Drosophila Gene Expression Pattern Annotation through Multi-Instance Multi-Label Learning

Ying-Xin Li, Shuiwang Ji, Sudhir Kumar, Jieping Ye, and Zhi-Hua Zhou

Abstract—In the studies of *Drosophila* embryogenesis, a large number of two-dimensional digital images of gene expression patterns have been produced to build an atlas of spatio-temporal gene expression dynamics across developmental time. Gene expressions captured in these images have been manually annotated with anatomical and developmental ontology terms using a controlled vocabulary (CV), which are useful in research aimed at understanding gene functions, interactions, and networks. With the rapid accumulation of images, the process of manual annotation has become increasingly cumbersome, and computational methods to automate this task are urgently needed. However, the automated annotation of embryo images is challenging. This is because the annotation terms spatially correspond to local expression patterns of images, yet they are assigned collectively to groups of images and it is unknown which term corresponds to which region of which image in the group. In this paper, we address this problem using a new machine learning framework, Multi-Instance Multi-Label (MIML) learning. We first show that the underlying nature of the annotation task is a typical MIML learning problem. Then, we propose two support vector machine algorithms under the MIML framework for the task. Experimental results on the FlyExpress database (a digital library of standardized *Drosophila* gene expression pattern images) reveal that the exploitation of MIML framework leads to significant performance improvement over state-of-the-art approaches.

Index Terms—Gene expression pattern, image annotation, machine learning, multi-instance multi-label (MIML) learning, support vector machine, *Drosophila*.

1 Introduction

Embryonic development is orchestrated by the spatio-temporal expression of a multitude of genes over time. Understanding of this expression dynamics paves the way toward unraveling the patterns and processes governing development. The fruit fly *Drosophila melanogaster* is a well-studied model organism for this purpose. To accelerate genome-wide studies of *Drosophila* embryogenesis, the Berkeley *Drosophila* Genome Project (BDGP) [41], [42] has produced comprehensive atlas of gene expression patterns during *Drosophila* embryonic development in the form of two-dimensional (2D) digital images by high-throughput RNA *in situ* hybridization to whole-mount embryos. Each *in situ* image records the spatial distribution of gene expression within an embryo at a particular *Drosophila* time (developmental stage range). These spatio-temporal images are stored in a database and organized in groups based on genes and developmental stages. Text-based anatomical and developmental ontology terms using a controlled vocabulary (CV) [42] are assigned to each image group, as illustrated in Fig. 1. These annotations indicate the gene expression and can be used for identifying genes with similar patterns and for connecting embryonic gene expression with their adult counterparts [17], [21], [27]. The annotation task is generally carried out manually by human curators. With the rapid accumulation of the *in situ* images, manual annotation becomes more and more intractable. Therefore, it is highly desired to develop computational methods to automate the annotation task [23], [52].

The task of automating expression annotation poses several significant challenges. First of all, in the BDGP database, the annotation terms are assigned collectively to groups of images, and some terms do not apply to all the images in the group. As illustrated in Fig. 1, each image group is assigned with multiple CV terms, but this does not imply that each image in the group is associated with all the terms. Moreover, the terms are generally relevant to regions rather than whole images, while in the BDGP database the correspondence between the regions and the terms is unknown. The problem is even more complicated by considering that even for the same term, the corresponding regions in different images may have significant variations in visual appearances. In addition, the variability and complexity in the morphology of embryo anatomical structures and the effects of overlapping structure in a 3D embryo on signal detection also cause difficulties for the annotation problem.

Several prior attempts for the automated annotation of fruit fly embryo *in situ* images have been reported. Zhou and Peng [52] conducted their study based on a simplified assumption that each image in the data set is annotated with CV terms. They represented each image by wavelet

- *Y.-X. Li and Z.-H. Zhou are with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China. E-mail: {liyx, zhouzh}@lamda.nju.edu.cn.*
- *S. Ji is with the Department of Computer Science, Old Dominion University, 4700 Elkhorn Avenue, Suite 3300, Norfolk, VA 23529-0162. E-mail: sji@cs.odu.edu.*
- *S. Kumar and J. Ye are with the Center for Evolutionary Medicine & Informatics, Biodesign Institute, Arizona State University, Tempe, AZ 85287. E-mail: {s.kumar, jieping.ye}@asu.edu.*

Manuscript received 1 Mar. 2010; revised 2 Aug. 2010; accepted 12 Dec. 2010; published online 15 Apr. 2011.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-2010-03-0063. Digital Object Identifier no. 10.1109/TCBB.2011.73.

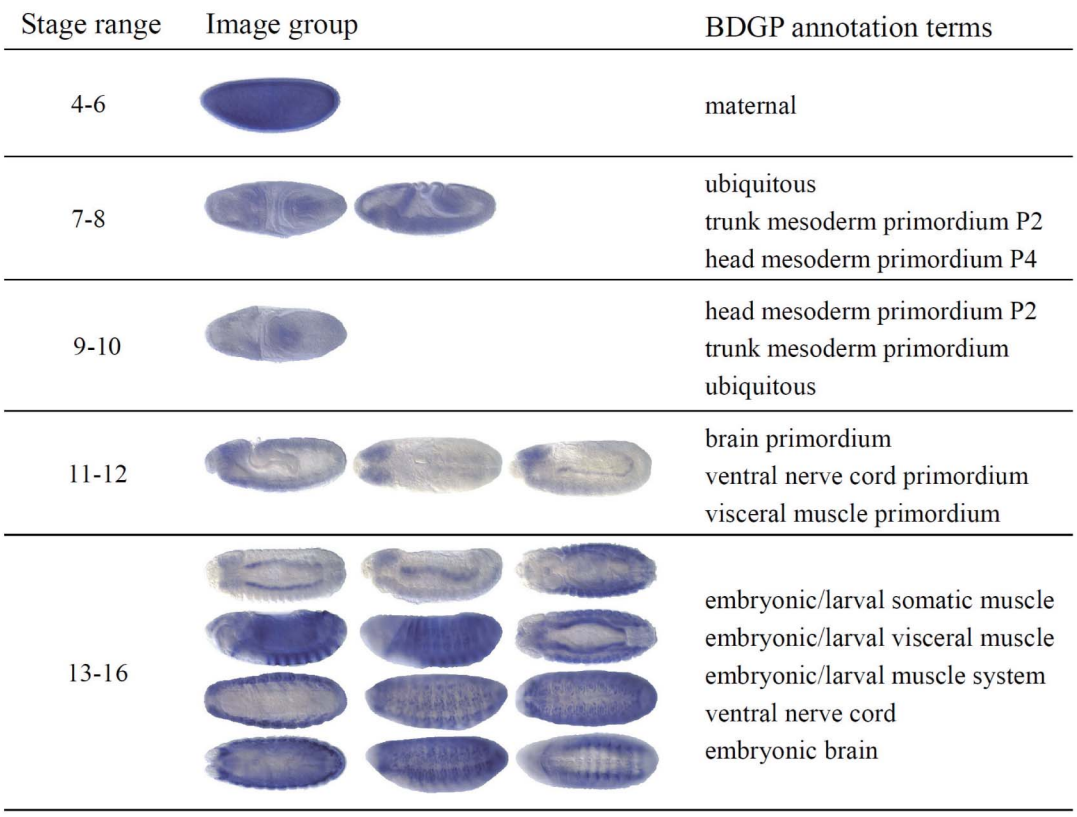


Fig. 1. Example image groups and associated annotation terms for the gene *Actn* in different *Drosophila* developmental stages (4-6, 7-8, 9-10, 11-12, and 13-16) in the BDGP database. The darkly stained region highlights the place where the gene is expressed. There are a total of 5 image groups presented above with various numbers of *in situ* images contained in each group and annotated with different CV terms.

embryo features and designed a two-tier automatic annotation system. The setting in their study, however, does not directly apply to the BDGP data due to their simplified assumption. Ji et al. [23] used pyramid match kernels [15], [28] to calculate the similarity between sets of images, and proposed a multiple-kernel-learning formulation based on a hypergraph to construct a predictive model. To achieve better annotation performance, a bag-of-words scheme was employed later [22] to represent the expression patterns of image groups, and a linear classification model based on shared-subspace learning framework was constructed. Recently, they built a local regularization based classification model with image groups represented by an improved bag-of-words scheme to perform computational annotation [24], and achieved the best annotation performance on the BDGP database to date.

In this paper, we first show that the learning task underlying the *Drosophila* gene expression pattern annotation problem matches well with a new machine learning framework, Multi-Instance Multi-Label (MIML) learning [56], [57]. Then, we propose two support vector machine algorithms under the MIML framework to tackle the annotation task. Experiments show that the exploitation of MIML framework can lead to performance superior to existing *Drosophila* gene expression pattern annotation approaches.

The rest of this paper is organized as follows: In Section 2, we focus on the formalization of the annotation problem and briefly introduce the MIML learning framework. In Section 3, we present the MIMLSVM^{+} method and extend it further by incorporating a strategy to take into account the correlations between annotations. Experimental results are reported in Section 4, followed by discussions in Section 5. Finally, Section 6 concludes the paper.

2 Data Set and Annotation Task Formalization

2.1 Data Set

To develop and test our automated annotation methods, we use the FlyExpress database (http://www.flyexpress.net) which is widely used to develop and test computational annotation approaches for *Drosophila* gene expression pattern images [22], [24], [57] instead of the raw BDGP database. The FlyExpress database collects the images of *Drosophila* embryos produced by whole-mount *in situ* hybridization technique from the BDGP database and standardized them semi-manually. That is, each image contains only one individual whole embryo; all the embryos are scaled to the same size of 320×128 pixels, and aligned with the embryo anterior to the left. Though this reduces the number of *in situ* images, it makes the annotation task easier in some sense. The organization of the images is the same as the BDGP database, as shown in Fig. 1. Currently, the FlyExpress database contains more than 9,000 image groups consisting of over 40,000 BDGP *in situ* images.

2.2 The Formalization of the *Drosophila* Annotation Task

For each image group, text-based terms from a preconstructed anatomical ontology are assigned to indicate gene

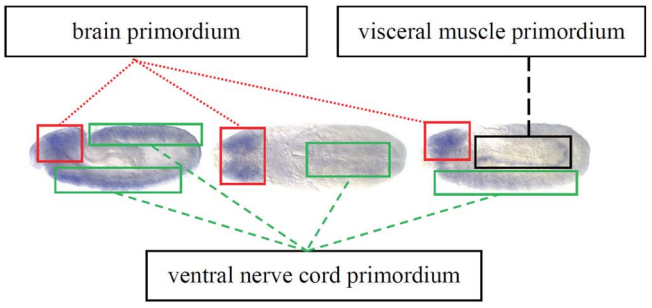


Fig. 2. Illustration of the underlying relationships between the CV terms and their corresponding image patterns. The image group of gene *Actn* in the stage range 11-12 in Fig. 1 is presented here. A set of terms is collectively assigned to a group of images, while we do not know which term is related to which region in which image.

expression (staining). As mentioned in Section 1, CV terms are collectively assigned to image groups, while the relations between terms and images are quite complicated. For example, in Fig. 2, the term "brain primordium" is related to the head position of all the three images, and if the head position of one image shows such a pattern, then the term will be annotated to the image group; the term "ventral nerve cord primordium" can be observed in all three images, but at different positions; while the term "visceral muscle primordium" is only visible in the third image.

A classical assumption in pattern recognition or image annotation is "similar patterns share similar annotations/labels." If we take this assumption directly, a poor performance might be obtained for the current task, because the "image similarity" is not directly related to "term similarity." For example, suppose an image A_i is similar to the first image in Fig. 2 in most areas, yet different in a small region at the head position; though on the whole the two images are similar, A_i does not have the term "brain primordium." For another example, suppose an image A_j is very different from the first image in Fig. 2 in most areas, yet similar at the head position; though on the whole the two images are dissimilar, A_j does have the term "brain primordium."

It is thus natural to consider the regions related to a concerned term, e.g., for "brain primordium" we only consider the head region; then, for a new image, we simply compare with its head region. However, this is infeasible for our task, since we do not know which term is related to which region in which image.

To address this challenge, we first need to represent the objects of interest. Considering that the terms are related to regions rather than the whole images, we extract many local patches from each image, and then represent each patch using a feature vector. Here, we extract dense regular patches from images [22], [23], [24], and use visual and spatial features to represent each patch. The radius and spacing of each regular patch are set to 16 pixels, and thus a total of 133 patches are extracted from each image since our images are of size 320×128 pixels [22]. For each patch, we use the popular SIFT descriptor [32], [33], an 128-dimensional vector, for the visual features, and directly used the coordinates of the center point of the patch as spatial features. Visual information are generally used in image annotation tasks. It may also be beneficial to include spatial information in addition to visual features.

Formally, let B_i denote the ith image group in the database, P_{ij} denote the jth image of B_i, and x_{ijk} denote the feature vector corresponding to the kth patch of image P_{ij}. The set of feature vectors collected from the images of B_i can be denoted as $X_i = \{x_{ijk}\}$ $(j = 1, 2, \ldots, m_i; k = 1, 2, \ldots, m_{ij})$, where m_i is the number of images contained in B_i, and m_{ij} is the number of patches extracted from the image P_{ij}. For convenience, we simply use $X_i = \{x_{is}\} = \{(x_{is\_0}, x_{is\_1})\}$ $(s = 1, 2, \ldots, n_i)$ to represent the collection of all the feature vectors gathered from B_i, where x_{is_0} and x_{is_1} denote the visual features and the spatial features of the sth patch of B_i, respectively; $n_i = \sum_{j=1}^{m_i} m_{ij}$. Here, X_i is called a *bag* of instances (or *bag* of feature vectors), and each feature vector x_{is} in the bag is called an *instance*. Note that in our study, we extract 133 patches per image, i.e., m_{ij} is a constant 133. Thus, if an image group contains three images, it forms a bag of $3 \times 133 = 399$ instances. We use $Y_i = \{y_{ip}\}$ $(p = 1, 2, \ldots, l_i)$ to represent the annotation terms of B_i, where l_i is the number of terms assigned to B_i. Therefore, from the view of machine learning, the automatic *Drosophila* gene expression annotation task can be considered as a learning problem of building a predictive model from a training set $\{(X_i, Y_i)\}(i = 1, 2, \ldots, n)$ to predict a set of proper labels Y of an unseen image group X.

This learning problem differs from the conventional supervised learning which learns concepts from objects each represented by a single feature vector and associated with a single label. Interestingly, this learning problem falls exactly into a new machine learning framework, MIML learning [56], [57]. MIML is motivated by learning problems involving complicated objects represented by a set of feature vectors and associated with multiple class labels. For example, in the text categorization task, a text document may belong to several categories simultaneously and can be represented by multiple feature vectors (in the same feature space) each corresponds to one section. Formally, let $\mathcal{X}$ denote the instance space and $\mathcal{Y}$ the class labels. MIML tries to learn a function $f : 2^{\mathcal{X}} \rightarrow 2^{\mathcal{Y}}$ from a training set $\{(X_i, Y_i)\} = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$, where $X_i = \{x_{i1}, x_{i2}, \ldots, x_{i,n_i}\} \subseteq \mathcal{X}$ is a set of instances, and $Y_i = \{y_{i1}, y_{i2}, \ldots, y_{i,l_i}\} \subseteq \mathcal{Y}$ is a set of labels. It is obvious that our concerned *Drosophila* gene expression annotation task is a typical MIML learning problem, and thus can be solved by MIML learning techniques.

3 Methods

3.1 The MIMLSVM$^+$ Method

In this section, we present an MIML support vector machine algorithm to address the task of *Drosophila* gene expression pattern annotation.

Two MIML algorithms, MIMLBoost and MIMLSVM, have been proposed in [57]. MIMLBoost solves MIML problems by degenerating the problems into multi-instance single-label problems through adding pseudolabels to each instance, while MIMLSVM solves MIML problems by degenerating the problems into single-instance multi-label problems through a specific clustering process. Both MIMLBoost and MIMLSVM have been shown to be effective [57], however, they were not designed for large-scale learning problems, e.g., our *Drosophila* computational annotation problem. Thus, new algorithms that can efficiently address large-scale data are desired.

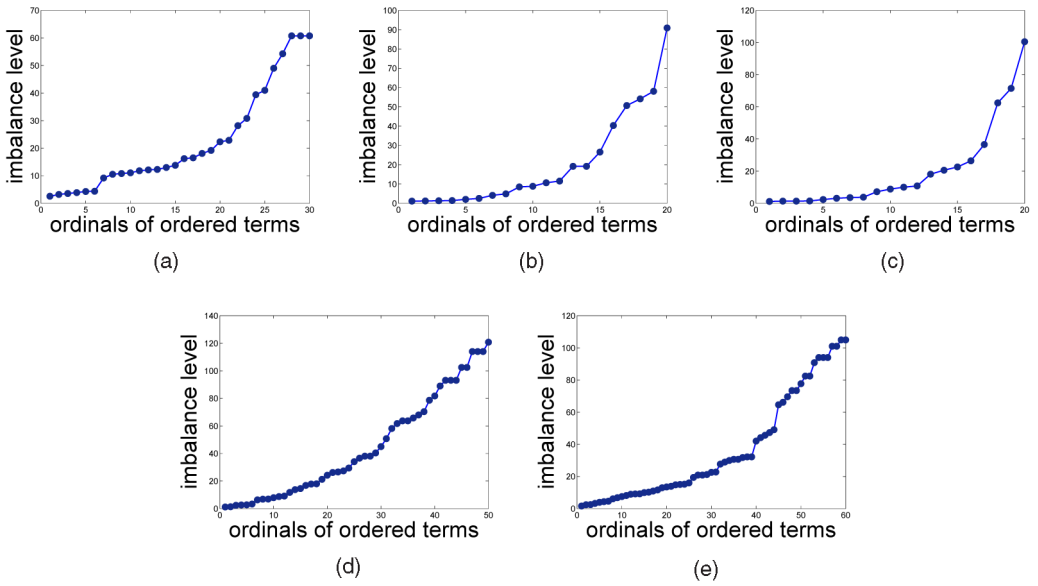


Fig. 3. The class imbalance levels of the annotation terms in different developmental stage ranges. For each developmental stage range, we sort the annotation terms in ascending order according to their imbalance levels, and use the horizontal axis in each subplot to represent the ordinals of ordered terms. (a) stage range 4-6, (b) stage range 7-8, (c) stage range 9-10, (d) stage range 11-12, and (e) stage range 13-16.

Support vector machine (SVM) [3], [44] is a widely used machine learning technique and has been applied successfully to many real-world applications [8]. For classification problems, SVM implements a large margin classifier by solving a quadratic optimization program based on the principle of structural risk minimization. Conventional SVM deals with traditional learning problems where the object is represented by a single instance and associated with a single class label. MIMLSVM [57] is designed for MIML learning problems based on SVM. In this section, we present a different SVM method, MIMLSVM$^+$, for MIML learning.

In MIMLSVM$^+$, we simply employ a degeneration strategy which decomposes the learning of multiple labels into a series of binary classification tasks. That is, we construct an SVM for each term; for a concerned term, we collect all the image groups with this term as positive samples, and the image groups without the term as negative samples. Generally, for a given developmental stage range, only a few image groups are associated with a particular term. Therefore, the class imbalance problem [45], [53] has to be taken into account; otherwise, the obtained classifier will be biased toward classifying test samples to be negative.

To illustrate the phenomenon of the class imbalance problem encountered in our annotation task, in Fig. 3 we plot the class imbalance levels of the terms to be analyzed under different developmental stage ranges. Here, the class imbalance level is defined as the number of negative samples divided by the number of positive samples. It can be seen from Fig. 3 that in all stage ranges, most terms suffer from class imbalance.

Suppose n is the number of training image groups; $y \in \mathcal{Y}$ is an annotation term; X_i is the bag of instances (local feature vectors) extracted from the ith image group in the training set. For each term y, let $\varphi(X_i, y)$ be the indicator function defined as: $\varphi(X_i, y) = 1$ if y is attached to the image group i, and $\varphi(X_i, y) = -1$ otherwise. The resulting SVM classification model involves the following optimization problem:

$$\begin{aligned} &\min_{w_y, b_y, \xi_{iy}} \ \frac{1}{2}\|w_y\|^2 + C\sum_{i=1}^{n}\xi_{iy}\tau_{iy} \\ &\text{s.t.}: \quad \varphi(X_i, y)(\langle w_y, \phi(X_i)\rangle + b_y) \geq 1 - \xi_{iy} \\ &\qquad \xi_{iy} \geq 0 \qquad (i = 1, 2, \ldots, n), \end{aligned} \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product; $\phi(X_i)$ is the function which maps the bag of instances X_i to a higher dimensional space $\mathcal{H}$; w_y and b_y are the parameters for representing a linear discriminant function in $\mathcal{H}$; ξ_{iy} is the nonnegative slack variable introduced in the constraints to permit some training bags to be misclassified [7]; $\|w_y\|^2$ is used to reflect the model complexity [44]; C is the parameter to trade off the model complexity and the sum of losses of the training bags; τ_{iy} is the amplification coefficient of the loss ξ_i for handing the class imbalance problem. Here, τ_{iy} is defined as

$$\tau_{iy} = \frac{1 + \varphi(X_i, y)}{2} R_y + \frac{1 - \varphi(X_i, y)}{2}, \tag{2}$$

where R_y is the class imbalance level of y, and is estimated on the training set through dividing the number of negative bags by the number of positive bags. It can be seen that $\tau_{iy} = R_y$ if $\varphi(X_i, y) = 1$, while $\tau_{iy} = 1$ when $\varphi(X_i, y) = -1$; this implies that the penalty on the loss of positive bag is amplified by R_y, and therefore, the penalties for the losses on the positive and negative bags are $R_y C$ and C, respectively. We can set a larger penalty factor on the loss of positive bags compared with that of negative ones if there are many more negatives than positives. This "rescaling" strategy is popular in handling class-imbalance problems [45], [53], [54].

The procedure of solving the optimization problem (1) is called "training," and it has been proved that the training procedure depends only on the data through dot products in $\mathcal{H}$ [3], [44], i.e., on functions of the form $K(X_i, X_j) = \langle \phi(X_i), \phi(X_j) \rangle$ where $K(X_i, X_j)$ is called "kernel function;" X_i and X_j denote two arbitrary training bags. In fact, kernel function is very crucial in support vector machine and needs to be predefined [44]. Different kernel functions can result in different support vector machines. Note that the kernel

TABLE 1
The MIMLSVM$^+$ Algorithm

Y = MIMLSVM$^+(\mathcal{S}, X)$

Input: $\mathcal{S}$ - the training set
X - the test bag of instances

Output: Y - the set of predicted annotation terms of X

1) For training set $\mathcal{S} = \{(X_i, Y_i)\}(i = 1, 2, \cdots, n)$, calculate the multi-instance kernel matrix $[K_{MI}(X_i, X_j)]$ $(i, j = 1, 2, \cdots, n)$.

2) For each label $y \in \mathcal{Y}$, derive a dataset $\mathcal{S}_y = \{(X_i, \varphi(X_i, y))|i = 1, 2, \cdots, n\}$, and train an SVM f_y = SVMTrain$(\mathcal{S}_y)$ based on $[K_{MI}(X_i, X_j)]$ using the formulation (1).

3) For a test bag X, its annotations are obtained by:

$$Y = \{\arg\max_{y \in \mathcal{Y}} f_y(X) | f_y(X) < 0, \forall y \in \mathcal{Y}\} \bigcup \{y | f_y(X) \geq 0, y \in \mathcal{Y}\}$$

function $K(X_i, X_j)$ employed here is based on bag of instances instead of single feature vectors. Theoretically, any kernel defined on set of instances [19] can be used to compute $K(X_i, X_j)$. In the current work, we simply adopt the well-known multi-instance kernel [13], which is defined as

$$K_{MI}(X_i, X_j) = \frac{1}{n_i n_j} \sum_{(x_{is\_0}, x_{is\_1}) \in X_i} \sum_{(x_{jz\_0}, x_{jz\_1}) \in X_j} e^{-\gamma_1 \|x_{is\_0} - x_{jz\_0}\|^2 - \gamma_2 \|x_{is\_1} - x_{jz\_1}\|^2}, \quad (3)$$

where n_i and n_j are normalization factors for taking into account the sizes of bags, defined as the numbers of instances in the bags X_i and X_j, respectively. Intuitively, the term $\|x_{is\_0} - x_{jz\_0}\|^2$ measures the similarity of visual features between the expression patterns of two patches, while $\|x_{is\_1} - x_{jz\_1}\|^2$ calculates the spatial distance between two patches. The visual and spatial information are combined with different weights γ_1 and γ_2 through the kernel trick. It is easy to verify that $K_{MI}(X_i, X_j)$ is a valid kernel because only dot product on two Gaussian kernels is presented in K_{MI}.

Once the kernel function $K_{MI}(X_i, X_j)$ is defined, the optimization problem (1) can be solved by many methods [8] implemented in many software packages, e.g., [5], [26]. Let X denote the bag of instances extracted from an unseen image group. Then, the resulting classification model for annotating term y to X is

$$f_y(X) = \langle w_y, \phi(X) \rangle + b_y = \sum_{i=1}^{n} \alpha_{iy} \varphi(X_i, y) K_{MI}(X_i, X) + b_y, \quad (4)$$

where $f_y(X)$ is the output which can be considered as a score indicating how likely X should be annotated with the term y.

After the term-specific predictive models $f_y(X)$ $(y \in \mathcal{Y})$ have been constructed for all the annotation terms, for an unseen image group, we employ the T-criterion [2], [57] to make the final annotations based on the outputs of the SVMs. That is, the unseen image group is first fed to every classification model as described in (4); then, the image group is labeled with all the terms whose corresponding predictive models produce positive outputs. In the case where all the predictive models output negative values, the image group is labeled with the term whose predictive model has the maximum output value. The pseudocode of MIMLSVM$^+$ is shown in Table 1.

3.2 E-MIMLSVM$^+$: Incorporating Term Correlations

MIMLSVM$^+$ decomposes the multi-label problem into a series of independent binary learning tasks each for a term. In this way, the correlation between the terms is neglected, while many studies have shown that an appropriate exploitation of the correlation will improve the performance since some information from one term may be helpful to other terms [31], [43], [50], [56]. Herein, we present the E-MIMLSVM$^+$ method which extends MIMLSVM$^+$ by incorporating the term correlations.

We incorporate the term correlations by utilizing multi-task learning techniques [1], [11], [12], [47] which consider the labeling of each term as a task. Since MIMLSVM$^+$ is a support vector machine algorithm, it is natural to employ the kernel-based multitask learning framework [12] for the extension. For convenience, we denote the procedure of learning a classification model for annotating the term $y \in \mathcal{Y}$ to image groups as task y. Suppose the classification function for annotating the term $y \in \mathcal{Y}$ can be written as

$$f_y(X) = \langle w_y, \phi(X) \rangle + b = \langle (w_0 + v_y), \phi(X) \rangle + b, \quad (5)$$

where w_0 is used to reflect the commonalities shared by different learning tasks, and v_y is the task-specific model parameter used to measure how distinct the task y is. Let $|\mathcal{Y}|$ indicate the number of terms in the label space $\mathcal{Y}$. The goal of multitask learning is to estimate w_0, v_t $(t = 1, 2, \ldots, |\mathcal{Y}|)$ as well as b simultaneously for achieving a better performance compared to learning the tasks independently. In this work, we extend the formulation (1) to

$$\begin{aligned} \min_{w_0, v_y, b, \xi_{iy}} & \frac{1}{2} \left(\sum_{y \in \mathcal{Y}} \|v_y\|^2 + \mu \|w_0\|^2 \right) + C \sum_{y \in \mathcal{Y}} \sum_{i=1}^{n} \xi_{iy} \tau_{iy} \\ s.t. : & \varphi(X_i, y)(\langle (w_0 + v_y), \phi(X_i) \rangle + b) \geq 1 - \xi_{iy} \\ & \xi_{iy} \geq 0, \end{aligned} \quad (6)$$

where μ is employed to reflect the similarity between the tasks. The larger the μ, the smaller the w_0, and vice versa. Thus, μ can be used to tune the closeness between each model

TABLE 2
The E-MIMLSVM$^+$ Algorithm

Y = E-MIMLSVM$^+(\mathcal{S}, K, X)$

Input: $\mathcal{S}$ - the training set
K - the number of clusters
X - the test bag of instances

Output: Y - the predicted annotation terms of X

1) Clustering the terms $\mathcal{Y}$ into K subgroups: $\mathcal{Y} = L_1 \bigcup L_2 \bigcup \cdots \bigcup L_K$ based on the label indicator matrix $Y = [\varphi(X_i, y)], \quad (i = 1, \cdots, n;\ y \in \mathcal{Y})$;

2) (a) For each subgroup L_k $(k = 1, \cdots, K)$, derive a data set from $\mathcal{S}$: $\mathcal{S}_k = \{(X_i, Y_i \bigcap L_k) \mid i = 1, 2, \cdots, n\}$;

(b) Calculate the multi-instance multi-task kernel matrix $[K_{ts}(X_i, X_j)] \quad (i, j = 1, 2, \cdots, n;\ t, s \in L_k)$ based on $\mathcal{S}_k$;

(c) Train SVMs based on $[K_{ts}(X_i, X_j)]$ using the formulation (6): $[f_y]$ = SVMTrain$(\mathcal{S}_k),\ y \in L_k$

3) For a test bag X, its annotations can be obtained by:

$$Y = \{\arg\max_{y \in \mathcal{Y}} f_y(X) | f_y(X) < 0, \forall y \in \mathcal{Y}\} \bigcup \{y | f_y(X) \geq 0, y \in \mathcal{Y}\}$$

parameter v_y and the shared model parameter w_0. It can be seen that (6) is very similar to (1), while the main difference is that (6) transforms the problem of learning multiple classification tasks simultaneously into a single learning problem in the form of support vector machine. If $|\mathcal{Y}| = 1$, i.e., there is only one task to be tackled, (6) degenerates to (1).

The optimization problem (6) can be solved by the same solver of MIMLSVM$^+$, and the resulting classification function for annotating term $y \in \mathcal{Y}$ to an unseen image group X is

$$f_y(X) = \sum_{i=1}^{n} \sum_{t \in \mathcal{Y}} \alpha_{it} \varphi(X_i, t) K_{ty}(X_i, X) + b, \tag{7}$$

where K_{ty} is the kernel function. Different from the kernel function defined in MIMLSVM$^+$ which measures the similarity between two bags of the same task, K_{ty} measures the similarity between two bags coming from the task t and the task y, respectively. Formally, K_{ty} is defined as

$$\begin{aligned} K_{ty}(X_i, X_j) &= \left(\frac{1}{\mu} + \delta(t = y)\right) \langle \phi(X_i), \phi(X_j) \rangle \\ &= \left(\frac{1}{\mu} + \delta(t = y)\right) K_{MI}(X_i, X_j) \quad (t, y \in \mathcal{Y}), \end{aligned} \tag{8}$$

where $\delta(t = y) = 1$ if $t = y$, and $\delta(t = y) = 0$ otherwise. It can be seen from (8) that K_{ty} is defined not only between bags of instances but also between tasks. Therefore, K_{ty} can be called as "multi-instance multitask" kernel, which bridges the multi-instance kernel and the multitask kernel. It is obvious that the kernel K_{ty} is convenient to get from the multi-instance kernel K_{MI}.

It can be seen from (6) that all the models f_y are forced to be close to a common one parameterized by w_0. However, this is too restrictive in our annotation problem, since some terms may not be related at all. Therefore, we first employ a clustering process to partition the terms into some subgroups based on the correlations between terms. As the consequence, the tasks in each cluster may be strongly correlated. From biological view, related tasks can be considered as annotating coexpressed regions. Thus, the clustering process can be seen as identifying the potentially coexpressed regions. Formally, let $Y = [\varphi(X_i, y)]_{n \times |\mathcal{Y}|}$ denote the label indicator matrix, i.e., $Y_{ij} = \varphi(X_i, \mathcal{Y}(j))$, where n is the number of training bags. We directly use the Pearson's correlation coefficient computed on columns of Y as the measure of correlation between terms [24] for clustering. For the terms in each cluster, we employ the formulation (6) to build a predictive model for annotating these terms. Finally, the T-criterion is employed to combine the predicted labels for the test bags. The pseudocode for the whole process is summarized in Table 2. Note that, if the cluster number K equals to the number of terms in the label space $\mathcal{Y}$, i.e., $K = |\mathcal{Y}|$, each cluster will contain only one unique term; in this case the E-MIMLSVM$^+$ degenerates to MIMLSVM$^+$. Therefore, MIMLSVM$^+$ can be regarded as a special case of E-MIMLSVM$^+$.

Note that E-MIMLSVM$^+$ is generally more time consuming than MIMLSVM$^+$ since learning multiple tasks simultaneously will result in many more bags of instances involved in the optimization procedure. Meanwhile, the memory requirement of E-MIMLSVM$^+$ is also much more than that of MIMLSVM$^+$.

3.3 Software

The codes of our methods are available at http://lamda.nju.edu.cn/datacode/MIMLAnnotator.htm, which are implemented with Matlab 7.6 in windows.

4 Experiments and Results

4.1 Experimental Configuration

The early embryogenesis of *Drosophila* is divided into six developmental stage ranges [37], i.e., 1-3, 4-6, 7-8, 9-10, 11-12, and 13-16, and most of the CV terms are stage range specific. So, we annotate the gene expression pattern images according to stage ranges. Table 3 summarizes the numbers of the annotation terms, the numbers of image groups, and the total numbers of images from each stage range in the FlyExpress database used in our experiments.

TABLE 3
Statistics of the FlyExpress Database

Stage range	1-3	4-6	7-8	9-10	11-12	13-16
# of CV terms	2	42	24	45	102	119
# of image groups	1,783	1,081	877	1,072	2,113	2,816
# of images	2,525	5,579	3,085	3,540	11,310	15,434

Since the stage range 1-3 has only two terms, we skip this stage range and perform experiments on the other five stage ranges. It can be seen from Table 3 that various CV terms are contained in different stage ranges. For each stage range, we start from the top 10 CV terms with the largest numbers of image groups, and the number of CV terms is increased by 10 for each round until no CV terms can be added. In each round, all the image groups annotated with at least one of the included terms are picked out to construct an experimental data set. There are some CV terms associated with only a few image groups, and these CV terms are not considered in our experiments. Overall, there are a total of 18 experimental data sets over different stage ranges, denoted by $D_1, D_2, \ldots, D_{18}$. The corresponding stage ranges and the numbers of CV terms of the experimental data sets are summarized in Table 4.

On each data set, the image groups are randomly partitioned into a training set and a test set according to the ratio about 1:1 for each term in pervious studies [22], [24]. The training set is used to build a predictive model, and the test set is used to evaluate its performance. In addition, to produce a reliable performance estimation, the training/test splitting procedure is repeated for 30 times and the average performance is reported [22], [24]. In order to make a fair comparison with these works, we employ the same setting with the same partitions of the data sets and report the average performance as in previous studies [22], [24].

4.2 Performance Evaluation Criteria

In the literature, for learning problems where each object is assigned with multi-labels, the performance evaluation criteria can be divided into two categories: 1) criteria that are directly extended from traditional single-label evaluation measures to multi-labels [16], [23], [24], such as the macro F1, micro F1, and AUC (the Area Under ROC Curve); 2) criteria that are specifically designed for the setting of multi-labels [48], [56], such as the hamming loss, one-error, coverage, average precision, and ranking loss.

The definitions of these measures can be found in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2011.73, of this paper. Briefly, the larger the values of macro F1, micro F1, AUC, and average precision, the better the performance, while the smaller the values of hamming loss, one-error, coverage, and ranking loss, the better the performance. It should be noted that the hamming loss is calculated directly from "accuracy," which is a performance measure commonly used in conventional single-label leaning task. However, it has been shown that "accuracy" suffers seriously from class imbalance [9], [36]. We thus exclude the hamming loss and use the other seven criteria to assess the performance of annotation methods. Note that these criteria measure the annotation performance from different aspects, and it is rare that one algorithm beats another algorithm on all these criteria.

4.3 Results and Analyses of $\mathrm{MIMLSVM}^+$

We first compare $\mathrm{MIMLSVM}^+$ with previous approaches on *Drosophila* gene expression pattern annotation introduced in Section 1. The competing approaches include the pyramid match kernel based method [23] and the two bag-of-words based methods [22], [24].

As for the pyramid match kernel based method [23], which is denoted by PMK in this paper, three different kernel combination schemes, i.e., star, clique [6], and Kernel Canonical Correlation Analysis (KCCA) [18] were employed to combine the kernels of different local descriptors, and produced three sets of annotation results. For each criterion, only the best result among these three schemes is reported as the performance of PMK in this paper. We use "ML_{SS}-BOW" to denote the method proposed in [22] which applied the shared subspace multitask formulation to implement annotation, and "ML_{GR}-SBOW" to denote the graph regularization based multitask learning method proposed in [24]; both of these two methods are based on the bag-of-words (BOW) representation scheme. The annotation performance of PMK and ML_{GR}-SBOW can be calculated directly from the classification results reported in [24]. Since the detailed prediction results of ML_{SS}-BOW in [22] are not available, we rerun ML_{SS}-BOW on all the 18 data sets and report the performance. For $\mathrm{MIMLSVM}^+$, we set the kernel parameters γ_1 and γ_2 as suggested in [13], i.e., γ_1 and γ_2 should be set in the order of magnitude of $1/(2d_1^2)$ and $1/(2d_2^2)$ or lower, respectively, where $d_1 = 128$ and $d_2 = 2$ are the dimensions of SIFT descriptor and region coordinates, respectively. Therefore, we set $\gamma_1 = 10^{-5}$ and $\gamma_2 = 10^{-2}$ in our experiments. The penalty factor C for $\mathrm{MIMLSVM}^+$ is tuned through 10-fold cross-validation on the training set in which the whole train set is randomly partitioned into 10 approximately equal-size subsets, and then for each of 10 trials we

TABLE 4
Stage Ranges and Numbers of CV Terms of the Experimental Data Sets

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}
stage range	4-6	4-6	4-6	7-8	7-8	9-10	9-10	11-12	11-12	11-12	11-12	11-12	13-16	13-16	13-16	13-16	13-16	13-16
# of terms	10	20	30	10	20	10	20	10	20	30	40	50	10	20	30	40	50	60

use a different subset as test set while the union of the remaining nine subsets is used as training set.

Fig. 4 plots the comparison results. Note that the performance of E-MIMLSVM$^+$, which will be discussed in the next section, is also plotted in Fig. 4. The win/tie/loss counts obtained after paired t-tests at 95 percent significant level of MIMLSVM$^+$ versus PMK, ML$_{SS}$-BOW and ML$_{GR}$-SBOW in terms of all the seven criteria on all the 18 experimental data sets are summarized in Table 5. The table provides a systematic comparison between MIMLSVM$^+$ and the three baselines.

It can be observed from Fig. 4 and Table 5 that MIMLSVM$^+$ outperforms PMK consistently. Both MIMLSVM$^+$ and PMK are kernel-based methods, however, PMK treats the annotation problem from the view of conventional supervised learning while MIMLSVM$^+$ works under MIML framework. Their different formulations result in different types of kernels employed and thus lead to different annotation performance. For the annotation task, the MIML formulation can help to capture more discriminant information from image groups than the conventional supervised formulation, and leads to much better annotation performance.

We can also observe that MIMLSVM$^+$ outperforms ML$_{SS}$-BOW consistently. ML$_{SS}$-BOW solves the annotation task by conventional single-instance multi-label formulation based on the popular bag-of-words representation in image annotation problems [38], and takes into account term correlations. However, as can be seen from Fig. 4 and Table 5, our MIMLSVM$^+$ is superior to the ML$_{SS}$-BOW, though MIMLSVM$^+$ is a MIML solution by degeneration which does not fully exploit the power of MIML framework.

ML$_{GR}$-SBOW achieves better annotation performance than ML$_{SS}$-BOW. We can observe that, however, on most cases MIMLSVM$^+$ is superior to ML$_{GR}$-SBOW, and ML$_{GR}$-SBOW is comparable to MIMLSVM$^+$ on only a few cases. To further examine whether MIMLSVM$^+$ significantly outperforms ML$_{GR}$-SBOW considering that there are a few comparable cases, we conduct the sign tests at 95 percent significant level on the t-tests results on all the 18 experimental data sets for every criterion. The results indicate that MIMLSVM$^+$ is significantly better than ML$_{GR}$-SBOW in terms of all the annotation evaluation performance criteria.

The above observations conclude that for the *Drosophila* gene expression pattern annotation problem, MIMLSVM$^+$ is superior to previous computational annotation approaches although it has not taken into account the term correlations. Our results demonstrate the power of the MIML learning framework for the annotation for *in situ* expression patterns.

4.4 Results and Analyses of E-MIMLSVM$^+$

Compared with MIMLSVM$^+$, E-MIMLSVM$^+$ exploits term correlations when building classifiers to annotate CV terms. The multi-instance multitask kernel K_{ty} can be easily computed from the multi-instance kernel used in MIMLSVM$^+$, as shown in the (8). The parameter μ and the penalty factor C in optimization (6) can be tuned by double cross-validation on training sets. To cluster the terms in the label space $\mathcal{Y}$ into subgroups, we use the k-means algorithm [10] to partition the CV terms. The number of clusters K is important yet hard to decide, and there is no guideline for selecting a proper K. In this work, we simply let $K = q|\mathcal{Y}|$, where q is a real number which is called "scattering ratio" in this paper. The larger the value of q, the larger the number of clusters. When $q = 1$, E-MIMLSVM$^+$ degenerates to MIMLSVM$^+$. In our experiments we set $q = 0.5$, and will discuss the influence of q at the end of this section.

The detailed annotation performance of E-MIMLSVM$^+$ is plotted in Fig. 4. To compare E-MIMLSVM$^+$ with MIMLSVM$^+$, the win/tie/loss counts of E-MIMLSVM$^+$ versus MIMLSVM$^+$ in terms of the seven performance criteria are summarized in Table 6. We can observe from Fig. 4 and Table 6 that on most cases, E-MIMLSVM$^+$ is superior to or at least comparable with MIMLSVM$^+$. Sign tests at 95 percent significant level on the t-tests results indicate that in terms of micro F1, one-error, coverage, average precision, and ranking loss, E-MIMLSVM$^+$ is significantly better than MIMLSVM$^+$, while in terms of macro F1 and AUC, E-MIMLSVM$^+$ presents comparable performance with MIMLSVM$^+$. This validates the exploitation of term correlations in E-MIMLSVM$^+$.

Nevertheless, the performance improvement gained by E-MIMLSVM$^+$ is not as much as expected. Previous studies [22], [24] indicated that much improvement can be obtained by taking into account the term correlations in building classifiers. Fig. 5 shows the performance improvement of E-MIMLSVM$^+$, ML$_{GR}$-SBOW, and ML$_{SS}$-BOW compared with their degeneration versions which do not consider term correlations, respectively, in terms of micro F1 as an example. The key difference between our work and the previous studies [22], [24] lies in the fact that previous studies employed the single-instance multi-label formulation, while our study is based on MIML formulation. Meanwhile, the degenerated version of E-MIMLSVM$^+$, i.e., MIMLSVM$^+$, is significantly better than ML$_{GR}$-SBOW and ML$_{SS}$-BOW though they have exploited term correlation information. Therefore, we conjecture that the improvement of E-MIMLSVM$^+$ over MIMLSVM$^+$ is not as much as expected because some useful information has already been captured by the MIML formulation.

To verify the conjecture, we examine whether MIMLSVM$^+$ is able to construct similar predictive models for correlated terms, though MIMLSVM$^+$ does not consider term correlations. In this study, we use the data set D_3 of stage range 4-6 and 30 terms for illustration, since on this data set E-MIMLSVM$^+$ shows comparable performance with MIMLSVM$^+$ in terms of all the criteria. Note that the data set is partitioned into training/test set repeatedly for 30 trials to obtain reliable performance estimations for CV terms. For each trial, let L denote the matrix representing the degrees of correlation between paired CV terms, i.e., $L = [l_{ty}]\ (t, y \in \mathcal{Y})$, where l_{ty} is the Pearson correlation coefficient between the terms t and y; P is the matrix representing the similarities of two SVM models, i.e., $P = [p_{ty}]\ (t, y \in \mathcal{Y})$, where p_{ty} is defined as the cosine of the angle between the model parameters $\mathcal{A}_t$ and $\mathcal{A}_y$. Here, $\mathcal{A}_t$ represents the model parameters of the obtained SVM f_t for annotating the term t, and is defined as $\mathcal{A}_t = [\alpha_{1t},\ \alpha_{2t}, \ldots, \alpha_{nt}, b_t]^T$. To study whether a larger l_{ty} can result in a higher value of p_{ty}, we calculate the Pearson correlation coefficient $\lambda_{LP} = \lambda(\{l_{ty}\}, \{p_{ty}\})$ for validation, where $\{l_{ty}\}$ represents the vector consisting of all the elements of L, and $\{p_{ty}\}$ is the corresponding vector consisting of the elements of P; $\lambda(\cdot, \cdot)$

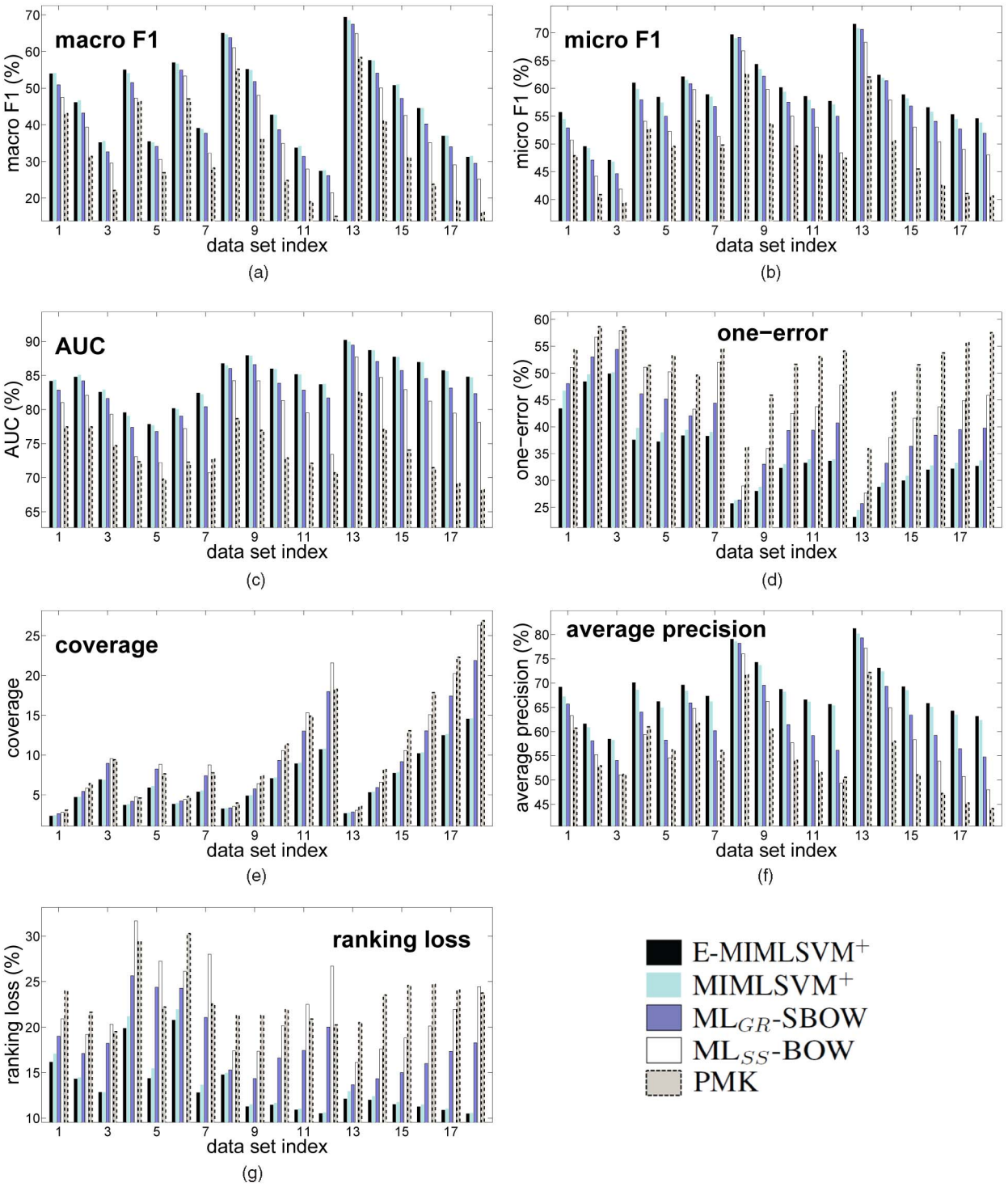


Fig. 4. Annotation performance of E-MIMLSVM$^+$, MIMLSVM$^+$, ML$_{GR}$-SBOW, ML$_{SS}$-BOW, and PMK. The horizontal axis in each subplot represents the index of experimental data sets $D_1, \ldots, D_{18}$. The larger the values of macro F1, micro F1, AUC, and average precision, the better the performance; the smaller the values of one-error, coverage, and ranking loss, the better the performance.

is used to calculate the Pearson correlation coefficient. If correlated terms result in similar predictive models, λ_{LP} will be high. Note that the Pearson correlation coefficient can only reflect the strength of linear dependence between variables.

To further study the relationship between correlation of terms and the similarity of the corresponding models, we also defined a measure, h_ρ, called "match accuracy" given by

$$h_\rho = \frac{\Psi(L_\rho, P_\rho)}{N_\rho}, \tag{9}$$

where ρ is a predefined positive value; L_ρ is the binary matrix indicating the entries of L whose values are no less than ρ, i.e., $L_\rho = [l_\rho(t,y)]\ (t, y \in \mathcal{Y})$ where $l_\rho(t,y) = \delta(l_{ty} \geq \rho)$; $N_\rho = \sum_t \sum_y l_\rho(t,y)$ represents the number of nonzero entries of L_ρ; P_ρ is the binary matrix indicating the top N_ρ entries of P with the maximum values, and can be obtained directly from P by

TABLE 5
The Win/Tie/Loss Counts of MIMLSVM^+ versus ML_{GR}-SBOW, ML_{SS}-BOW, and PMK After Paired t-Tests at 95 Percent Significance Level

Criterion	ML_{GR}-SBOW			ML_{SS}-BOW			PMK		
	win	tie	loss	win	tie	loss	win	tie	loss
macro F1	18	0	0	18	0	0	18	0	0
micro F1	16	2	0	18	0	0	18	0	0
AUC	18	0	0	18	0	0	18	0	0
one-error	16	2	0	18	0	0	18	0	0
coverage	18	0	0	18	0	0	18	0	0
average precision	17	1	0	18	0	0	18	0	0
ranking loss	17	1	0	18	0	0	18	0	0

setting the top N_ρ entries of P with the maximum values to 1 while the others to 0; $\Psi(L_\rho, P_\rho)$ is the match function counting the number of overlapped positive entries of L and P. $\Psi(L_\rho, P_\rho)$ is defined as

$$\Psi(L_\rho, P_\rho) = \sum_t \sum_y l_\rho(t, y) p_\rho(t, y) \ . \tag{10}$$

Intuitively, L_ρ locates the term pairs with correlation coefficients no less than the threshold ρ, and P_ρ locates the most similar model pairs of the same size to the located term pairs of L_ρ. Therefore, h_ρ measures the percentage of similar model pairs whose corresponding term pairs have correlation coefficients no less than ρ.

We calculate the λ_{LP} and h_ρ on the 30 distinct training sets extracted from D_3, and report the mean and standard deviation of λ_{LP} and h_ρ. The obtained correlation coefficient is $\lambda_{LP} = 0.7667 \pm 0.0131$. This reflects the existence of strong correlation between the term correlations and the model similarities. Fig. 6 plots the $h_\rho \sim \rho$ curve. It can be observed from the figure that the match index h_ρ is increasing along with the increase of the threshold ρ. This implies that the larger the correlation between terms, the more similar the corresponding models. This result confirms our conjecture that although MIMLSVM^+ does not explicitly take the term correlations into account, the MIML formulation has already captured some useful term correlation information, and thus MIMLSVM^+ can achieve good annotation performance and

TABLE 6
The Win/Tie/Loss Counts of E-MIMLSVM^+ versus MIMLSVM^+ After Paired t-Tests at 95 Percent Significance Level

Evaluation criterion	win	tie	loss
macro F1	2	16	0
micro F1	15	3	0
AUC	3	14	1
one-error	13	5	0
coverage	14	4	0
average precision	15	3	0
ranking loss	12	6	0

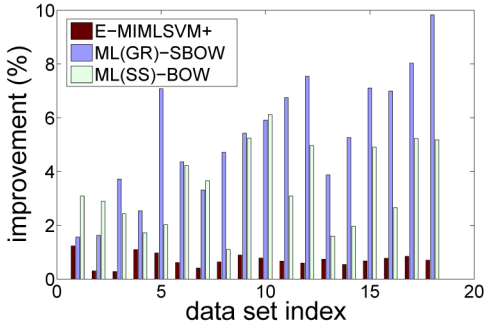


Fig. 5. The performance improvement in terms of micro F1 of E-MIMLSVM^+, ML_{GR}-SBOW, and ML_{SS}-BOW compared with their degenerated versions which do not consider term correlations, respectively.

the improvement of E-MIMLSVM^+ over MIMLSVM^+ is not as large as expected.

In general, if one'd like to build an efficient automated annotation system for large-scale data, MIMLSVM^+ is a good option since it is simple and more efficient than methods which take into account term correlations explicitly. While if the computational load is not a concern and a higher prediction accuracy is desired, E-MIMLSVM^+ is preferred.

Recall that the performance of E-MIMLSVM^+ is dependent on the "scattering ration" q which determines the number of term clusters. To study the influence of q, we conduct additional experiments. Table 7 summarizes the annotation performance of E-MIMLSVM^+ in terms of different criteria with q varying from 0.9 to 0.3 with an interval of 0.1. It can be observed from the table that the performance advantage of E-MIMLSVM^+ against MIMLSVM^+ is relatively small when q is quite large or quite small. Indeed, when q is quite large, many less-related terms may be clustered together and regarded as correlated; while when q is quite small, many related terms may be put into different small clusters and regarded as noncorrelated. In general, a moderate value of q is preferred, such as $q = 0.5$ adopted in our experiments.

5 Discussions

5.1 Identifying Problematic BDGP Annotations

The necessity of building a computational annotation system stems from the need of an objective approach that

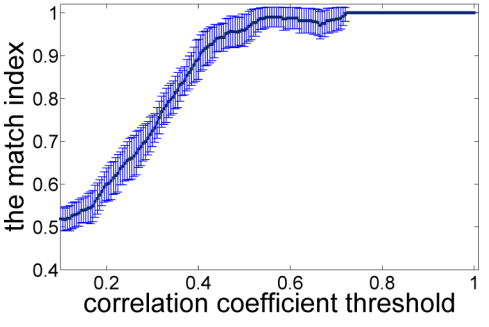


Fig. 6. The relationship between the match index h_ρ and the threshold of correlation coefficient ρ. When $\rho \geq 0.7250$, $h_\rho = 1$ since only diagonal items are left for L and P on each training split S_j.

TABLE 7
The Win/Tie/Loss Counts of E-MIMLSVM$^+$ versus MIMLSVM$^+$ After Paired t-Tests at 95 Percent Significance Level, with the "Scattering Ratio" q Varying from 0.9 to 0.3 with an Interval 0.1

q	macro F1	micro F1	AUC	ranking loss	one-error	coverage	average precision
0.9	0 / 18 / 0	3 /15 / 0	0 / 18 / 0	0 / 18 / 0	2 / 16 / 0	1 / 17 / 0	5 / 13 / 0
0.8	1 / 17 / 0	9 / 9 / 0	0 / 18 / 0	5 / 13 / 0	6 / 12 / 0	9 / 9 / 0	7 / 11 / 0
0.7	2 / 16 / 0	11 / 7 / 0	1 / 17 / 0	6 / 12 / 0	7 / 11 / 0	13 / 5 / 0	8 / 10 / 0
0.6	1 / 17 / 0	12 / 6 / 0	1 / 17 / 0	9 / 9 / 0	8 / 10 / 0	15 / 3 / 0	9 / 9 / 0
0.5	2 / 16 / 0	15 / 3 / 0	3 / 14 / 1	12 / 6 / 0	13 / 5 / 0	14 / 4 / 0	15 / 3 / 0
0.4	0 / 10 / 8	16 / 2 / 0	1 / 13 / 4	11 / 7 / 0	13 / 5 / 0	7 / 11 / 0	12 / 6 / 0
0.3	1 / 4 / 13	14 / 4 / 0	0 / 10 / 8	8 / 9 / 1	14 / 4 / 0	6 / 7 / 5	11 / 7 / 0

can produce reliable annotations for expression images within a relatively small time cost. This would be very helpful since qualified human curators are experts who have received over 20 years education and are costly. Fig. 7 shows some annotation results obtained by MIMLSVM$^+$ on the data set D_{14} for annotating 20 CV terms to image groups of stage range 13-16. Note that since D_{14} is partitioned into 30 training/test splits in our work, we simply selected some results of the first trial for illustration. It can be seen from Fig. 7 that overall, the automated annotation algorithm, MIMLSVM$^+$, produces very promising results. A large portion of the predicted CV terms coincide well with the manual annotations. This confirms the feasibility of annotating gene expression pattern with our proposed method.

Among all the 1,438 test image groups of the first trial on D_{14}, there are 427 cases whose top-predicted terms are not identical to their BDGP annotation terms. This may be due to the mistakes produced by our predictive model or potentially by the human curator. For further examination, we picked out a small portion of these cases and invited an expert to carefully reexamine the annotations. Interestingly, some image groups were found to be improperly manually annotated in the BDGP database. For example, the image groups of genes *cad* and *Rab-RP3* as shown in Fig. 7 were misannotated. For the gene *cad*, the term "embryonic midgut" was predicted by the MIMLSVM$^+$. Indeed, part of the midgut is stained and can be observed clearly from the images; however, the midgut is not identified by the BDGP curator. Similarly, for the gene *Rab-RP3*, the CV term "embryonic midgut" is also the top-predicted term of the MIMLSVM$^+$ yet not a BDGP term. These examples imply that though the human annotators do a great job in annotating the BDGP images, there may exist some problematic cases. It is expected no *in situ* image databases will be perfect. Manually reexamining all the annotated images would require tremendous efforts and a tremendous amount of time. Therefore, only a very small number of image groups were reexamined in our study.

Meanwhile, although our methods achieve good annotation performance, different kinds of errors can be observed from the prediction results, as illustrated by the prediction results of genes *r-l* and *CG9518* in Fig. 7. Prediction errors can arise from different reasons, including the misannotations of the database discussed above and the insufficiencies of training samples for some anatomical structures. For example, Fig. 8 plots the number of positive training samples of all the 60 CV terms of the experimental data set D_{18}. It can be observed from Fig. 8 that many terms possess very small numbers of training samples. This leads to the inadequate learning of these terms and results in their low annotation performance.

We believe that the expertise of human curators would not be simply replaced by computational annotation approaches. However, when human curators are annotating new images, it would be very helpful to launch our approach simultaneously, to help validate and double check the annotation results for increasing the reliability of the annotations.

5.2 Advantages to Bag-of-Words Representation

The automated gene expression annotation task can be considered as an image annotation problem from the view of machine learning, such as image classification and object recognition [4], [25], [38]. Thus, the ideas and methods employed in traditional image annotation studies can be borrowed to address the annotation problem. One representative solution is the learning scheme based on the bag-of-words representation of objects (images or image groups) [25], [46], [49]. In this scheme, all the local patches generated from training objects are first clustered to create some representative local patterns used as "keywords;" then, for an individual object, each of its local patches is compared to all the keywords, and the closest keyword is used to represented it; finally, the object is represented by a feature vector where each element counts the frequencies of a distinct keyword appearing in the object. After objects represented by feature vectors, machine learning methods can be applied to deal with the learning problems. This strategy has been adopted in [23] and [24] for addressing the *Drosophila* annotation problem. The information conveyed by the bag-of-words representation scheme, however, is just the statistics of "keywords" appearing in objects. This is an indirect way for describing objects and may result in the loss of discriminant information. For example, when a local pattern is re-represented by its closest global "keyword,"

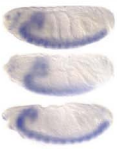
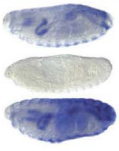
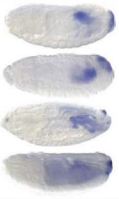
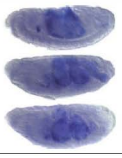
Genes	Images	BDGP terms	Predicted terms
CG6765		ventral nerve cord embryonic brain embryonic central nervous system sensory system head	ventral nerve cord embryonic brain embryonic central nervous system sensory system head
r-1		embryonic midgut embryonic Malpighian tubule embryonic hindgut embryonic anal pad embryonic/larval muscle system dorsal prothoracic pharyngeal muscle	embryonic midgut embryonic Malpighian tubule embryonic hindgut embryonic anal pad embryonic/larval muscle system
CG9518		embryonic hindgut embryonic proventriculus embryonic ventral epidermis embryonic dorsal epidermis	embryonic head epidermis embryonic proventriculus embryonic ventral epidermis embryonic dorsal epidermis
cad		embryonic anal pad embryonic Malpighian tubule embryonic hindgut	embryonic midgut embryonic anal pad embryonic Malpighian tubule embryonic hindgut
Rab-RP3		ventral nerve cord embryonic brain	embryonic midgut ventral nerve cord embryonic brain

Fig. 7. Sample annotation results for annotating 20 CV terms to image groups under the stage range 13-16. BDGP terms denote the manual annotations in the BDGP database, and predicted terms denote the CV terms predicted by the MIMLSVM$^+$. For each image group, the predicted terms are ranked in descending order according to their MIMLSVM$^+$ output scores.

some specific information of this local region may have been lost. When the number of occurrences of "keywords" is simply aggregated to derive frequencies, useful relation information between the local patterns is neglected. In this paper, we use the MIML representation of image groups, and design learning algorithms based on the MIML learning framework to address the *Drosophila* annotation task. Our representation is helpful to preserve characteristics of local patterns since it represents local patterns directly using information of local regions. It is also possible to exploit the relationship information between local patterns [55]. Compared with bag-of-words, our representation scheme is more direct and natural.

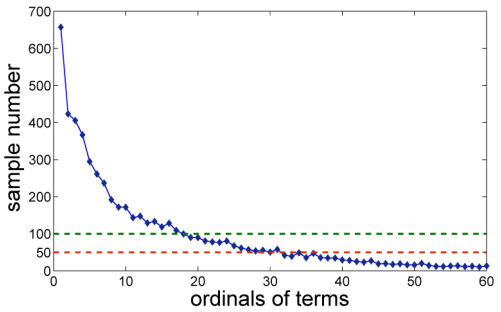


Fig. 8. Numbers of positive training samples of 60 annotation terms of the experimental data set D_{18}. The vertical axis indicates the number of positive training samples. Since the data set is partitioned into training/test sets for 30 times, the average number of positive training samples across the 30 trials is presented for each term.

5.3 Potential of MIML to Other Bioimages

It is important to note that many computational annotation methods, such as the ones described in [22], [52], and [24], require the view information (lateral, dorsal, or others) of images for annotating the anatomical terms. While in our MIML learning methods, all the patches extracted from images of different views are simply collected together to create a bag of instances, and no information about views is utilized. This simplifies the annotation process in some sense, and makes our methods more widely applicable for bioimage annotation tasks. Therefore, the main limitation of our methods lies in the process of image normalization. That is, each image should contain only one embryo scaled to the same size and adjusted to the same orientation, as done in the FlyExpress database where all the embryos are of the same size and aligned with anterior to the left.

So far, many model organism specific *in situ* image databases have been established besides BDGP for mapping

gene expression of different species, such as ANISEED [40], ABA [29], and ZFIN [39]. Like the BDGP database, the ANISEED database collects *in situ* expression images during ascidian (Ciona, Halocynthia, and Phallusia) embryonic development, and organizes these images in groups based on genes and developmental stages with text-based anatomical ontology terms annotated to these image groups. This makes it much easier for researchers to build an annotation system based on our MIML learning methods after normalizing the images. Similarly, the ZFIN database [39] gathers expression images during zebrafish embryogenesis, and also possesses text-based anatomical ontology annotations for images. But, in general, the terms are directly annotated to individual images instead of image groups. Nevertheless, our MIML learning based methods can also be applied under this setting, since the bag of instances can be formed simply from the local patches of individual image. This can be carried out similarly as the cases in our *Drosophila* annotation problem in that each image group contains only one single image. The Allen Brain Atlas (ABA) [29] contains a large scale collection of *in situ* images of (both adult and developing) mouse brains, and possesses a distinct anatomy-based annotation scheme. That is, for each developmental stage, anatomical reference atlases are provided as templates for end users to find out which anatomical structures are matched with the stained image regions. Although this annotation scheme can avoid explicit manual annotation errors to be presented in database, it relies heavily on the ability of individual user to fulfill the image annotation process. Intuitively, it is possible to use our MIML learning based method to accomplish the annotation task if we can collect enough images annotated with text-based anatomical terms to be used as training samples. In general, our work provides an effective framework that can be applied to a wide range of *in situ* images for implementing the computational annotation.

Besides *in situ* gene expression pattern images, there are also large amounts of image data in the fields of biology. For example, the image data on subcellular locations of proteins which describes the spatial distribution of proteins expressed in a given cell type [14]. Similar to gene expression images, the images of protein subcellular location patterns can also be 2D microscopy images photographed for 3D objects by digital cameras. Analogous to our gene expression pattern annotation task, the aim of automated analysis of protein subcellular location images is to identify which organelles show expression of a protein within a cell. During the past decade, methods for automated analysis of protein subcellular location images have been developed based on machine learning techniques and have achieved great successes [14], [20], [34]. Since proteins can display high specialized locations in cells, such as being localized to mitochondrial inner membrane, protein expression patterns of individual organelles can be obtained. Thus, methods [51] for recognizing multiple protein subcellular locations within cells are often established on the assumption that the expression patterns of constituent individual organelles can be acquired in advance. Obviously, these methods could not be applied to our gene expression pattern annotation problem, since genes often express in multiple structures simultaneously instead of individual anatomy, and thus expression patterns of specific individually isolated named anatomical structures are hard to be obtained. By contrast, our MIML learning based methods are designed for addressing a more general learning problem, and thus can be easily applied to the protein subcellular location identification problem for annotating multiple subcellular locations for each image. That is, our MIML learning based methods can provide general solutions for addressing problems of automated analysis of bioimages besides *in situ* gene expression images.

6 Conclusions

In this paper, we address the problem of automated annotation of *Drosophila* gene expression pattern images, which extends our preliminary research [30]. We first show that the underlying nature of the annotation task matches well with a new machine learning framework, MIML learning [56], [57]. Then, we propose two MIML learning algorithms, MIMLSVM$^+$ and E-MIMLSVM$^+$, to deal with the annotation problem. Experiments show that our MIML solutions to the annotation problem can lead to performance superior to state-of-the-art *Drosophila* gene expression pattern annotation method, which demonstrates the effectiveness of the MIML learning framework for annotating bioimages.

The maturing of technologies for high-throughput data production and curation makes genome-wide studies and the integration of genomic data with high volume of various information feasible. Although, large amounts of image data have been produced, automatically interpreting these data is still in its infancy. Effective and efficient approaches are demanded. The work described in this paper provides a new solution to automated understanding of bioimages. Although our work is based on the *Drosophila* embryo image data, it can also be applied to image data of other species or other bioimage-related problems. This makes our methods valuable for the general areas of bioimage informatics [35].

The proposed MIML learning methods achieve promising prediction results, however, there is still much room to improve. This can be achieved either by designing more descriptive features for representing expression patterns or by enhancing the models to utilize training information more effectively, for example, identifying high-order correlations among annotation terms and trying to make use of them. Currently, our work focuses more on the problem of data representation and model construction, and less on the image normalization problem. In the future, we plan to work out computational methods that can implement fully automated analysis of the original expression images.

Acknowledgments

The authors want to thank Dr. Charlotte Konikoff for examining the *in situ* image groups, and the associate editor and anonymous reviewers for helpful comments and suggestions. This research was partially supported by the National Fundamental Research Program of China (2010CB327903), the National Science Foundation of China (60721002, 61073097), the Jiangsu Science Foundation

(BK2008018), the Postdoctoral Science Foundation of China (20090461086), the Jiangsu Postdoctoral Foundation (0802001C), the National Institutes of Health (HG002516), and the US National Science Foundation (IIS-0612069, IIS-0953662).

REFERENCES

[1] B. Bakker and T. Heskes, "Task Clustering and Gating for Bayesian Multitask Learning," *J. Machine Learning Research,* vol. 4, pp. 83-99, 2003.

[2] M.R. Boutell, J. Luo, X. Shen, and C.M. Brown, "Learning Multi-Label Scene Classification," *Pattern Recognition,* vol. 37, no. 9, pp. 1757-1771, 2004.

[3] C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery,* vol. 2, no. 2, pp. 121-167, 1998.

[4] G. Carneiro, A.B. Chan, P.J. Moreno, and N. Vasconcelos, "Supervised Learning of Semantic Classes for Image Annotation and Retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 3, pp. 394-410, Mar. 2007.

[5] C.-C. Chang and C.-J. Lin, "LIBSVM : A Library for Support Vector Machines," http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.

[6] F.R.K. Chung, *Spectral Graph Theory.* Am. Math. Soc. Press, 1997.

[7] C. Cortes and V. Vapnik, "Support Vector Networks," *Machine Learning,* vol. 20, no. 3, pp. 273-297, 1995.

[8] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge Univ. Press, 2000.

[9] S. Daskalaki, I. Kopanas, and N.M. Avouris, "Evaluation of Classifiers for an Uneven Class Distribution Problem," *Applied Artificial Intelligence,* vol. 20, no. 5, pp. 381-417, 2006.

[10] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification,* second ed. John Wiley & Sons, Inc., 2001.

[11] T. Evgeniou, C.A. Micchelli, and M. Pontil, "Learning Multiple Tasks with Kernel Methods," *J. Machine Learning Research,* vol. 6, pp. 615-637, 2005.

[12] T. Evgeniou and M. Pontil, "Regularized Multi-Task Learning," *Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 109-117, 2004.

[13] T. Gärtner, P.A. Flach, A. Kowalczyk, and A.J. Smola, "Multi-Instance Kernels," *Proc. 19th Int'l Conf. Machine Learning,* pp. 179-186, 2002.

[14] E. Glory and R.F. Murphy, "Automated Subcellular Location Determination and High-Throughput Microscopy," *Developmental Cell,* vol. 12, no. 1, pp. 7-14, 2007.

[15] K. Grauman and T. Darrell, "Approximate Correspondences in High Dimensions," *Advances in Neural Information Processing Systems 19,* B. Schölkopf, J. Platt, and T. Hofmann, eds., pp. 505-512, MIT Press, 2007.

[16] M. Gribskov and N.L. Robinson, "Use of Receiver Operating Characteristic (ROC) Analysis to Evaluate Sequence Matching," *Computers and Chemistry,* vol. 20, no. 1, pp. 25-33, 1996.

[17] R. Gurunathan, B.V. Emden, S. Panchanathan, and S. Kumar, "Identifying Spatially Similar Gene Expression Patterns in Early Stage Fruit Fly Embryo Images: Binary Feature versus Invariant Moment Digital Representations," *BMC Bioinformatics,* vol. 5, article no. 202, 2004.

[18] D.R. Hardoon, S. Szedmák, and J. Shawe-Taylor, "Canonical Correlation Analysis: An Overview with Application to Learning Methods," *Neural Computation,* vol. 16, no. 12, pp. 2639-2664, 2004.

[19] D. Haussler, "Convolution Kernels on Discrete Structures," Technical Report UCSC-CRL-99-10, Dept. of Computer Science, Univ. of California at Santa Cruz, Santa Cruz, CA, July 1999.

[20] K. Huang and R.F. Murphy, "From Quantitative Microscopy to Automated Image Understanding," *J. Biomedical Optics,* vol. 9, no. 5, pp. 893-912, 2004.

[21] K.S. Imai, K. Hino, K. Yagi, N. Satoh, and Y. Satou, "Gene Expression Profiles of Transcription Factors and Signaling Molecules in the Ascidian Embryos: Towards a Comprehensive Understanding of Gene Networks," *Development,* vol. 131, no. 16, pp. 4047-4058, 2004.

[22] S. Ji, Y.-X. Li, Z.-H. Zhou, S. Kumar, and J. Ye, "A Bag-of-Words Approach for *Drosophila* Gene Expression Pattern Annotation," *BMC Bioinformatics,* vol. 10, article no. 119, 2009.

[23] S. Ji, L. Sun, R. Jin, S. Kumar, and J. Ye, "Automated Annotation of *Drosophila* Gene Expression Patterns Using a Controlled Vocabulary," *Bioinformatics,* vol. 24, no. 17, pp. 1881-1888, 2008.

[24] S. Ji, L. Yuan, Y.-X. Li, Z.-H. Zhou, S. Kumar, and J. Ye, "*Drosophila* Gene Expression Pattern Annotation Using Sparse Features and Term-Term Interactions," *Proc. 15th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 407-416, 2009.

[25] Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towards Optimal Bag-of-Features for Object Categorization and Semantic Video Retrieval," *Proc. Sixth ACM Int'l Conf. Image and Video Retrieval,* pp. 494-501, 2007.

[26] T. Joachims, "Making Large-Scale SVM Learning Practical," *Advances in Kernel Methods—Support Vector Learning,* B. Schölkopf, C.J.C. Burges, and A.J. Smola, eds., pp. 41-56, MIT Press, 1998.

[27] S. Kumar, K. Jayaramanc, S. Panchanathan, R. Gurunatha, A. Marti-Subirana, and S.J. Newfeld, "Best: A Novel Computational Approach for Comparing Gene Expression Patterns from Early Stages of Drosophlia Melanogaster Development," *Genetics,* vol. 162, no. 4, pp. 2037-2047, 2002.

[28] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 2169-2178, 2006.

[29] E.S. Lein et al, "Genome-Wide Atlas of Gene Expression in the Adult Mouse Brain," *Nature,* vol. 445, no. 7124, pp. 168-176, 2006.

[30] Y.-X. Li, S. Ji, S. Kumar, J. Ye, and Z.-H. Zhou, "*Drosophila* Gene Expression Pattern Annotation Through Multi-Instance Multi-Label Learning," *Proc. 21st Int'l Joint Conf. Artificial Intelligence,* pp. 1445-1450, 2009.

[31] Y. Liu, R. Jin, and L. Yang, "Semi-Supervised Multi-Label Learning by Constrained Non-Negative Matrix Factorization," *Proc. 21st Nat'l Conf. Artificial Intelligence,* pp. 421-426, 2006.

[32] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[33] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 10, pp. 1615-1630, Oct. 2005.

[34] R.F. Murphy, M. Velliste, and G. Porreca, "Robust Numerical Features for Description and Classification of Subcellular Location Patterns in Fluorescence Microscope," *J. Very Large Scale Integration Signal Processing,* vol. 35, no. 3, pp. 311-321, 2003.

[35] H. Peng, "Bioimage Informatics: A New Area of Engineering Biology," *Bioinformatics,* vol. 24, no. 17, pp. 1827-1836, 2008.

[36] F. Provost and T. Fawcett, "Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions," *Proc. Third ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 43-48, 1997.

[37] *Drosophila: A Practical Approach,* D.B. Roberts, ed. Oxford IRL Press, 1998.

[38] J. Sivic and A. Zisserman, "Efficient Visual Search of Videos Cast as Text Retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 4, pp. 591-606, Apr. 2009.

[39] J. Sprague et al., "The Zebrafish Information Network: The Zebrafish Model Organism Database," *Nucleic Acids Research,* vol. 34, suppl 1, pp. D581-D585, 2006.

[40] O. Tassy et al., "The ANISEED Database: Digital Representation, Formalization, and Elucidation of a Chordate Developmental Program," *Genome Research,* vol. 20, no. 10, pp. 1459-1468, 2010.

[41] P. Tomancak, B.P. Berman, A. Beaton, R. Weiszmann, E. Kwan, V. Hartenstein, S.E. Celniker, and G.M. Rubin, "Global Analysis of Patterns of Gene Expression During *Drosophila* Embryogenesis," *Genome Biology,* vol. 8, no. 7, p. R145, 2007.

[42] P. Tomancak et al., "Systematic Determination of Patterns of Gene Expression during *Drosophila* Embryogenesis," *Genome Biology,* vol. 3, no. 12, p. R088, 2002.

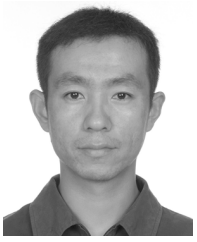
[43] N. Ueda and K. Saito, "Parametric Mixture Models for Multi-Labeled Text," *Advances in Neural Information Processing Systems 15,* S. Becker, S. Thrun, and K. Obermayer, eds., pp. 721-728, MIT Press, 2003.

[44] V. Vapnik, *The Nature of Statistical Learning Theory.* Springer-Verlag, 1995.

[45] G.M. Weiss, "Mining with Rarity: A Unifying Framework," *SIGKDD Explorations,* vol. 6, no. 1, pp. 7-19, 2004.

[46] J. Yang, Y.-G. Jiang, A.G. Hauptmann, and C.-W. Ngo, "Evaluating Bag-of-Visual-Words Representations in Scene Classification," *Proc. Ninth ACM SIGMM Int'l Workshop Multimedia Information Retrieval,* pp. 197-206, 2007.

[47] J. Zhang, Z. Ghahramani, and Y. Yang, "Flexible Latent Variable Models for Multi-Task Learning," *Machine Learning,* vol. 73, no. 3, pp. 221-242, 2008.
[48] M.-L. Zhang and Z.-H. Zhou, "ML-kNN: A Lazy Learning Approach to Multi-Label Learning," *Pattern Recognition,* vol. 40, no. 7, pp. 2038-2048, 2007.
[49] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding Bag-of-Words Model: A Statistical Framework," *Int'l J. Machine Learning and Cybernetics,* vol. 1, no. 1, pp. 43-52, 2010.
[50] Y. Zhang and Z.-H. Zhou, "Multi-Label Dimensionality Reduction via Dependency Maximization," *ACM Trans. Knowledge Discovery from Data,* vol. 4, no. 3, article no. 14, Oct. 2010.
[51] T. Zhao, M. Velliste, M.V. Boland, and R.F. Murphy, "Object Type Recognition for Automated Analysis of Protein Subcellular Location," *IEEE Trans. Image Processing,* vol. 14, no. 9, pp. 1351-1359, Sept. 2005.
[52] J. Zhou and H. Peng, "Automatic Recognition and Annotation of Gene Expression Patterns of Fly Embryos," *Bioinformatics,* vol. 23, no. 5, pp. 589-596, 2007.
[53] Z.-H. Zhou and X.-Y. Liu, "Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem," *IEEE Trans. Knowledge and Data Eng.,* vol. 18, no. 1, pp. 63-77, Jan. 2006.
[54] Z.-H. Zhou and X.-Y. Liu, "On Multi-Class Cost-Sensitive Learning," *Computational Intelligence,* vol. 26, no. 3, pp. 232-257, 2010.
[55] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li, "Multi-Instance Learning by Treating Instances as Non-i.i.d. Samples," *Proc. 26th Int'l Conf. Machine Learning,* pp. 1249-1256, 2009.
[56] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li, "MIML: A Framework for Learning with Ambiguous Objects," *Computer Research Repository (CoRR),* vol. abs/0808.3231, 2008.
[57] Z.-H. Zhou and M.-L. Zhang, "Multi-Instance Multi-Label Learning with Application to Scene Classification," *Advances in Neural Information Processing Systems 19,* B. Schölkopf, J. Platt, and T. Hofmann, eds., pp. 1609-1616, MIT Press, 2007.



Ying-Xin Li received the PhD degree in pattern recognition and intelligent systems from Beijing University of Technology, China, in 2006. He joined the LAMDA group as a postdoctoral fellow in 2008. He joined the Beijing Jingwei Textile Machinery New Technology Co., Ltd. in 2010 and is currently the leader of the Institute of Machine Vision and Machine Intelligence. His research interests include machine learning, data mining, bioinformatics, and machine vision.

Shuiwang Ji received the PhD degree in computer science from Arizona State University in 2010. He is currently an assistant professor in the Department of Computer Science, Old Dominion University, Norfolk, Virginia. He is the receipt of the Outstanding PhD Student Award in Computer Science at Arizona State University in 2010. He was one of the designers of the human action recognition system that achieved the best performance on three tasks in the TRECVID video surveillance evaluation in 2009. He has served on the program committees of International Conference on Machine Learning, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, and International Conference on Acoustics, Speech, and Signal Processing. His research interests are computational biology, machine learning, data mining, and computer vision.

Sudhir Kumar received the bachelor's degree in electrical and electronics engineering and the masters' degree in biological sciences from the Birla Institute of Technology and Science (BITS) at Pilani in India. He is a professor of life sciences and director of the Center for Evolutionary Medicine and Informatics, Biodesign Institute, at Arizona State University. His doctoral work in genetics and postdoctoral research in molecular evolution was conducted at Pennsylvania State University. He is an interdisciplinary scientist bringing the problem-solving skills from his engineering background and his knowledge of evolutionary genetics to tackle long-standing problems in functional genomics, medicine, and evolutionary biology.

Jieping Ye received the PhD degree in computer science from the University of Minnesota, Twin Cities, in 2005. He is an associate professor in the Department of Computer Science and Engineering at Arizona State University. His research interests include machine learning, data mining, and biomedical informatics. He won the outstanding student paper award at ICML in 2004, the SCI Young Investigator of the Year Award at ASU in 2007, the SCI Researcher of the Year Award at ASU in 2009, the US National Science Foundation (NSF) CAREER Award in 2010, and the KDD best research paper award honorable mention in 2010. He is a member of the IEEE.

Zhi-Hua Zhou (S'00-M'01-SM'06) received the BSc, MSc, and PhD degrees in computer science from Nanjing University, China, in 1996, 1998, and 2000, respectively, all with the highest honors. He joined the Department of Computer Science and Technology at Nanjing University as an assistant professor in 2001, and is currently professor and director of the LAMDA group. His research interests are in artificial intelligence, machine learning, data mining, pattern recognition and image retrieval. In these areas he has published more than 80 papers in leading international journals or conference proceedings, and holds 11 patents. He has won various awards/honors including the National Science & Technology Award for Young Scholars of China, the Fok Ying Tung Young Professorship 1st-Grade Award, the Microsoft Young Professorship Award, PAKDD2006 Data Mining Competition Grand Champion, and a number of international journals/conferences paper awards. He is an associate editor-in-chief of the *Chinese Science Bulletin*, associate editor of the *IEEE Transactions on Knowledge and Data Engineering* and *ACM Transactions on Intelligent Systems and Technology*, and on the editorial boards of various other journals. He is the founding steering committee cochair of ACML, a steering committee member of PAKDD and PRICAI, program committee chair/cochair of PAKDD07, PRICAI08 and ACML09, vice chair, or area chair or senior PC of conferences including IEEE ICDM06, IEEE ICDM08, SIAM DM09, ACM CIKM09, ACM SIGKDD10, ECML PKDD10, ICPR10, AAAI'11, IJCAI'11, ACM SIGKDD'11, etc., and general chair/cochair or program committee chair/cochair of various native conferences in China. He is the chair of the Machine Learning Technical Committee of the Chinese Association of Artificial Intelligence, vice chair of the Artificial Intelligence & Pattern Recognition Technical Committee of the China Computer Federation, and the chair of the IEEE Computer Society Nanjing Chapter. He is a senior member of the ACM and the IEEE.