AUTOMATED GENE EXPRESSION PATTERN ANNOTATION IN THE MOUSE BRAIN

TAO YANG[1,2], XINLIN ZHAO[1,2], BINBIN LIN[2], TAO ZENG[3], SHUIWANG JI[3], JIEPING YE[1,2]

[1]*Department of Computer Science and Engineering,*
[2]*Center for Evolutionary Medicine and Informatics, The Biodesign Institute,*
Arizona State University, Tempe, AZ 85287, USA
[3]*Department of Computer Science,*
Old Dominion University, Norfolk, VA 23529, USA
E-mail: [1,2]{*T.Yang, Xinlin.Zhao, Binbin.Lin, Jieping.Ye*}*@asu.edu,* [3]{*tzeng, sji*}*@cs.odu.edu*

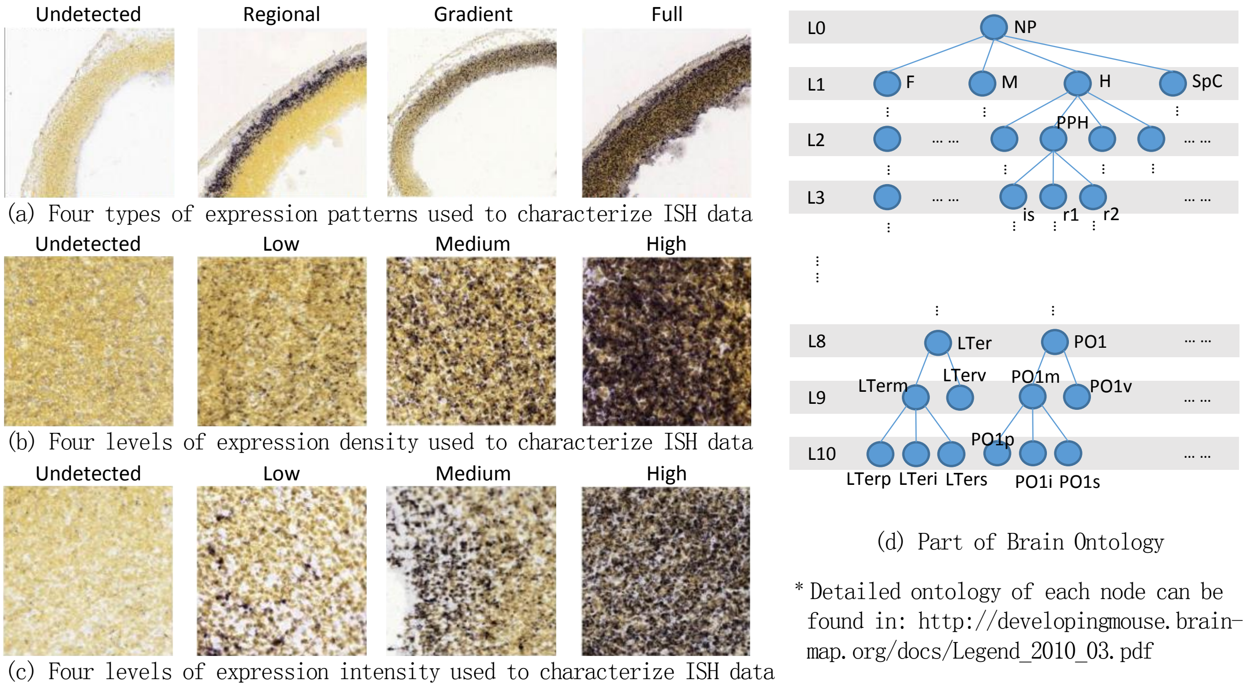
Brain tumor is a fatal central nervous system disease that occurs in around 250,000 people each year globally and it is the second cause of cancer in children. It has been widely acknowledged that genetic factor is one of the significant risk factors for brain cancer. Thus, accurate descriptions of the locations of where the relative genes are active and how these genes express are critical for understanding the pathogenesis of brain tumor and for early detection. The Allen Developing Mouse Brain Atlas is a project on gene expression over the course of mouse brain development stages. Utilizing mouse models allows us to use a relatively homogeneous system to reveal the genetic risk factor of brain cancer. In the Allen atlas, about 435,000 high-resolution spatiotemporal *in situ* hybridization images have been generated for approximately 2,100 genes and currently the expression patterns over specific brain regions are manually annotated by experts, which does not scale with the continuously expanding collection of images. In this paper, we present an efficient computational approach to perform automated gene expression pattern annotation on brain images. First, the gene expression information in the brain images is captured by invariant features extracted from local image patches. Next, we adopt an augmented sparse coding method, called Stochastic Coordinate Coding, to construct high-level representations. Different pooling methods are then applied to generate gene-level features. To discriminate gene expression patterns at specific brain regions, we employ supervised learning methods to build accurate models for both binary-class and multi-class cases. Random undersampling and majority voting strategies are utilized to deal with the inherently imbalanced class distribution within each annotation task in order to further improve predictive performance. In addition, we propose a novel structure-based multi-label classification approach, which makes use of label hierarchy based on brain ontology during model learning. Extensive experiments have been conducted on the atlas and results show that the proposed approach produces higher annotation accuracy than several baseline methods. Our approach is shown to be robust on both binary-class and multi-class tasks and even with a relatively low training ratio. Our results also show that the use of label hierarchy can significantly improve the annotation accuracy at all brain ontology levels.

Keywords: Gene Expression Pattern, Image Annotation, Sparse Learning, Imbalanced Learning, Multi-label classification, Label Hierarchy

1. Introduction

Brain tumor is a fatal central nervous system disease and it is the second cause of cancer in children.[1] Previous studies indicate that preventing and detecting brain tumors at early stages are effective methods to reduce brain damage; these studies also show the potential benefit of utilizing the genetic determinants.[2] Accurate descriptions of the locations of where the relative genes are active and how these genes express are critical for understanding the pathogenesis of brain tumor and for early detection.

An accurate characterization of the gene expression and its role on brain tumor requires extensive experimental resources on brain. A recent study[2] uses mouse to reveal the genetic risk factor of brain cancer. However, such study was performed on a limited set of genes. The Allen Developing Mouse Brain Atlas (ADMBA) is an online public repository of extensive gene



(a) Four types of expression patterns used to characterize ISH data

(b) Four levels of expression density used to characterize ISH data

(c) Four levels of expression intensity used to characterize ISH data

(d) Part of Brain Ontology

* Detailed ontology of each node can be found in: http://developingmouse.brain-map.org/docs/Legend_2010_03.pdf

Fig. 1. Sample schemas of different gene expression metrics and brain ontology

expression and neuroanatomical data over different mouse brain developmental stages.[3,4] The knowledge is documented as high-resolution spatiotemporal *in situ* hybridization (ISH) images for approximately 2,100 genes from embryonic through postnatal stages of brain development. In addition, a brain ontology has been designed to hierarchically organize brain structure for the developing form of mouse brain, which facilitates gene expression pattern annotation to specific brain areas. For a complete description of the status of gene expression revealed by *in situ* hybridization, three kinds of metrics, *i.e.*, pattern, density and intensity, are utilized at the Reference Atlas for ADMBA (R-ADMBA). These metrics were scored for each brain region according to a set of standard schemes; some examples are shown in Figure 1.

It is worthwhile to mention that such annotation tasks are very costly. The entire atlas contains around 435,000 ISH images and there are over 1,000 brain regions that need to be annotated in the designed brain ontology. To precisely assign gene expression metrics to specific brain areas, current reference atlas uses expert-guided manual annotation, which was performed by Dr. Martinez's team at Spain.[4,5] However, it is labor-intensive since it requires expertise in neuroscience and image analysis, and it does not scale with the continuously expanding collection of images. Therefore, developing an effective and efficient automated gene expression pattern annotation method is of practical significance.

The gene expression pattern annotation problem can be formulated as an image annotation problem, which has been widely studied in computer vision and machine learning. Specifically, a key to solve the problem is to learn effective feature representations of images. The scale-invariant feature transform (SIFT) algorithm has been commonly applied to transform image content into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters.[6] SIFT has been shown to be a powerful tool to capture patch-level characteristics of images. Based on those local image descriptors, the next step is to construct high-level feature representations of the ISH images. A common approach is to use the bag-of-words (BoW) model to represent high-level features, which has been used in a recent study.[7] However, BoW is not efficient to learn a large number of keywords or deal with large scale data atlas. In this study, we employ sparse coding to construct high-level features, which has been demonstrated to be effective in many fields including image recognition.[8] Sparse coding aims to using sparse linear combinations of basis vectors to reconstruct data vectors and learn a non-orthogonal and over-complete dictionary, which has more flexibility to represent the data.[9–11] The previous study[7] uses BoW instead of sparse coding mainly due to the high computational cost of solving the sparse coding problem especially for large-scale data in ADMBA. In this study, we adopt a novel implementation of sparse coding, called Stochastic Coordinate Coding (SCC),[12] which has been shown to be much more efficient than existing approaches.

Besides the image representation problem, many other difficulties are also inherent in the annotation tasks. First of all, for a specific set of ISH images, current reference atlas uses up to four categories [see Figure 1, (a)-(c)] to give an accurate description of the gene expression status for a specific metric. Thus, the annotation problem we are facing is indeed a multi-class classification problem. Secondly, the imbalanced class distribution is often involved in each annotation task, while traditional machine learning methods will often be biased and fail to provide reliable models.[13] In addition, annotating gene expression pattern over the brain ontology is essentially a multi-label classification problem. However, if we simply treat each label separately, we do not make full use of the structural relationships among labels [as shown in Figure 1 (d)] in the learning procedure, resulting in suboptimal prediction performance.[14,15]

In this paper, we propose an efficient computational approach to perform automated gene expression pattern annotation based on ADMBA ISH images. We first employ the SIFT method to construct local image descriptors. We next use sparse coding to efficiently learn the dictionary from SIFT descriptors of all ISH images and generate patch-level sparse feature representations of the images. Different pooling methods are utilized to combine patch-level representations to form image-level features, and further generate gene-level representations. To discriminate gene expression patterns over each brain area, we employ sparse logistic regression classifier and its multi-task extension to learn models for binary-class and multi-class classification. In addition, random undersampling and majority voting strategies are utilized to deal with imbalanced class distribution inherent within each annotation task. Furthermore, we make full use of the label hierarchy and dependency by developing a novel structure-based multi-label classification approach, which consists of two learning phases. In the first phase, a set of interested tasks (at the bottom of the label hierarchy) are learned individually, and in the second phase, knowledge learned from the first phase will be utilized to train models for the remaining tasks. We test our proposed approach on the four embryonic mouse developmental stages. Annotation results show that the adopted sparse coding approach outperforms the bag-of-words method. The proposed method provides favourable classification accuracy on both binary-class and multi-class tasks and even with a relatively low ratio of training. Experiment results also show that the structure-based multi-label classification approach can significantly improve the annotation accuracy at all brain ontology levels.

The remaining part of the paper is organized as follows: Section 2 details our feature extraction framework; Section 3 introduces several regularized learning methods, our strategies for learning from imbalanced data, and the proposed structure-based multi-label classification approach; Section 4 presents extensive empirical studies and Section 5 concludes the paper.

2. Proposed Feature Extraction Framework

2.1. *Image-level feature extraction*

Extracting and characterizing features from images is the key for image annotation. To capture as much gene expression details as possible over the entire brain ontology, ADMBA provides numerous spatiotemporal high-resolution ISH images. However, those raw images are not well aligned since they were taken from different samples and at different spatial slices. This makes it challenging to generate features from raw ISH images. A commonly used approach in such case is to employ the well-known scale-invariant feature transform method to construct local image descriptors. Specifically, the SIFT method first detects multiple localized keypoints (patches) from a raw image, and then transforms those image content into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters. We use the SIFT detection in VLFeat[16] and an average of 3,500 keypoints have been captured for each ISH image. In this study, each patch is represented by a 128-dimensional SIFT descriptor.

2.2. *High-level feature construction*

Based on the SIFT descriptors, we next apply sparse coding to construct high-level features. Sparse coding has been applied in many fields such as audio processing and image recognition. It refers to the process of using sparse linear combinations of basis vectors to reconstruct data and learning a non-orthogonal and over-complete dictionary. We can write the sparse coding problem as follows:

$$\min_{\mathbf{D},\mathbf{z}_1,\ldots,\mathbf{z}_n} \sum_{i=1}^{n} (\frac{1}{2}\|\mathbf{D}\mathbf{z}_i - \mathbf{a}_i\|_2^2 + \lambda\|\mathbf{z}_i\|_1)$$
$$s.t.\ \|\mathbf{D}_{\cdot j}\|_2 \leq 1, 1 \leq j \leq p \qquad (1)$$

where $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n] \in \mathbb{R}^{m\times n}$ is the set of SIFT descriptors constructed from image patches, each SIFT descriptor $\mathbf{a}_i \in \mathbb{R}^m$ is a m-dimension column vector with zero mean and unit norm, $\mathbf{D} \in \mathbb{R}^{m\times p}$ is the dictionary, λ is the regularization parameter, and $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n] \in \mathbb{R}^{p\times n}$ is the set of sparse feature representations of the original data. In addition, to prevent $\mathbf{D}$ from taking arbitrarily large values, the constraint, $\mathbf{D}_{\cdot j}, 1 \leq j \leq p$, restricts each column of $\mathbf{D}$ to be in a unit ball.

It has been known that solving the sparse coding problem is computationally expensive, especially when dealing with large-scale data and learning a large size of dictionary. The main computational cost comes from the updating of sparse codes and the dictionary. In our study, we adopt a new approach, called Stochastic Coordinate Coding (SCC), which has been shown to be much more efficient than existing methods.[12] The key idea of SCC is to alternately update the sparse codes via a few steps of coordinate descent and update the dictionary via second order stochastic gradient. In addition, by focusing on the non-zero components of the sparse codes and the corresponding dictionary columns during the updating procedure, the computational cost of sparse coding is further reduced.

In our study, the dictionary is learned from SIFT descriptors of all ISH images. The constraint, $\mathbf{z}_i \geq 0$, $1 \leq i \leq n$, is further added to ensure the non-negativity of sparse codes. To generate image-level features based on patch-level representations, we apply the max-pooling operation. Max-pooling takes the strongest signal among multiple patches to represent the image, which has been shown to be powerful in combining low-level sparse features.[17]

2.3. *Gene-level feature pooling*

Recall that a specific ISH image is obtained from particular brain spatial coordinates and it may not be able to present the gene expression pattern over the entire brain ontology. In order to describe expression pattern at all brain regions, we use a gene-level feature pooling. Since it remains unclear what kind of pooling methods will perform better on those high-level representations, both average-pooling and max-pooling are employed in our study.

3. Gene Expression Pattern Classification Methods

In this section, we introduce several regularized learning methods for gene expression pattern classification as well as our strategies for learning from imbalanced data. In addition, we present a structure-based multi-label classification approach for annotation.

3.1. *Sparse logistic regression*

We first consider the simple case: binary classification. Specifically, for a certain metric of gene expression, we convert the original annotation task into a binary classification problem by treating the category "undetected" as one class and all remaining categories as the other class. We employ the regularized supervised learning methods, which have been widely used in machine learning and bioinformatics. Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n \in \mathbb{R}^{n \times p}$ denote a p dimensional data set with n observations, and $\mathbf{y} = \{y_i\}_{i=1}^n \in \mathbb{R}^{n \times 1}, y_i \in \{-1, 1\}$ be the corresponding labels. Then, we can write the sparse logistic regression problem as follows:

$$\min_{\mathbf{w}} \mathcal{L}(\mathbf{X}\mathbf{w}, \mathbf{y}) + \lambda \|\mathbf{w}\|_1, \tag{2}$$

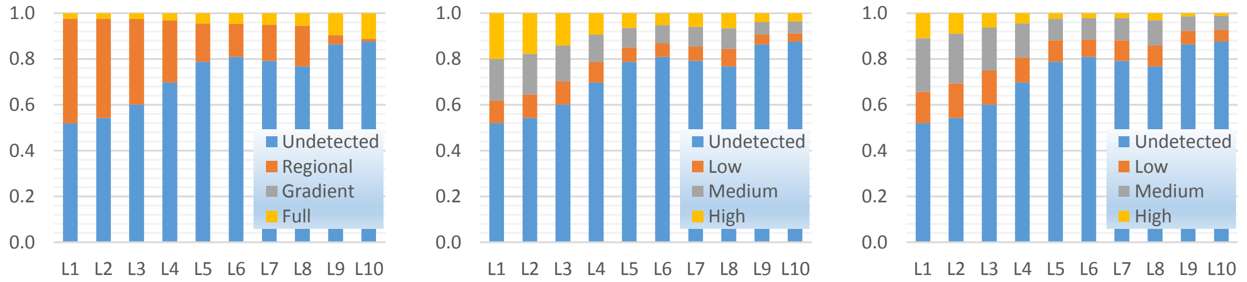
where $\mathcal{L}(\cdot)$ denotes the logistic loss, $\mathbf{w} \in \mathbb{R}^{p \times 1}$ is the model weight vector and λ is the l_1-norm regularization parameter. The solution of the above system will yield sparsity in $\mathbf{w}$, and the significant columns of $\mathbf{X}$ are determined by the corresponding non-zero entries in $\mathbf{w}$. In our study, $\mathbf{x}_i$ is a gene-level representation (after patch-level pooling and image-level pooling) and y_i encodes the annotation of gene expression status for a specific brain region.

3.2. *Multi-task sparse logistic regression*

We also propose to directly solve the multi-class annotation problem via multi-task learning. Suppose there are k classes ($k = 3$ or 4 in our study). We can represent the category of a sample by a k-tuple, where $y_{ik} = 1$ if sample i belongs to class k and $y_{ik} = -1$ otherwise. Then we can rewrite the response $\mathbf{Y}$ as $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^n \in \mathbb{R}^{n \times k}$. We employ the following multi-task sparse logistic regression formulation for the multi-class case:

$$\min_{\mathbf{W}} \mathcal{L}(\mathbf{X}\mathbf{W}, \mathbf{Y}) + \lambda \|\mathbf{W}\|_{2,1}, \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{p \times k}$, and the i-th column of $\mathbf{W}$ is the model weight for the i-th task. The $l_{2,1}$-norm penalty on $\mathbf{W}$ results in grouped sparsity, which restricts all tasks to share a common set of features. In this paper, we employ this multi-task model to solve the multi-class annotation problem. The SLEP[18] package is utilized to solve both Problem (2) and Problem (3).



(a) Pattern (b) Density (c) Intensity

Fig. 2. Percentage of different categories of gene expression status at each brain level

3.3. *Undersampling and majority voting*

In this study, regions in the brain ontology are divided into 10 levels. Figure 2 shows the statistics of annotation distribution at each brain ontology level. It can be observed that even for the binary classification case, the data imbalance problem is particularly severe. A desired training set should contain approximately equal numbers of observations from each category. Traditional machine learning methods may be very sensitive to imbalance issue since the models will be biased toward the majority class of samples. To learn a better model from an imbalanced data set, a simple and intuitive idea is to balance the training set. Some existing studies suggest that random undersampling method is effective in dealing with data imbalance.[13] Besides undersampling, model ensemble is also beneficial for learning from imbalanced data.[19] Ensemble methods refer to the process of combining multiple models to improve predictive performance. The idea of classifier ensemble is to build a prediction model by combining a set of individual decisions from multiple classifiers.[20] In this study, we employ undersampling multiple times, combine a set of learning models, one for each undersampled data, and finally use majority voting to infer the predictions.

3.4. *Structure-based multi-label annotation over brain ontology*

Annotating gene expression patterns over the brain ontology is indeed a multi-label classification problem. In the reference atlas, the expression patterns of a single gene are recorded based on a hierarchically organized ontology of anatomical structures. In practice, it is possible to propagate annotation to parent or child structures under a set of systematic rules.[4] Rather than simply treating each individual annotation task separately, if we build all prediction models together by utilizing the structure information among labels, the predictive performance can potentially be significantly improved.[14,15]

In this study, we propose a novel structure-based multi-label classification approach. Suppose we are given n training data points $\{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^n$, where $\mathbf{x}^i \in \mathbb{R}^p$ is a data point of p features, and $\mathbf{y}^i \in \mathbb{R}^k$ is the corresponding label vector of k tasks. Let $j \in \{1, \ldots, k\}$ denote the j-th learning task. We then divide the learning procedure into two phases. Assuming there are t tasks ($t < k$) at the bottom level of the hierarchy, in the first phase, each of those tasks is learned individually by:

$$\tilde{y}_j = \mathcal{F}_j(\tilde{\mathbf{x}}), \; 1 \leq j \leq t < k, \tag{4}$$

where $\mathcal{F}_j(\cdot)$ denotes a learnt model by the j-th task, $\tilde{\mathbf{x}} \in \mathbb{R}^p$ is an arbitrary data point, and

$\tilde{y}_j \in \mathbb{R}$ is the prediction of $\tilde{\mathbf{x}}$ for the j-th task. The learned knowledge in (4) is then used to learn the remaining tasks (*i.e.*, $t+1 \le j \le k$) in the second phase. Specifically, we augment the feature set by adding the prediction probabilities learnt in the previous phase, *i.e.*, we denote $\tilde{\mathbf{x}}' = [\tilde{\mathbf{x}}, (\tilde{y}_1, \ldots, \tilde{y}_t)]$. Annotation tasks in the second phase will be performed based on this augmented feature set $\tilde{\mathbf{x}}'$.

The tasks in the first phase can be considered as the auxiliary tasks in the second phase.[21] We apply the two-stage approach in our case since the tasks are not symmetric due to the hierarchical label structure. With the prediction probabilities from the previous learning phase, we make use of label dependency along with the original image representations. Intuitively, if a new learning task is related to some of the tasks learnt in the first phase, then such approach is expected to achieve better classification accuracy. In our study, since the tasks associated with the bottom of the label hierarchy are related to the remaining tasks in the hierarchy, the prediction performance is expected to be improved by the two-stage learning approach. This is confirmed in our experiments presented in the next section.

4. Experiments

We design a serial of experiments to evaluate the proposed approach for gene expression pattern annotation on the Allen Developing Mouse Brain Atlas. Specifically, we evaluate our approach in the following four aspects: (1) Comparison of sparse coding and bag-of-words, (2) Comparison of different training ratios, (3) Comparison of different multi-class annotation methods, and (4) Comparison of annotation with and without brain ontology.

4.1. *Data description and experimental setup*

The gene expression ISH images are obtained from the Allen Developing Mouse Brain Atlas. Specifically, to ensure the consistency of brain ontology over different mouse developmental stages, we focus our experiments on the four embryonic stages, namely, E11.5, E13.5, E15.5 and E18.5. The ADMBA provides approximately 2,100 genes within each stage and an average of 15∼20 images are used for each gene to capture the expression information over the entire 3D brain. The total number of ISH images in these four stages are 142,425. We use the SIFT method to detect local gene expression and apply sparse coding to learn sparse feature representations for image patches. Considering the resolution of the ISH images and the number of areas of the mouse brain ontology, a dictionary size of 2,000 is chosen, *i.e.*, $\mathbf{D} \in \mathbb{R}^{128 \times 2000}$. To generate gene-level representations, both max-pooling and average-pooling are used.

To evaluate the effectiveness of the proposed methods, we compare our approach with the well-known bag-of-words (BoW) method. Specifically, the BoW is performed in two different settings: the first approach, called non-spatial BoW, concatenates three BoW representations of SIFT features, where each BoW is learned from the ISH images at a specific scale; the second approach, called spatial BoW, divides the brain sagitally into seven intervals according to the spatial coordinate of each image, and then 21 regional BoW representations are built (7 intervals × 3 scales).[7] At each scale, a fixed number of 500 clusters (keywords) are constructed from SIFT features and an extra dimension is used to count the number of zero descriptors.

R-ADMBA uses three different metrics including pattern, density and intensity, to evaluate the gene expression pattern on each brain ontology area. As discussed in the previous section, we consider the annotation tasks as either binary-class or multi-class classification problem. For the simple binary-class case, the category "undetected" is treated as the negative class, which refers to the scenario that no gene expression pattern is detected at the specific brain area, and all remaining categories are treated as the positive class, which means some kind of expression pattern has been detected. It is worthwhile to note that, at such a binary-class situation, if the annotation metric "pattern" is marked as "undetected", then metrics "density" and "intensity" must be "undetected", and *vice versa*. That is, it is possible to use a single metric to evaluate the gene expression status at this case.

In addition, in order to balance the class distributions of training sets, random undersampling are performed for 11 times. To give a baseline performance of the traditional method, the experiment results of using Support Vector Machine (SVM) classifier[22] is also reported. To better describe the classification performance under the circumstances of data imbalance, we use the area under the curve (AUC) of a receiver operating characteristic (ROC) curve as the performance measure for binary-class classification. The accuracy is used as the performance measure for the multi-class case.

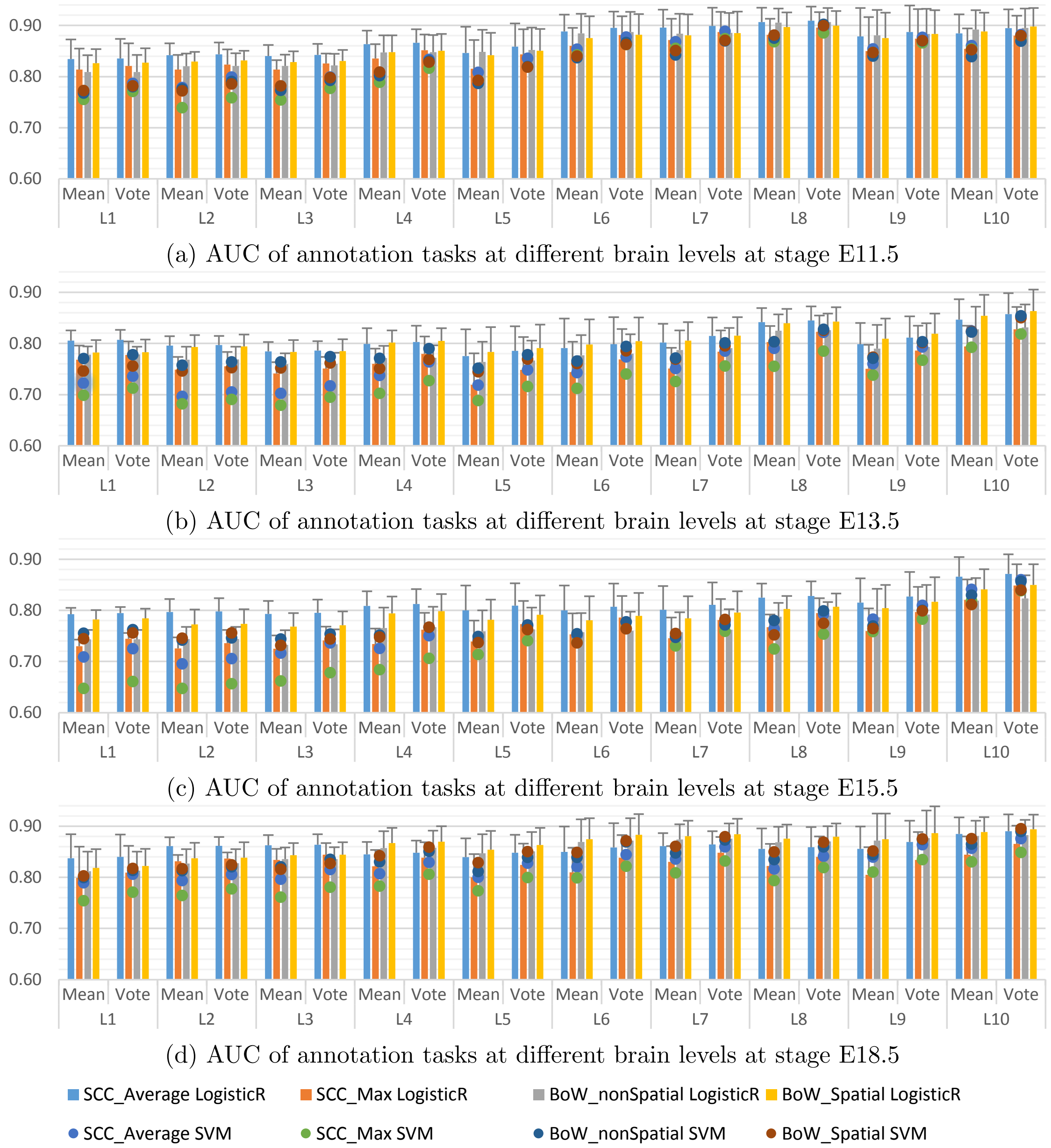
4.2. *Comparison of sparse coding and bag-of-words*

We use the first serial of experiments to compare sparse coding with the bag-of-words method. Specifically, we generate the training data from raw gene expression ISH images using the following four methods: (1) SCC_Average, using SCC to learn image-level representations and average-pooling to generate gene-level features; (2) SCC_Max, similar to (1) but using max-pooling to generate gene-level features; (3) BoW_nonSpatial, generating single bag-of-words representation using all ISH images; (4) BoW_Spatial, generating multiple bag-of-words representations using ISH images from different spatial coordinates. Here we only consider the simple binary-class situation, and the entire data set is being randomly partitioned into training set and testing set for each annotation task using a ratio of 4:1. In addition, in comparison with the proposed majority voting strategy, the average classification performance of 11 times undersampling is also recorded. The overall classification performance for each brain ontology level at different developmental stages are summarized in Figure 3.

We can observe from Figure 3 that the proposed approach achieves the highest overall AUC of 0.9095, 0.8573, 0.8717 and 0.8903 at mouse brain developmental stages E11.5, E13.5, E15.5 and E18.5 respectively. For the comparison of different types of image representations, SCC_Average provides the best overall performance among all four stages. Although in some annotation tasks, BoW_Spatial provides competitive performance to SCC_Average, it is worthwhile to note that, the spatial BoW ensembles 21 single dictionaries and contains more than 10,000 features. Thus, spatial BoW is far more complex than SCC and involves higher computational costs. We can also observe that the use of undersampling and majority voting strategies improves the individual model by 1% $\sim$ 3% in terms of AUC. Moreover, in comparison with SVM classifier, the sparse logistic regression classifier achieves better predictive performance. Those experimental results verify the superiority of our proposed methods.

4.3. *Comparison of different training ratios*

In this experiment, we compare the classification performance of using different training ratios. More specifically, we would like to verify the robustness of the presented approach when using a



(a) AUC of annotation tasks at different brain levels at stage E11.5

(b) AUC of annotation tasks at different brain levels at stage E13.5

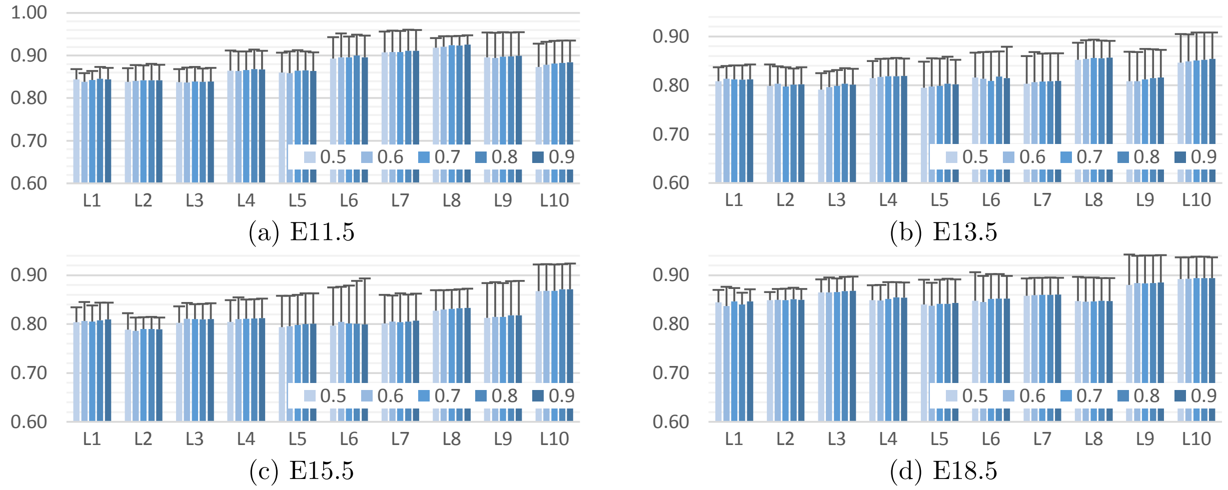
(c) AUC of annotation tasks at different brain levels at stage E15.5

(d) AUC of annotation tasks at different brain levels at stage E18.5

Fig. 3. Comparison of the proposed approach and bag-of-words method. Each column bar represents the performance of using sparse logistic regression classifier for a specific set of gene-level image representations. Each dot represents the performance of using SVM classifier for a specific set of gene-level image representations. The error bar of each column is the standard deviation of annotation performance within the corresponding brain level. "Mean" group records the average performance of 11 sub-models. "Vote" group records the performance of using majority voting.

relatively small number of samples for training. According to the first serial of experiments, we use the SCC_Average to construct features in this experiment. For each annotation task, we fix 10% of the samples as testing set and vary the ratio of training set in $\{50\%, 60\%, 70\%, 80\%, 90\%\}$. The experimental results are summarized in Figure 4.

We can observe from the figure that, at all four mouse brain developmental stages and all brain levels, no significant difference is observed between different training ratios. We can



(a) E11.5

(b) E13.5

(c) E15.5

(d) E18.5

Fig. 4. Classification performance (AUC) of the proposed approach of using different training ratios. Shading of the bars from light to dark indicates training ratio from 0.5 ∼ 0.9. The error bar of each column is the standard deviation of annotation performance within the corresponding brain level.

conclude from this experiment that our proposed approach is robust even with a low training ratio, thus accurate models for gene expression annotation can be learned based on a relative small number of manually annotated images.

4.4. *Comparison of different multi-class annotation methods*

In this experiment, we evaluate our multi-task sparse logistic regression (mcLR) approach in the multi-class annotation situation. Data set SCC_Average is employed and we use the multi-class SVM (mcSVM) as the baseline for performance comparison. In this experiment, 80% of the samples from each class are randomly chosen as the training set, and the remain 20% of the samples are used as the testing set. We only include annotation classes if there are more than 100 samples available for a specific class. The accuracy is used as the performance measure and the results are reported in Table 1.

We can observe that our proposed approach using sparse logistic regression with grouped sparsity constraint provides favourable predictive accuracy for this multi-class annotation task. Specifically, the classification accuracy of mcLR is significantly higher than mcSVM at all brain stages and levels. All detailed gene expression status measured by pattern, density and intensity can be well distinguished by our classifiers. These results imply that those multiple classes are inherently related and it is beneficial to learn four (or three) classification models simultaneously by restricting all models to share a common set of features. We plan to explore other multi-task learning models in our future work.[23]

4.5. *Comparison of annotation with and without brain ontology*

Recall that the expert-guided manual annotations are based on a hierarchically organized ontology of anatomical structures. Rather than learning each task individually, it may be beneficial to utilize the hierarchy among the labels for a joint annotation. As we can observe from previous experiments, models learned in a lower level typically have better predictive performance. Thus it is natural to make use of the lower-level models and label structures to improve the prediction performance of high-level tasks.

Table 1. Classification accuracy (in percentage) of multi-class annotation.

		Pattern		Density		Intensity	
		mcSVM	mcLR	mcSVM	mcLR	mcSVM	mcLR
(a) E11.5	L5	77.26	**80.38**	71.52	**74.93**	76.98	**80.90**
	L6	79.29	**81.78**	80.68	**82.97**	79.93	**83.57**
	L7	77.69	**80.43**	77.34	**79.88**	79.33	**82.54**
	L8	81.61	**84.35**	83.40	**85.46**	83.98	**85.80**
	L9	77.02	**81.10**	85.40	**87.16**	84.84	**87.04**
	L10	—[a]	—	—	—	—	—
(c) E15.5	L5	80.91	**86.78**	70.13	**74.52**	72.17	**77.15**
	L6	79.66	**83.09**	76.42	**80.03**	76.28	**80.83**
	L7	—	—	74.55	**78.53**	75.36	**80.05**
	L8	74.97	**79.79**	70.93	**75.36**	72.83	**78.23**
	L9	78.84	**82.39**	80.49	**83.26**	80.32	**83.97**
	L10	—	—	86.16	**87.61**	87.03	**88.26**

		Pattern		Density		Intensity	
		mcSVM	mcLR	mcSVM	mcLR	mcSVM	mcLR
(b) E13.5	L5	73.53	**80.10**	67.91	**73.87**	70.51	**76.62**
	L6	75.80	**81.79**	72.87	**77.76**	73.35	**78.40**
	L7	72.07	**77.56**	72.09	**77.05**	73.71	**78.85**
	L8	71.83	**77.03**	70.82	**75.02**	73.90	**78.35**
	L9	78.54	**82.26**	81.12	**84.66**	80.57	**84.34**
	L10	—	—	85.36	**87.48**	83.22	**86.00**
(d) E18.5	L5	75.41	**78.93**	72.73	**76.18**	75.22	**78.96**
	L6	83.08	**87.37**	76.67	**79.55**	78.94	**82.01**
	L7	77.34	**79.65**	75.78	**78.77**	77.67	**80.79**
	L8	—	—	73.79	**77.48**	75.89	**79.58**
	L9	79.49	**81.42**	79.17	**81.56**	80.59	**83.01**
	L10	79.38	**81.45**	82.94	**84.96**	83.52	**85.56**

In this study, we compare our proposed structure-based multi-label learning (SMLL) method with the simple individual annotation, which builds models for different tasks independently. Again, we employ the SCC_Average method to construct the data. At each brain developmental stage, around 200 genes are randomly pre-selected as the testing set for the annotation tasks over the entire brain ontology and the remaining genes are included in the training set. For SMLL method, 432 tasks (regions) at level 10 (L10) are learned individually in the first phase. The prediction probabilities of L10 tasks will be used as the additional features in the data. In this experiment, we consider the binary-class situation and results are summarized in Table 2.

We can observe from Table 2 that the overall annotation performance achieved by SMLL is higher than the individual model. Improvements in terms of AUC can be observed at most of the brain ontology levels among all developmental stages. This verifies the effectiveness of the proposed structured-based multi-label learning approach.

5. Conclusion

In this paper, we propose an efficient computational approach to perform automated gene expression pattern annotation on mouse brain images. The key information in spatiotemporal *in situ* hybridization images is first captured by the SIFT method from local image patches. Image-level features are then constructed via sparse coding. To generate gene-level representations, different pooling method are adopted. Regularized learning methods are employed to build classification models for annotating gene expression pattern at different brain regions. To utilize hierarchy information among the brain ontology, a novel structure-based multi-label classification approach is proposed. Extensive experiments have been conducted on the atlas and results demonstrate the effectiveness of the proposed approach. One of our future directions is to explore deep learning models to learn feature representations from ISH images. In addition, we plan to explore other multi-task learning models to make more effective use of the label hierarchy in the annotation.

[a] "—" means the experiment is not applicable for the specific brain ontology level.

Table 2. Classification performance (AUC) of structure-based multi-label annotation.

	E11.5				E13.5				E15.5				E18.5			
	LogisticR		SVM		LogisticR		SVM		LogisticR		SVM		LogisticR		SVM	
	Single	SMLL	Single	SMLL	Single	SMLL	Single	SMLL	Single	SMLL	Single	SMLL	Single	SMLL	Single	SMLL
L1	0.837	0.811	0.806	**0.837**	0.793	0.781	0.737	**0.778**	0.744	**0.749**	0.657	**0.699**	0.890	0.878	0.845	**0.879**
L2	0.866	0.850	0.854	**0.877**	0.774	0.772	0.744	**0.785**	0.755	**0.764**	0.632	**0.695**	0.894	0.882	0.831	**0.884**
L3	0.898	0.884	0.884	**0.903**	0.799	0.797	0.766	**0.808**	0.781	**0.788**	0.634	**0.710**	0.893	0.885	0.833	**0.885**
L4	0.941	**0.941**	0.932	**0.951**	0.868	**0.874**	0.843	**0.873**	0.796	**0.803**	0.665	**0.709**	0.891	0.890	0.852	**0.888**
L5	0.905	**0.908**	0.904	**0.922**	0.843	**0.855**	0.822	**0.848**	0.838	**0.844**	0.710	**0.746**	0.871	**0.876**	0.837	**0.850**
L6	0.935	**0.937**	0.937	**0.947**	0.898	**0.907**	0.882	**0.898**	0.843	**0.851**	0.744	**0.760**	0.871	**0.878**	0.844	**0.855**
L7	0.951	0.950	0.950	**0.959**	0.860	**0.866**	0.842	**0.863**	0.846	**0.858**	0.743	**0.777**	0.894	**0.896**	0.874	**0.890**
L8	0.980	**0.982**	0.980	**0.984**	0.932	**0.937**	0.905	**0.932**	0.835	**0.841**	0.810	**0.836**	0.894	**0.896**	0.863	**0.882**
L9	0.966	**0.969**	0.971	**0.972**	0.890	**0.896**	0.877	**0.884**	0.865	**0.873**	0.811	**0.816**	0.871	**0.872**	0.852	0.843
L10	0.971	—	0.976	—	0.906	—	0.904	—	0.877	—	0.837	—	0.896	—	0.884	—

Acknowledgments

This work is supported in part by research grants from NIH (R01 LM010730) and NSF (IIS-0953662, IIS-1421057, IIS-1421100, DBI-1147134 and DBI-1350258).

References

1. W. H. Organization, World cancer report 2014. (2014).
2. K. M. Reilly, *Brain pathology* **19**, 121 (2009).
3. ©2013 Allen Institute for Brain Science, Allen Developing Mouse Brain Atlas [Internet]. Available from: `http://developingmouse.brain-map.org`.
4. ©2013 Allen Institute for Brain Science, *Allen Developing Mouse Brain Atlas*, tech. rep.
5. C. L. Thompson, L. Ng, V. Menon, S. Martinez *et al.*, *Neuron* (2014).
6. D. G. Lowe, Object recognition from local scale-invariant features, in *IEEE ICCV*, 1999.
7. T. Zeng and S. Ji, *Automated Annotation of Gene Expression Patterns in the Developing Mouse Brain Atlas*, tech. rep. (2014).
8. A. Szlam, K. Gregor and Y. LeCun, Fast approximations to structured sparse coding and applications to object classification, in *Computer Vision–ECCV 2012*, (Springer, 2012) pp. 200–213.
9. B. A. Olshausen *et al.*, *Nature* **381**, 607 (1996).
10. S. S. Chen, D. L. Donoho and M. A. Saunders, *SIAM J. Sci. Comput.* **20**, 33 (1998).
11. D. L. Donoho and M. Elad, *Proceedings of the National Academy of Sciences* **100**, 2197 (2003).
12. B. Lin, Q. Li, Q. Sun, M.-J. Lai, I. Davidson, W. Fan and J. Ye, *CoRR* **abs/1407.8147** (2014).
13. H. He and E. A. Garcia, *Trans. Knowl. Data Eng., IEEE Trans. on* **21**, 1263 (2009).
14. C. N. Silla Jr and A. A. Freitas, *Data Mining and Knowledge Discovery* **22**, 31 (2011).
15. G. Tsoumakas, I. Katakis and I. Vlahavas, Mining multi-label data, in *Data mining and knowledge discovery handbook*, (Springer, 2010) pp. 667–685.
16. A. Vedaldi and B. Fulkerson, VLFeat: An open and portable library of computer vision algorithms `http://www.vlfeat.org/`, (2008).
17. Y.-L. Boureau, J. Ponce and Y. LeCun, A theoretical analysis of feature pooling in visual recognition, in *Proceedings of the 27th ICML*, 2010.
18. J. Liu, S. Ji and J. Ye, *SLEP: Sparse Learning with Efficient Projections.* ASU, (2009).
19. R. Dubey, J. Zhou, Y. Wang, P. M. Thompson and J. Ye, *NeuroImage* **87**, 220 (2014).
20. R. Polikar, *Circuits and Systems Magazine, IEEE* **6**, 21 (2006).
21. R. K. Ando and T. Zhang, *The Journal of Machine Learning Research* **6**, 1817 (2005).
22. C.-C. Chang and C.-J. Lin, *ACM TIST* **2**, 27:1 (2011), `www.csie.ntu.edu.tw/~cjlin/libsvm`.
23. J. Zhou, J. Chen and J. Ye, *MALSAR.* ASU, (2011).