td>
<td>0.4586</td>
<td>0.4375</td>
<td>0.3968</td>
<td>0.4531</td>
<td>0.5224</td>
<td>0.5053</td>
</tr>
<tr>
<td>kCCA</td>
<td>0.5448</td>
<td>0.5443</td>
<td>0.5230</td>
<td>0.4917</td>
<td>0.5377</td>
<td>0.5585</td>
</tr>
<tr>
<td>SVM1</td>
<td>0.5973</td>
<td>0.6163</td>
<td>0.5985</td>
<td>0.5387</td>
<td>0.6121</td>
<td>0.6211</td>
</tr>
<tr>
<td>Uniform</td>
<td>0.6258</td>
<td>0.6462</td>
<td>0.6155</td>
<td>0.5691</td>
<td>0.6417</td>
<td>0.6495</td>
</tr>
<tr>
<td>BIK</td>
<td>0.4956</td>
<td>0.4830</td>
<td>0.4687</td>
<td>0.4247</td>
<td>0.4994</td>
<td>0.4814</td>
</tr>
<tr>
<td>Z&amp;P</td>
<td>0.3298</td>
<td>0.3244</td>
<td>0.3182</td>
<td>0.2311</td>
<td>0.2455</td>
<td>0.2453</td>
</tr>
</tbody>
</table>

This table shows the performance of each method in terms of precision. See the footnotes of Table 1 for detailed explanations.
### Table 4. Performance of integrated kernels on gene expression pattern annotation in terms of recall

<table>
<thead>
<tr>
<th>No. of terms</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of sets</td>
<td>1000</td>
<td>1500</td>
<td>2000</td>
<td>1000</td>
<td>1500</td>
<td>2000</td>
</tr>
<tr>
<td>Star</td>
<td>0.6482</td>
<td>0.6892</td>
<td>0.5019</td>
<td>0.5303</td>
<td>0.5117</td>
<td>0.5033</td>
</tr>
<tr>
<td>Clique</td>
<td>0.6331</td>
<td>0.6654</td>
<td>0.5238</td>
<td>0.5649</td>
<td>0.4824</td>
<td>0.4820</td>
</tr>
<tr>
<td>kCCA</td>
<td>0.5952</td>
<td>0.6285</td>
<td>0.5966</td>
<td>0.4111</td>
<td>0.4483</td>
<td>0.4259</td>
</tr>
<tr>
<td>SVM1</td>
<td>0.4890</td>
<td>0.5383</td>
<td>0.5211</td>
<td>0.3494</td>
<td>0.4237</td>
<td>0.3890</td>
</tr>
<tr>
<td>Uniform</td>
<td>0.4830</td>
<td>0.5378</td>
<td>0.5198</td>
<td>0.3403</td>
<td>0.4219</td>
<td>0.3948</td>
</tr>
<tr>
<td>BIK</td>
<td>0.4990</td>
<td>0.5504</td>
<td>0.5881</td>
<td>0.4242</td>
<td>0.3990</td>
<td>0.4460</td>
</tr>
<tr>
<td>Z&amp;P</td>
<td>0.5952</td>
<td>0.6285</td>
<td>0.5966</td>
<td>0.4111</td>
<td>0.4483</td>
<td>0.4259</td>
</tr>
</tbody>
</table>

This table shows the performance of each method in terms of recall. See the footnotes of Table 1 for detailed explanations.

### Fig. 3. Annotation results for sample patterns in the six stage ranges. BDGP terms denote terms that are assigned by human curators in the Berkeley Drosophila Genome Project (Tomancak et al., 2002), and predicted terms denote terms predicted by the proposed computational framework. These patterns are randomly sampled from each stage range, and hence they may not correspond to the same gene.

The pyramid match algorithm can compute kernels between variable-sized sets of images. Thus, terms can be predicted for image sets of any size. Overall, the proposed computational framework achieves promising performance on annotating gene expression patterns. Meanwhile, we realize that the current framework suffers from some potential limitations. By comparing the BDGP terms and the predicted terms for patterns in stage ranges 7–8 and 9–10, we can see that the structures related to endoderm are predicted correctly while some of those related to mesoderm are prone to error. This may be due to the fact that, when viewed laterally, structures related to mesoderm are more prone to be hidden than those related to endoderm. This phenomenon becomes clearer when we examine the results for stage range 13–16 in Figures 3 and 4. As shown in Figure 4, there are a total of five images in this set in the original BDGP database. Among these five images, only two of them (the first and third) are taken from the lateral view and hence are used in our experiments. The second and the fourth images are taken from the ventral view, and the fifth image is taken from the dorsal view. The structure **ventral midline** can only be documented by digital images taken from the ventral view as can be seen from the second
Fig. 4. The original five images in stage range 13–16 from BDGP. The first and the third images are taken from lateral view; the second and the fourth images are taken from ventral view; the fifth image is taken from dorsal view. Only the first and the third images are used in our experiments shown in the bottom of Figure 3.

and the fourth images in Figure 4. Since we only use images from the lateral view, it can be seen from Figure 3 that the proposed framework cannot predict this term correctly. This problem can potentially be solved by using images taken from other views such as ventral and dorsal. However, incorporation of images with multiple views may complicate the computational procedure, so requires a special care.

To evaluate the scalability of the proposed formulations, we vary the number of terms and the number of image sets, and compare the change of computation time. On a machine with Pentium 4 3.40 GHz CPU and 1 GB of RAM, when the number of terms is increased from 20 to 60 on a dataset of 500 image sets, the computation time increases from approximately 4s to 11s. In terms of the number of image sets, datasets of 1500 and 2000 image sets with 60 terms take around 3 and 4 min, respectively.

5 CONCLUSIONS AND DISCUSSIONS

In this article, we have presented a computational framework for annotating gene expression patterns of Drosophila. We propose to extract invariant features from gene expression pattern images and construct kernels between these sets of features. To integrate multiple kernels effectively, we propose multi-label, multiple kernel learning formulations based on hypergraph. Experimental evaluation shows that the integrated kernels consistently outperform the best individual kernel. Currently, the annotation of patterns by human curators requires multiple passes, and the proposed framework can be used as a preprocessing step whose annotation is further refined by human curators.

In future work, we plan to perform a detailed analysis of the weights obtained by the MKL formulation, and investigate how they are related to the relevance of each kernel. Our experimental results show that features extracted on smaller grids tend to yield better results. However, computational resource limitations prevent the use of a grid size smaller than 16 pixels. We plan to explore ways to overcome this problem. Retrieving gene expression patterns by combining information from images and annotations is an interesting and challenging research issue. The proposed framework can assign a probability of associating each term to each image, producing a probability vector for unannotated images from various high-throughput experiments. Such information can potentially be exploited to facilitate pattern retrieval. Detailed analysis of the annotation results produced by the proposed framework indicates that integration of gene expression pattern images taken from multiple views can potentially improve the annotation performance.

In this case, the current pyramid match algorithms need to be adapted so that only images taken from the same view are matched. It can be seen from the third and fifth images in Figure 4 that the annotation terms can also be associated with partial patterns in each image. These partial patterns have been removed in our FlyExpress database (Fig. 3), so these terms cannot be predicted correctly by the proposed framework. We plan to explore ways to incorporate these partial patterns in the future.

ACKNOWLEDGEMENTS

We thank Kristen Grauman, John Lee and Zhi-Hua Zhou for fruitful discussions, Hanchuan Peng for the feature selection code, Kristi Garboushian for editorial support, and Bernard Van Emden for help with access to the gene expression data.

Funding: This work is supported in part by research grants from National Institutes of Health under No. HG002516 and National Science Foundation under No. IIS-0612069.

Conflict of Interest: none declared.

REFERENCES


