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

Ying-Xin Li[1] and **Shuiwang Ji**[2] and **Sudhir Kumar**[2] and **Jieping Ye**[2] and **Zhi-Hua Zhou**[1*]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Center for Evolutionary Functional Genomics, Arizona State University, Temple, AZ, USA
{liyx, zhouzh}@lamda.nju.edu.cn {shuiwang.ji, jieping.ye, s.kumar}@asu.edu

Abstract

The Berkeley *Drosophila* Genome Project (BDGP) has produced a large number of gene expression patterns, many of which have been annotated textually with anatomical and developmental terms. These terms spatially correspond to local regions of the images; however, they are attached collectively to groups of images, such that it is unknown which term is assigned to which region of which image in the group. This poses a challenge to the development of the computational method to automate the textual description of expression patterns contained in each image. In this paper, we show that the underlying nature of this task matches well with a new machine learning framework, Multi-Instance Multi-Label learning (MIML). We propose a new MIML support vector machine to solve the problems that beset the annotation task. Empirical study shows that the proposed method outperforms the state-of-the-art *Drosophila* gene expression pattern annotation methods.

1 Introduction

Widely studied in developmental biology, the fruit fly *Drosophila* melanogaster is one of the most well-known model organisms used in scientific research. The Berkeley *Drosophila* Genome Project (BDGP) aims to make extensive studies on the genomics of *Drosophila* melanogaster, and it has produced a comprehensive atlas of spatial patterns of gene expressions during *Drosophila* embryogenesis using high-throughput RNA *in situ* hybridization techniques. These spatial-temporal gene expression pattern data are documented in the form of a large number of digital images of individual embryos [Tomancak *et al.*, 2002]. In addition, anatomical and developmental ontology terms in a controlled vocabulary (CV) are assigned to these images (e.g., Figure 1) to facilitate the search and comparison of gene expression patterns during *Drosophila* embryogenesis. This annotation is of great importance in the study of developmental biology, as it provides a direct way to reveal the interactions and biological functions of genes based on gene expressions, thus shedding light on the research of gene regulatory networks. The annotation work, however, is currently conducted manually by human curators. With the swift and escalating procurement of more and more images, manual annotation becomes increasingly infeasible, and it is now highly desirable and even necessary to automatically annotate the gene expression patterns [Zhou and Peng, 2007; Ji *et al.*, 2008].

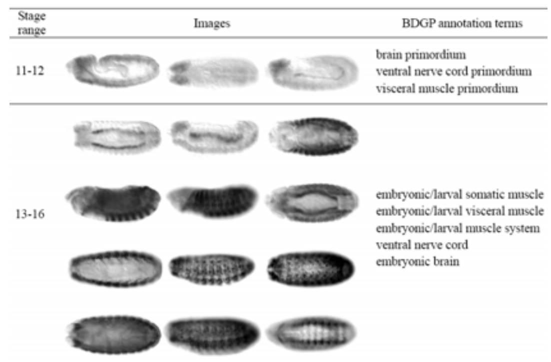


Figure 1: Samples of images and associated annotation terms of the gene *Actn* in the stage ranges 11-12 and 13-16 in the BDGP database. The darkly stained region highlights the place where the gene is expressed. The darker the region, the higher the gene expression.

Nevertheless, a significant challenge awaits those who aspire to develop computational methods to automate the annotation of gene expression images during *Drosophila* embryogenesis. That is, the gene expression pattern of a specific anatomical and developmental ontology term is body-part related and presents in local regions of images, while in the BDGP, the terms are attached collectively to groups of images with the identity and precise placement of the term remaining a mystery. As shown in Figure 1, each image panel is assigned a group of annotation terms, but this does not mean

*This research was partially supported by NSFC (60635030, 60721002), 863 Program (2007AA01Z169), JiangsuSF (BK2008-018), Jiangsu Postdoc Fund (0802001C), NIH (HG002516) and NSF (IIS-0612069).

that all the annotations apply to every image in an image group, nor does it mean that the terms must appear together for a specific image.

In fact, such situations often occur in bio-research and are not at all uncommon. For instance, protein molecules can possess different conformations and exhibit varying biochemical functions. Predicting biochemical functions of molecules with a collection of various conformations remains a crucial task in biochemical and pharmaceutical studies, a fact that burdens the researcher in the scenario of currently lacking knowledge of which conformation is responsible for a specific function. Therefore, a good solution to the problems inherent in the *Drosophila* gene expression annotation task may also illustrate a promising remedy for other bio-problems with similar underlying difficulties.

In this paper, we disclose that the underlying nature of the *Drosophila* gene expression pattern annotation problem matches well with a recent machine learning framework, i.e., Multi-Instance Multi-Label learning (MIML) [Zhou and Zhang, 2007]. We propose a new MIML support vector machine algorithm, MIMLSVM$^+$, and our empirical study on the BDGP database shows that its performance is superior to the state-of-the-art *Drosophila* gene expression pattern annotation methods.

The rest of the paper is organized as follows. Section 2 briefly reviews some related work. Section 3 shows the relation between the annotation problem and the MIML learning framework, and presents the MIMLSVM$^+$ algorithm. Section 4 reports on experimental results. Section 5 concludes.

2 Related Work

The *Drosophila* gene expression pattern annotation problem can be traced back to efforts to construct computational approaches for the comparison of spatial expression patterns between two genes. To automate the comparison process, an algorithm called BESTi [Kumar *et al.*, 2002] was proposed. Each image was represented by a binary feature vector (BFV), and the distance between two BFVs was used to measure the similarity between the expression patterns of two images. The BESTi algorithm was further improved by Gurunathan *et al.* [2004]. These studies used images collected from published literatures, which often exhibited large variations. BDGP produces a large number of gene expression images under the same experimental conditions, thus providing high-quality data for further study; i.e., annotating body-part structures to gene expression patterns.

There are only a few published works on the *Drosophila* gene expression pattern annotation task. Zhou and Peng [2007] represented each image with multi-resolution 2D wavelet features, and the annotation problem was decomposed into a series of binary classification tasks each for a term; the linear discriminant analysis algorithm was employed to construct the binary classifiers for annotation. Ji *et al.* [2008] proposed a multi-kernel learning method for the *Drosophila* gene expression pattern annotation problem. They extracted local descriptors before calculating pyramid match kernels on different descriptors. These kernels were then combined using a hypergraph learning method to build a classifier for annotation. Both Zhou and Peng [2007] and Ji *et al.* [2008] constructed annotation systems under the conventional supervised learning framework. The main difference is that Ji *et al.* [2008] worked in the setting described in Section 1, that is, it is not known which term was assigned to which region of which image in a image group, while Zhou and Peng [2007] worked in the setting in which the relation between the terms and the images was assumed to be known.

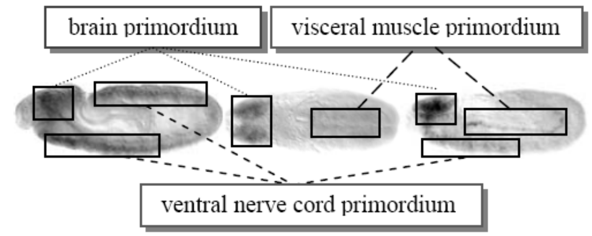


Figure 2: Illustration of the underlying relationships between the annotation terms and their corresponding local expression patterns. The image panel of gene *Actn* in the stage range 11-12 in Figure 1 is presented here.

3 The Proposed Method

3.1 Formulation as a MIML Problem

The actual *Drosophila* embryos are 3D objects. However, in the BDGP, they were documented as 2D images with different views (lateral, dorsal, ventral and intermediate view) of embryos taken to capture the genes' complete expression patterns. These images were organized as an image panel, and the CV terms representing anatomical and developmental ontology structures were annotated by human curators if the gene showed expression in these structures. Thus, images in the same group could be taken from different embryos describing the expression patterns of a specific gene, or taken from different views of a specific embryo. This leads to the facts that: (1) some embryonic structures can only be captured by one of the images in the panel, and (2) images in a panel taken from different embryos share some anatomical structures with variations in shape and position due to genetic variations and limitations of image processing techniques. Furthermore, the anatomical terms are body-part related, and the corresponding expression pattern of a specific term only presents in some local regions of images in the image panel as illustrated in Figure 2. Therefore, automatic annotation of anatomical terms is challenging since it is unclear which term is assigned to which region of which image in the group, as mentioned above.

Formally, let B_i denote the image panel of the i-th gene; let P_{iu} $(u = 1, 2, \cdots, u_i)$ denote the u-th image of B_i, and let x_{iuv} $(v = 1, 2, \cdots, n_{iu})$ denote the local features of the expression pattern of the v-th patch (local region) extracted from the image P_{iu}. For convenience, we use $X_i = \{x_t\}$ to represent the collection of all the local feature vectors of B_i, and Y_i are the terms assigned to B_i. From the machine learning view, X_i is a *bag* containing instances $\{x_t\}$, and Y_i is the label set of X_i. Thus, the annotation task can be viewed as a problem of predicting proper labels Y^* of a test bag

X^* given a training set $\{(X_i, Y_i)\}$ $(i = 1, 2, \cdots, n)$. However, this learning problem is dramatically different from the conventional supervised learning method that learns concepts from objects represented by a single instance associated with a single label, since there is no explicit relationship between a local feature vector x_t and a label $y_{ij} \in Y_i$. The only information provided by the training object (X_i, Y_i) is that for any label $y_{ij} \in Y_i$, there must exist at least one instance $x_t \in X_i$ responsible for the label y_{ij}.

It is interesting that the above problem falls exactly into the Multi-Instance Multi-Label (MIML) learning framework that was proposed recently [Zhou and Zhang, 2007]. 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 $\{(X_i, Y_i)\}$, realizing a 'many-to-many' mapping between instances and class labels. The MIML framework has been found to be helpful in tasks involving ambiguous objects [Zhou and Zhang, 2007]. As shown in Figure 2, it is evident that our concerned *Drosophila* gene expression pattern annotation problem matches well with the MIML learning framework. Here, we regard each image panel as an object (a bag) that is described by many local feature vectors (multi-instances) and labeled with a group of terms (multi-labels). Therefore, it is natural to address this problem within the MIML learning framework.

3.2 The MIMLSVM$^+$ Algorithm

Zhou and Zhang [2007] have proposed two MIML algorithms: MIMLBoost and MIMLSVM. It has been shown that although these two algorithms work by degenerating MIML problems to solve either multi-instance single-label problems or single-instance multi-label problems, the MIML algorithms still achieved better performance than conventional supervised learning methods [Zhou and Zhang, 2007]. However, neither algorithm has been designed for large-scale problems, while our concerned *Drosophila* gene expression pattern annotation problem involves a vast database. Therefore, new algorithms for efficiently addressing MIML learning problems are desired.

In this paper, we present a new MIML support vector machine algorithm and show how we applied it to the task of *Drosophila* gene expression pattern annotation. In contrast to the working routine of MIMLSVM, which degenerates MIML problems to single-instance multi-label problems, the proposed algorithm works by degenerating MIML problems to multi-instance single-label problems for addressing MIML problems. To distinguish it from MIMLSVM, the proposed algorithm is denoted as MIMLSVM$^+$.

Veering from the MIMLBoost that degenerates MIML problems to multi-instance single-label problems by adding pseudo-labels to each instance, here, we take a direct approach. That is, each time we train a classifier for a label; we collect all the bags with this label as positive bags, and bags without the label as negative ones. Thus, we get a series of binary classification tasks, each tackled by a support vector machine. Since the negative class is obtained by merging all the bags without the concerned label, the number of negative bags can be much larger than that of positive bags. To deal with this class-imbalance, different penalty parameters can be used for positive and negative relaxation terms respectively [Osuna *et al.*, 1997].

Formally, for each label $y \in \mathcal{Y}$, let $\varphi(X_i, y) = +1$ if $y \in Y_i$ and -1 otherwise. Then the formulation of the corresponding SVM is as follows:

$$\begin{aligned} \min_{w,b,\xi} \quad & \tfrac{1}{2}\|w\|^2 + C^+ \textstyle\sum_{\varphi(X_i,y)=1} \xi_i + C^- \textstyle\sum_{\varphi(X_i,y)=-1} \xi_i \\ \text{subject to:} \quad & \varphi(X_i, y)(w^{'}\phi(X_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \qquad\qquad (i = 1, 2, \cdots, n) \end{aligned}$$

where $\phi(X_i)$ is the mapping function that maps bag of instances X_i to a kernel space; $\varphi(X_i, y)$ indicates whether y is a proper label of X_i; ξ_i is the hinge loss; n is the number of image panels in the training set; and w and b are parameters for representing a linear discrimination function in the kernel space. C^+ and C^- are the penalty parameters for errors resulting from positive bags and negative bags, respectively. We choose $C^+ > C^-$ to make the classifier biased toward positive bags.

One well-known kernel for representing spaces that are not mere attribute-value vectors is the convolution kernel [Haussler, 1999]. Based on the convolution kernel, the standard set kernel over sets $X = \{x_1, x_2, \cdots, x_n\}$ and $X^{'} = \{x_1^{'}, x_2^{'}, \cdots, x_m^{'}\}$ can be defined as:

$$K_{SET}(X, X^{'}) = \textstyle\sum_{i=1}^{n} \sum_{j=1}^{m} K(x_i, x_j^{'}) \ ,$$

where $K(\cdot, \cdot)$ is some instance-level kernel. For separating multi-instance bags, the standard set kernel $K_{set}(X, X^{'})$ is modified by exponentiating $K(\cdot, \cdot)$ by a power to multi-instance kernel $K_{MI}(X, X^{'})$, and it can be proved theoretically that this kind of kernel can separate multi-instance concepts with a proper value of p [Gärtner *et al.*, 2002]. The multi-instance kernel is defined as follows, where $p \geq 1$.

$$K_{MI}(X, X^{'}) = \textstyle\sum_{i=1}^{n} \sum_{j=1}^{m} K^p(x_i x_j^{'}),$$

As for the *Drosophila* gene expression pattern annotation problem, both the local visual features and the spatial information are important for describing the expression pattern of a patch. This is because (1) the gene expressed in different embryonic structures may result in similar local visual features, but it may be presented in different local regions of the embryo. This validates the importance of exploiting spatial information in the analysis of gene expression patterns; and (2) generally, the gene expressed in different embryonic structures can lead to different visual features, and this validates the necessity of using visual characteristics of expression patterns. These two facts are also the key reasons for utilizing the RNA *in situ* technique instead of the DNA microarray in the study of gene expression patterns during embryogenesis, since the DNA microarray is commonly used to measure the averaged gene expression levels.

Therefore, we use $X_i = \{x_t\} = \{(x_{t0}, x_{t1})\}$ to represent the collection of local feature vectors of an image panel B_i, where x_{t0} and x_{t1} denote the visual feature vector and the spatial information respectively, characterizing the expression patterns of patch t. For an efficient combination of visual information with spatial information, we define the multi-instance kernel as:

$$K_{MI_D}(X_i, X_j) = \sum_{(x_{t0},x_{t1})\in X_i} \sum_{(x_{k0},x_{k1})\in X_j} e^{-\gamma_1\|x_{t0}-x_{k0}\|^2-\gamma_2\|x_{t1}-x_{k1}\|^2}$$

The instance-level kernel used in K_{MI_D} is: $K(x_t, x_k) = e^{-\gamma_1\|x_{t0}-x_{k0}\|^2-\gamma_2\|x_{t1}-x_{k1}\|^2}$. Intuitively, the first term $\|x_{t0}-x_{k0}\|^2$ of the exponent measures the similarity of visual features between the expression patterns of two patches; the second part $\|x_{t1}-x_{k1}\|^2$ of the exponent calculates the spatial distance between two patches. Thus, the visual information and the spatial information are combined directly with different weights γ_1 and γ_2 through the kernel trick. This strategy can also be seen as a preliminary attempt to capture the structure information among instances of bags. It is easy to check that $K(x_t, x_k)$ is a valid kernel, because only the dot product of two Gaussian kernels is presented in $K(x_t, x_k)$. Clearly, the contributions of visual information and spatial information for classification can be balanced by tuning the parameters γ_1 and γ_2. There is no explicit exponential parameter p presented in K_{MI_D}, since the parameter p can be chosen implicitly when choosing the parameters γ_1 and γ_2. Note that for the Gaussian RBF kernel, $K^p(\cdot,\cdot)$ is the Gaussian RBF kernel [Gärtner *et al.*, 2002].

The resulting classifier can be used directly to classify bags of instances. The discriminant function is

$$h_y(X^*) = \sum_{i=1}^{\#sv} \alpha_i \varphi(X_i, y) K_{MI_D}(X_i, X^*) + b$$

where $\#sv$ is the number of support vectors; α_i is the parameter learned from the dual form of the SVM formulation described above.

In the testing phase, the T-criterion [Boutell *et al.*, 2004] is used as in the original MIMLSVM. That is, a test bag is labeled by all the class labels with positive SVM scores, or by the class label with the top score when all the SVM scores are negative. Once a MIML training set is presented, the multi-instance kernel matrix $[K_{MI_D}(X_i, X_j)]$ $(i, j = 1, 2, ...n)$ can be calculated with the training bags and then used directly for training classifiers. The pseudo-code for MIMLSVM$^+$ is described in Table 1.

Table 1: The MIMLSVM$^+$ algorithm

1. For training set $\{(X_i, Y_i)\}(i = 1, \cdots, n)$, calculate multi-instance kernel matrix $[K_{MI_D}(X_i, X_j)]$ $(i, j = 1, \cdots, n)$.
2. For each label $y \in \mathcal{Y}$, derive dataset $D_y = \{(X_i, \varphi(X_i, y))\}$ $(i = 1, \cdots, n)$, and then train an SVM h_y based on $[K_{MI_D}$ $(X_i, X_j)]$: h_y=SVMTrain(D_y).
3. The annotation for test bag X^* is obtained by:
$$Y^* = \{\arg\max_{y\in\mathcal{Y}} h_y(X^*) | h_y(X^*) < 0, \forall y \in \mathcal{Y}\} \bigcup \{y | h_y(X^*) \geq 0, y \in \mathcal{Y}\}$$

4 Empirical Study

4.1 Configuration

We evaluated the performance of our proposed method on a data set containing 119 terms and 15,434 images representing the expression atlas of a total of 2,816 genes. These images were obtained from the FlyExpress repository (http://www.flyexpress.net) that collects images generated from the BDGP study. All the images have already been well-aligned with the anterior to the left, and standardized to the size of 320 × 128 pixels. On each image, dense local features were extracted on regular patches, which is widely used for aligned images. We used the SIFT descriptor [Lowe, 2004], a very popular local descriptor used in the field of computer vision [Mikolajczyk and Schmid, 2005], calculated on each patch to generate visual features of the corresponding gene expression patterns. The radius and spacing of the regular patches are all set to 16 pixels. Consequently, there are a total of 133 local regions cropped from each image. The coordinates of the center point of each local region were employed to represent the corresponding spatial information.

We compare MIMLSVM$^+$ with the multi-pyramid match kernel learning method (abbreviated as 'MKL-PMK') [Ji *et al.*, 2008] in our experiments. The MKL-PMK method currently achieves the best performance in solving the *Drosophila* gene expression pattern annotation problem. Another method, i.e., Zhou and Peng [2007]'s method, is not included in our empirical study because it requires embryo images to be annotated individually in the training set, which differs from our task.

To study the effectiveness of exploiting spatial information in the annotation task, we also evaluate the performance of two degenerated variants of MIMLSVM$^+$. The first is MIMLSVM$^+_{SV}$, which works with $K(x_t, x_k) = e^{-\gamma_1\|(x_{t0},x_{t1})-(x_{k0},x_{k1})\|^2}$. In other words, MIMLSVM$^+_{SV}$ uses both the spatial information and visual information, but it merges them into a single feature vector (x_{t0}, x_{t1}). The second variant is MIMLSVM^+_V, which works with $K(x_t, x_k) = e^{-\gamma_1\|x_{t0}-x_{k0}\|^2}$. That is, MIMLSVM$^+_V$ does not use spatial information.

The original MIMLSVM algorithm [Zhou and Zhang, 2007] employs a clustering process to transform the MIML task into a single-instance multi-label problem. It is quite slow when dealing with large-scale problems, and we find that it could not fulfill our annotation task within a reasonable timeframe. Therefore, we randomly sampled a small data set of 10 terms with 167 image groups to compare the performance of the original MIMLSVM against MIMLSVM$^+$. For MIMLSVM, the spatial information was combined with visual features by adding the region coordinates as two additional dimensions to each SIFT descriptor. For reference, we also reported the results of MIMLSVM$^+_{SV}$ on the small sampled data set, since both MIMLSVM and MIMLSVM$^+_{SV}$ utilize the spatial information in the same way.

In each experiment, the whole data set is randomly partitioned into a training set and a test set using a ratio of 1:1. The training set is used to build classifiers, and the test set is used to evaluate the annotation performance. Each experiment is repeated with random training/test splits for 30 times on the full data set and 20 times on the small sampled subset. All the model parameters are tuned with cross validation on training sets. For the MIMLSVM$^+$ series algorithms, γ_1

Table 2: Comparisons of annotation performance (mean±std.). The best performance of each criterion is highlighted with boldface. 'Ave. Precision' denotes average precision; 'Rankloss' denotes ranking loss, and 'Hammloss' represents hamming loss. ↓ indicates 'the smaller, the better'; ↑ denotes 'the larger, the better'. '10S' indicates the small sampled data set.

# terms	# groups	Algorithms	macro-F1 ↑	micro-F1 ↑	AUC ↑	Ave. Precision ↑	one-error ↓	coverage ↓	Rankloss ↓	Hammloss ↓
10	2228	MIMLSVM^+	**0.643±0.011**	**0.689±0.007**	**0.883±0.004**	**0.779±0.005**	**0.272±0.008**	**2.994±0.056**	**0.150±0.006**	**0.150±0.004**
		MIMLSVM^+_{SV}	0.627±0.010	0.676±0.006	0.869±0.004	0.773±0.005	0.277±0.011	3.073±0.048	0.157±0.004	0.156±0.003
		MIMLSVM^+_V	0.619±0.011	0.667±0.007	0.863±0.004	0.764±0.005	0.291±0.009	3.139±0.044	0.164±0.004	0.160±0.003
		MKL-PMK	0.584±0.009	0.621±0.009	0.825±0.006	0.722±0.007	0.343±0.011	3.483±0.072	0.198±0.006	0.196±0.006
20	2476	MIMLSVM^+	**0.468±0.015**	**0.587±0.007**	**0.862±0.003**	**0.673±0.008**	**0.357±0.011**	**6.189±0.117**	**0.152±0.005**	**0.114±0.002**
		MIMLSVM^+_{SV}	0.454±0.012	0.574±0.008	0.845±0.003	0.660±0.009	0.364±0.013	6.481±0.119	0.163±0.005	0.118±0.003
		MIMLSVM^+_V	0.445±0.012	0.566±0.006	0.840±0.004	0.651±0.008	0.377±0.011	6.609±0.114	0.169±0.004	0.119±0.002
		MKL-PMK	0.410±0.007	0.506±0.006	0.771±0.006	0.580±0.007	0.445±0.009	8.082±0.122	0.230±0.005	0.144±0.003
30	2646	MIMLSVM^+	**0.368±0.012**	**0.541±0.007**	**0.850±0.003**	**0.623±0.007**	**0.377±0.010**	**9.406±0.173**	**0.153±0.003**	**0.087±0.002**
		MIMLSVM^+_{SV}	0.354±0.001	0.527±0.006	0.829±0.004	0.605±0.007	0.388±0.010	9.964±0.195	0.166±0.004	0.090±0.002
		MIMLSVM^+_V	0.340±0.012	0.517±0.007	0.822±0.004	0.596±0.007	0.399±0.010	10.183±0.189	0.171±0.004	0.091±0.002
		MKL-PMK	0.310±0.008	0.455±0.008	0.741±0.007	0.511±0.008	0.488±0.011	13.010±0.2413	0.243±0.006	0.142±0.003
10S	167	MIMLSVM^+	**0.460±0.041**	**0.606±0.026**	**0.807±0.191**	**0.733±0.019**	**0.311±0.034**	**3.508±0.262**	**0.186±0.015**	**0.171±0.019**
		MIMLSVM^+_{SV}	0.424±0.049	0.569±0.033	0.774±0.017	0.710±0.027	0.354±0.047	3.667±0.199	0.204±0.016	0.191±0.015
		MIMLSVM	0.176±0.047	0.367±0.054	0.629±0.041	0.592±0.028	0.468±0.060	4.792±0.300	0.318±0.029	0.241±0.097

and γ_2 can be set as suggested in [Gärtner *et al.*, 2002]; i.e., the parameters γ_1 and γ_2 should be in the order of magnitude of $1/(2d_1^2)$ and $1/(2d_2^2)$ or lower respectively, where d_1 and d_2 are the dimensions of the SIFT descriptor and that of the region coordinates, respectively. Therefore, we simply set $\gamma_1 = 10^{-5}$ and $\gamma_2 = 10^{-2}$ for all the labels in our experiments on the full data set to avoid time-consuming cross validations. To avoid numerical problems, $K_{MI_D}(X_i, X_j)$ is normalized with the (i, j)th term divided by $\sqrt{N_i}\sqrt{N_j}$, where N_i and N_j are the numbers of instances in the bags X_i and X_j respectively. For the MKL-PMK method, three different kernel combination schemes (star, clique and kernel canonical correlation analysis) were employed, and they produced three sets of annotation results. For each criterion, only the best among these three schemes was reported as the performance of MKL-PMK.

4.2 Results

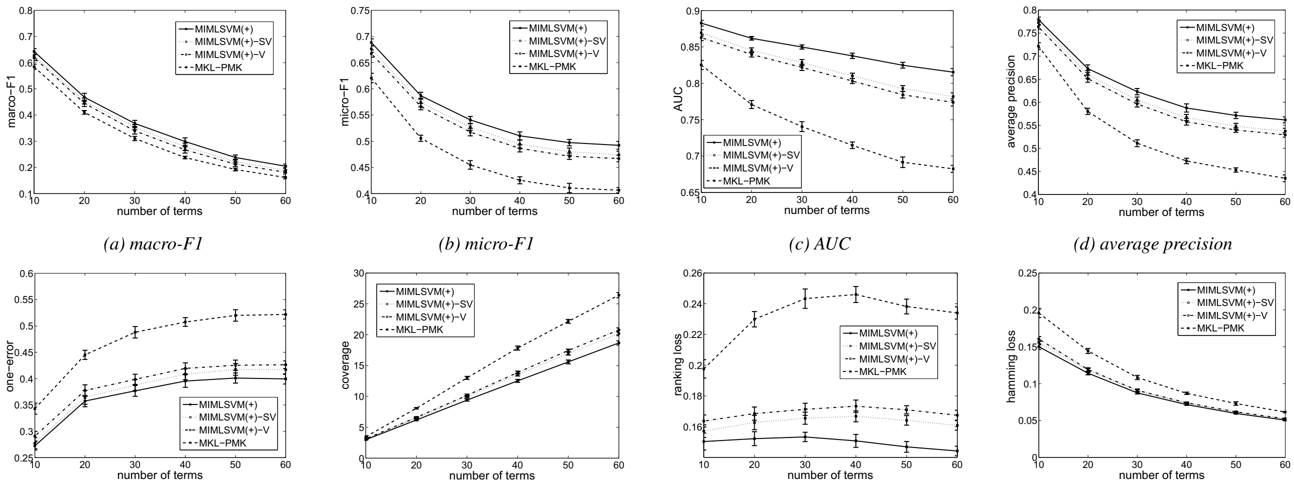
We evaluated the annotation performance in terms of eight criteria. The first three criteria, macro-F1, micro-F1 and AUC (area under ROC curve), have been used in the evaluation of the annotation performance [Ji *et al.*, 2008]. Macro-F1 is the averaged F1 value across all the labels, while micro-F1 is the F1 value calculated from the sum of per-label contingency tables. The AUC criteria used for the annotation task is the averaged AUC value across all the labels. The larger the values of these measures, the better the performance.

The other five criteria – average precision, one-error, coverage, ranking loss and hamming loss – have been popularly used in multi-label learning and MIML [Schapire and Singer, 2000; Zhou and Zhang, 2007]. Briefly speaking, average precision evaluates the average fraction of labels ranked above a particular label; one-error measures how many times the top-ranked label is not a proper label of an object; coverage reflects how far it is needed, on the average, to go down the list of labels to cover all the proper labels of an object; ranking loss evaluates the averaged fraction of label pairs mis-ordered for an object; and hamming loss measures the percentage of misclassified object-label pairs. The larger the average precision while the smaller the values of the other four criteria, the better the performance. It is evident that these eight criteria measure the performance of a method from different aspects.

Table 2 presents the annotation performance (mean ± standard deviation) on the top 10, 20 and 30 most frequent terms on the full set, and that of 10 terms on the small subset is tagged by 10S. It is impressive that MIMLSVM^+ outperforms all the other algorithms on all the criteria.

Compared with MKL-PMK, MIMLSVM^+ is more direct and natural for capturing the underlying nature of the gene expression annotation problem and thus leads to good results. Note that the computational complexity of MIMLSVM^+ is much smaller than that of MKL-PMK. MIMLSVM^+_V is the worst among MIMLSVM^+, MIMLSVM^+_{SV} and MIMLSVM^+_V. Considering that MIMLSVM^+_V does not use spatial information, it is clear that exploiting spatial information is helpful to improve the annotation performance. Table 2 also shows that the performance of MIMLSVM^+_{SV} is much better than that of the original MIMLSVM, although both of them employed the same method to utilize the spatial information of expression patterns. A possible reason is that the MIMLSVM algorithm employs a clustering process to transform MIML examples to multi-instance single-label examples, while this may lose some important discriminative information in the case of our annotation task.

To study the influence of the number of terms on the annotation performance, we run experiments with different numbers of terms and plot all the criteria in Figure 3. Since there are some terms annotated only to a few image panels, Ji *et al.* [2008] shows their results up to 60 CV terms. Hence, we follow the same set-up, and the top 60 most frequent CV terms are used for experiments. It can be found that when more terms are involved, the annotation performance drops. Nevertheless, the proposed MIMLSVM^+ algorithm is always the best among all competing algorithms.



(a) macro-F1 *(b) micro-F1* *(c) AUC* *(d) average precision*

(e) one-error *(f) coverage* *(g) ranking loss* *(h) hamming loss*

Figure 3: The performance of different methods under different number of terms. The MIMLSVM(+), MIMLSVM(+)-SV, MIMLSVM(+)-V in the legends represent MIMLSVM$^+$, MIMLSVM$^+_{SV}$ and MIMLSVM^+_V, respectively.

5 Conclusion

In this paper, a computational method for automatically annotating *Drosophila* gene expression patterns is proposed. We disclose that the essence of the gene expression pattern annotation task is a typical MIML learning problem, and we propose a simple yet effective MIMLSVM$^+$ algorithm for addressing this task. In the algorithm, visual features and spatial information of gene expression patterns are integrated for annotating anatomical and developmental terms to image panels. Empirical study on the BDGP image database validates the effectiveness of the proposed method.

Similar to previous MIML algorithms such as MIMLBoost and MIMLSVM, the MIMLSVM$^+$ algorithm also works by degeneration. On one hand, the superior performance of MIMLSVM$^+$ verifies the power of the MIML framework; on the other hand, it can be expected that if the problem can be tackled without degeneration, a better performance can be achieved, especially when a large number of terms needs to be annotated. One of our future proposals is to develop new MIML algorithms without degeneration to further improve gene expression pattern annotation performance.

Acknowledgements: The authors want to thank Ms. Kristi Garboushian for editorial support.

References

[Boutell *et al.*, 2004] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[Gärtner *et al.*, 2002] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola. Multi-instance kernels. In *ICML*, pages 179–186, Sydney, Australia, 2002.

[Gurunathan *et al.*, 2004] 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*, 5:202, 2004.

[Haussler, 1999] D. Haussler. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, Department of Computer Science, University of California at Santa Cruz, 1999.

[Ji *et al.*, 2008] S. Ji, L. Sun, R. Jin, S. Kumar, and J. Ye. Automated annotation of *drosophila* gene expression patterns using a controlled vocabulary. *Bioinformatics*, 24(17):1881–1888, 2008.

[Kumar *et al.*, 2002] 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*, 169:2037–2047, 2002.

[Lowe, 2004] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[Mikolajczyk and Schmid, 2005] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[Osuna *et al.*, 1997] E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. Technical Report AI Memo 1602, MIT Artificial Intelligence Laboratory, 1997.

[Schapire and Singer, 2000] R. E. Schapire and Y. Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2-3):135–168, 2000.

[Tomancak *et al.*, 2002] P. Tomancak, A. Beaton, R. Weiszmann, E. Kwan, S. Shu, S. E. Lewis, S. Richards, and et al. Systematic determination of patterns of gene expression during drosophila embryogenesis. *Genome Biology*, 3(12):R88, 2002.

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

[Zhou and Zhang, 2007] Z.-H. Zhou and M.-L. Zhang. Multi-instance multi-label learning with application to scene classification. In *NIPS 19*, pages 1609–1616. MIT Press, Cambridge, MA, 2007.