SUPPLEMENT TO "AUTOMATED ANNOTATION OF *DROSOPHILA* GENE EXPRESSION PATTERNS USING A CONTROLLED VOCABULARY"

Citations are to the **References** in the main paper.

Pyramid match kernels

We consider the space of feature sets: $S = \{X | X = \{x_1, \cdots, x_n\}\}$, where $x_i \in V \subseteq \mathbb{R}^d$ is the feature vector, V is the vector space of features, and n is the cardinality of the feature set. Note that the pyramid match algorithm requires the dimensions of the feature vectors to be the same, i.e., d, while the cardinalities of different sets can be different. Given two vector sets Y and Z in S, a partial matching between Y and Z is an assignment of each vector in the smaller set to a unique vector in the larger set. In the general case of sets with different cardinalities, there are some vectors left unmatched in the larger set. The optimal partial matching between two vector sets corresponds to the assignment that results in maximum total similarity obtained by summing up the similarities between each pair of matched vectors. Exact computation of the optimal partial matching is computationally expensive. The pyramid matching is an efficient approximation to the optimal partial matching with linear (in the number of feature vectors in each set) time complexity.

The pyramid match algorithm partitions the feature space V into bins (regions) with increasingly larger granularity. At the finest level, all distinct vectors are guaranteed to be in different bins. The size of the bin increases gradually until a single bin occupies the entire feature space. Each partition of the feature space results in a histogram, and the different partitions lead to a hierarchy (pyramid) of histograms with increasingly coarser resolution (Fig. 2). It follows from this construction that no vectors share the same bin at the finest level of resolution while all vectors share the same bin at the coarsest level. Hence, any two feature vectors from any two sets begin to share a bin at some level in this histogram pyramid, and they are considered to be matched at this point. The distance between any two vectors can be bounded from above by the size of the bin in which they are matched. Thus the pyramid match algorithm can extract an approximate matching score between two sets of vectors without computing any of the pairwise similarities.

Let the histogram pyramid for a vector set X be defined as:

$$\Psi(X) = [H_0(X), \cdots, H_{L-1}(X)],$$

where $X \in S$, $L = \lceil \log_2 R \rceil + 1$, R is the maximal range of the components of vectors in V, and $H_i(X)$ is a histogram vector for the set X corresponding to the histogram with bins of side length 2^i. In this construction of histogram pyramid, the side length of the bin is doubled at each of the successive steps. To calculate the similarity between two histograms, we need to define the histogram intersection function I as

$$I(U, V) = \sum_{i=1}^{m} \min(U_i, V_i),$$

where U and V are two histograms with m bins, and U_i and V_i are the counts for the ith bins of U and V, respectively. A newly matched pair at certain level is defined as a pair of vectors that share a bin at this level of resolution, but are in different bins at any finer resolution level. The number of newly matched pairs C_i at the ith level can be obtained by subtracting the intersection function evaluated for the ith level by that of the immediate preceding level as

$$C_i = I(H_i(Y), H_i(Z)) - I(H_{i-1}(Y), H_{i-1}(Z)).$$

The similarity between the two sets Y and Z is defined as a weighted sum of the number of newly matched pairs at each level of the histogram pyramid. Intuitively, vectors that are matched in smaller bins should be given larger weights than these matched in larger bins. In the initial construction proposed in Grauman and Darrell (2005), the weight for the ith level histogram is set to $d2^i$ to reflect the geometric bound for the distance between any two feature vectors sharing a particular bin. The (unnormalized) similarity between Y and Z can be defined as:

$$\tilde{P}(\Psi(Y), \Psi(Z)) = \sum_{i=0}^{L-1} d2^i C_i.$$

Finally, to avoid favoring sets with larger cardinality, the similarity is normalized by dividing each set's self-similarities as:

$$P(\Psi(Y), \Psi(Z)) = \frac{1}{N} \tilde{P}(\Psi(Y), \Psi(Z)),$$

where $N = \tilde{P}(\Psi(Y), \Psi(Y)) \tilde{P}(\Psi(Z), \Psi(Z))$.

The original pyramid match algorithm proposed in Grauman and Darrell (2005) generates uniform bins over the feature space, and the size of the bin is doubled at each of the successive steps in the pyramid. Such a pre-determined construction fails to take advantage of the underlying structure in the feature space, and it suffers from distortion factors that increase linearly with the dimension of the features (Grauman and Darrell, 2006). Grauman and Darrell (2006) proposed the vocabulary-guided pyramid match algorithm in which the positions and sizes of bins in the multi-resolution histograms are determined by applying the hierarchical clustering algorithm on the feature vectors. Each level in the hierarchical clustering corresponds to one level in the histogram pyramid, and the position and size of each bin at a particular level is determined by the clusters at that level. The weight for each match can also be made data-adaptive by estimating the inter-feature distance geometrically. It was shown that this data-dependent hierarchical decomposition scheme can maintain consistent accuracy when the dimension of the feature space increases (Grauman and Darrell, 2006).

Proof of Theorem 3.1

We show in the following lemma that maximization of the objective function in Eq. (1) is equivalent to the minimization of an alternative criterion.

LEMMA 5.1. *The kernel matrix that maximizes the objective function in Eq. (1) is also the minimizer of the following objective function:*

$$F_1(K, B) = \sum_{i=1}^{k} \left(\|K\beta_i - h_i\|^2 + \lambda \beta_i^T K \beta_i \right), \qquad (6)$$

where $B = (\beta_1, \cdots, \beta_k)$, h_i is the ith column of H where $HH^T = C$.

PROOF. Since the null space of $K^2 + \lambda K$ lies in the null space of KCK, the optimal value achieved by the optimization problem in Eq. (1) is given by trace $\left((K^2 + \lambda K)^+ KCK\right)$. Consider the maximization of the following objective function with respective to B:

$$F_2(K, B) = \sum_{i=1}^{k} \frac{\left(\beta_i^T K h_i\right)^2}{\beta_i^T \left(K^2 + \lambda K\right) \beta_i}. \quad (7)$$

The optimal β_i is given by $\beta_i^* = (K^2 + \lambda K)^+ K h_i$. Thus the maximum value of $F_2(B, K)$ achieved by $B^* = [\beta_1^*, \cdots, \beta_k^*]$ is given by $F_2^*(K) = \text{trace}\left((K^2 + \lambda K)^+ KCK\right)$. This shows that, when optimized with respective to B, the objective functions in Eqs. (1) and (7) achieve the same value.

We next show the equivalence between objective functions in Eqs. (6) and (7). The objective function $F_1(K, B)$ in Eq. (6) is the least squares cost function in the kernel-induced feature space, and its optimal value is

$$F_1^*(K) = -\sum_{i=1}^{k} (Kh_i)^T (K^2 + \lambda K)^+ K h_i + h_i^T h_i. \quad (8)$$

Thus maximizing $\sum_{i=1}^{k} (Kh_i)^T (K^2 + \lambda K)^+ K h_i$ with respective to K is equivalent to minimizing $F_1^*(K)$. This completes the proof.

Note that the matrix C obtained from standard clique and star expansions is symmetric and positive semidefinite. Thus the matrix H is always well-defined. We are now ready to prove Theorem 3.1.

PROOF. Let $\xi_i = K\beta_i - h_i$. Define the Lagrangian function for the optimization problem in Eq. (6) as follows:

$$L = \sum_{i=1}^{k} \|\xi_i\|^2 + \lambda \sum_{i=1}^{k} \beta_i^T K \beta_i - \sum_{i=1}^{k} \psi_i^T (K\beta_i - h_i - \xi_i), \quad (9)$$

where the ψ_i's are the vectors of Lagrangian dual variables. By following the standard Lagrangian technique, we get the Lagrangian dual function as

$$g(\psi_1, \cdots, \psi_k) = \sum_{i=1}^{k} \left(-\frac{1}{4} \psi_i^T \left(I + \frac{1}{\lambda} K \right) \psi_i + \psi^T h_i \right). \quad (10)$$

The optimal $\psi_1^*, \cdots, \psi_k^*$ can be obtained by maximizing the dual function. Since strong duality holds, the primal and dual objectives coincide and the optimal K can be computed by solving the following optimization problem:

$$\min_{\substack{\theta \,:\, \theta \geq 0, \\ \theta^T r = 1}} \; \max_{\psi_1, \cdots, \psi_k} \left\{ \sum_{i=1}^{k} \left(-\frac{1}{4} \psi_i^T \left(I + \frac{1}{\lambda} \sum_{j=1}^{p} \theta_j K_j \right) \psi_i + \psi_i^T h_i \right) \right\}.$$

Since $\sum_{j=1}^{p} \theta_j r_j = 1$, the above objective function can be expressed as

$$\sum_{j=1}^{p} \theta_j \sum_{i=1}^{k} \left(-\frac{r_j}{4} \psi_i^T \psi_i - \frac{1}{4\lambda} \psi_i^T K_j \psi_i + r_j \psi_i^T h_i \right),$$

Thus it follows from the definition of $S_j(\Psi)$ in Eq. (5) that the optimization problem in Eq. (5) can be expressed equivalently as

$$\max_{\theta : \theta \geq 0, \theta^T r = 1} \; \min_{\Psi} \sum_{j=1}^{p} \theta_j S_j(\Psi). \quad (11)$$

Assume Ψ^* is the optimal solution to the problem in Eq. (11), and define $\gamma^* = \sum_{j=1}^{p} \theta_j S_j(\Psi^*)$ as the minimum value. We have $\sum_{j=1}^{p} \theta_j S_j(\Psi) \geq \gamma^*$, for all Ψ. By defining $\gamma = \min_{\Psi} \sum_{j=1}^{p} \theta_j S_j(\Psi)$ and substituting γ into the objective, we prove this theorem.

Solving the SILP problem

The optimization problem in Eq. (3) is an SILP, since both θ and γ are linearly constrained, and there are an infinite number of constraints, one for each possible value of Ψ. As in Sonnenburg *et al.* (2006), we propose to use the *column generation* technique to solve this SILP problem. In this technique, the optimal θ and γ are computed for a restricted subset of constraints in Eq. (4). This problem is called the "restricted master problem". Constraints that are not satisfied by current θ and γ are added successively to the restricted master problem until all constraints are satisfied. For fast convergence of the algorithm, it is desirable to add a constraint that maximizes the violation for current θ and γ. That is, the Ψ value that solves:

$$\Psi_\theta = \operatorname*{argmin}_{\Psi} \sum_{j=1}^{p} \theta_j S_j(\Psi), \quad (12)$$

is desired. If $\sum_{j=1}^{p} \theta_j S_j(\Psi^\theta) \geq \gamma$, then all the constraints are satisfied, and θ and γ reach their optimal values. Otherwise, this constraint is added to the restricted master problem, and the iteration continues.

It follows from the definition of $S_j(\Psi)$ in Eq. (5) that the problem in Eq. (12) can be written as:

$$\min_{\Psi} \sum_{i=1}^{k} \left\{ \frac{1}{4} \psi_i^T \psi_i + \frac{1}{4\lambda} \psi_i^T \left(\sum_{j=1}^{p} \theta_j K_j \right) \psi_i - \psi_i^T h \right\}. \quad (13)$$

For a fixed θ, the problem in Eq. (13) is an unconstrained convex quadratic program whose solution can be obtained analytically. To avoid computing the matrix inverse, we obtain Ψ by solving the following k systems of linear equations:

$$\left(\frac{1}{2} I + \frac{1}{2\lambda} \sum_{j=1}^{p} \theta_j K_j \right) \psi_i = h_i, \quad \text{for } i = 1, \cdots, k. \quad (14)$$

After Ψ is computed, the corresponding constraint is added to the restricted master problem to obtain the intermediate θ and γ. Since the restricted master problem is a linear program, the proposed algorithm for solving the SILP problem alternates between solving k linear systems and a linear program. Note that the coefficient matrix is the same for all the k linear systems, and the LU decomposition (Golub and Van Loan, 1996) needs to be computed only once. Only the forward/backward substitution needs to be performed k times to obtain the solutions.

The alternating algorithm for solving the proposed SILP problem belongs to a family of algorithms for solving general SIP problems called the *exchange methods*, in which the constraints are exchanged at each iteration. It follows from Theorem 7.2 in Hettich and Kortanek (1993) that these methods are guaranteed to converge. Similar to the convergence criterion used in Sonnenburg *et al.* (2006), the algorithm returns when $|1 - \sum_{j=1}^{p} \theta_j^{(t-1)} S_j(\Psi^{(t)}) / \gamma^{(t-1)}| \leq \epsilon$, where $\theta_j^{(t-1)}$, for $j = 1, \cdots, p$, and $\gamma^{(t-1)}$ are the optimal solutions to the restricted master problem at the $(t-1)$th iteration, $\Psi^{(t)}$ is the Ψ value that maximizes the constraint violation at the tth iteration, and ϵ is a user-specified tolerance parameter. We set $\epsilon = 5 \times 10^{-4}$ in our experiments.

Table 5. Performance of individual kernels on gene expression pattern annotation in terms of macro F1 score.

Feature	SIFT		SC		PCA-SIFT		SPIN		SF		DI		CF		MI		CC	
Grid size	16	32	16	32	16	32	16	32	16	32	16	32	16	32	16	32	16	32
Star	0.5326	0.5222	0.4659	0.4648	**0.5418**	0.5173	0.5217	0.5187	0.5279	0.5197	0.5295	0.5132	0.5319	0.5255	0.4951	0.4519	0.5294	0.5087
Clique	**0.5363**	**0.5272**	**0.4695**	**0.4718**	0.5417	**0.5194**	**0.5270**	**0.5251**	**0.5330**	**0.5233**	**0.5311**	**0.5172**	**0.5346**	**0.5319**	**0.4977**	**0.4550**	**0.5324**	**0.5123**
kCCA	0.5218	0.5140	0.4555	0.4557	0.5297	0.5079	0.5091	0.5097	0.5170	0.5094	0.5189	0.4993	0.5195	0.5164	0.4877	0.4391	0.5242	0.4990
Original	0.4619	0.4652	0.2517	0.2971	0.4765	0.4383	0.4030	0.4207	0.4610	0.4379	0.4489	0.4038	0.4703	0.4673	0.4050	0.2755	0.4640	0.4127
MIMLSVM	–	0.2760	–	0.2001	–	0.1980	–	0.3011	–	0.1922	–	0.1790	–	0.2062	–	0.1633	–	0.2729

This table shows the macro F1 scores obtained by each individual kernel for a data set of 1000 image sets and 10 terms. Total of nine local descriptors ('SIFT': SIFT descriptor; 'SC': shape context; 'PCA-SIFT': PCA-SIFT; 'SPIN': spin image; 'SF': steerable filters; 'DI': differential invariants; 'CF': complex filters; 'MI': moment invariants; 'CC': cross correlation) applied on grids of two different sizes (16 and 32 pixels) are used. The row headed with "Original" shows the performance of kernels without embedding. We compare the proposed formulations with the multi-instance multi-label method MIMLSVM proposed in Zhou and Zhang (2007). The codes are obtained from the authors, and the features are normalized to the interval $[-1, 1]$ across all bags (as suggested by the authors). The results for MIMLSVM on grid size of 16 are not available due to computational resource limitations. The performance shown in this table is the averaged scores over ten random partitions of the entire data set into training and test sets with ratio 1:1.

Table 6. Performance of individual kernels on gene expression pattern annotation in terms of micro F1 score.

Feature	SIFT		SC		PCA-SIFT		SPIN		SF		DI		CF		MI		CC	
Grid size	16	32	16	32	16	32	16	32	16	32	16	32	16	32	16	32	16	32
Star	0.5478	0.5353	0.4767	0.4721	0.5576	0.5291	0.5394	0.5355	0.5419	0.5348	0.5439	0.5275	0.5453	0.5395	0.5065	0.4654	0.5449	0.5213
Clique	**0.5526**	**0.5401**	**0.4806**	**0.4795**	**0.5585**	**0.5317**	**0.5444**	**0.5424**	**0.5471**	**0.5389**	**0.5469**	**0.5322**	**0.5482**	**0.5461**	**0.5101**	**0.4678**	**0.5482**	**0.5256**
kCCA	0.5393	0.5290	0.4685	0.4652	0.5494	0.5220	0.5304	0.5293	0.5342	0.5286	0.5365	0.5172	0.5348	0.5319	0.5012	0.4548	0.5420	0.5141
Original	0.5039	0.5065	0.3160	0.3526	0.5152	0.4824	0.4545	0.4640	0.5018	0.4867	0.4965	0.4570	0.5133	0.5107	0.4519	0.3507	0.5058	0.4651
MIMLSVM	–	0.3282	–	0.2738	–	0.2731	–	0.3489	–	0.2771	–	0.2884	–	0.3116	–	0.2394	–	0.3302

This table shows the performance of each individual kernel in terms of micro F1 score. See the footnotes of Table 5 for detailed explanations.

Performance of individual kernels

We evaluate the performance of each individual kernel on a data set of 10 terms and 1000 image sets. To incorporate the correlation among patterns sharing common embryonic structures, we apply the proposed hypergraph spectral learning framework in which patterns sharing common embryonic structures tend to be close to each other. Tables 5 and 6 show the macro and micro F1 scores obtained with kernels constructed from various local descriptors. In general, kernels constructed from global features achieve lower performance than those constructed from local features, and their results are omitted due to space constraints. By treating each feature set as a bag, the automated annotation of gene expression patterns can also be approached with multi-instance learning methods. Recently, multi-instance learning for multi-label data has been proposed by Zhou and Zhang (2007), and we also report the performance of the MIMLSVM algorithm (Zhou and Zhang, 2007) for this data set. Note that the existing approach (Zhou and Peng, 2007) can only handle the case in which terms are associated with individual images, and hence it cannot be applied to our data set.

We can observe from Tables 5 and 6 that in terms of both macro and micro F1 scores, kernels constructed by clique expansion consistently outperform other kernels. The original kernels achieve lower performance than those constructed with hypergraph. This implies that incorporation of the correlation information among patterns sharing common structures captured by the hypergraph can significantly improve the performance of kernels. The multi-instance, multi-label learning algorithm MIMLSVM achieves the lowest performance in terms of both measures.

By comparing the performance of kernels derived from different local descriptors, we can see that the kernels constructed from SIFT and PCA-SIFT achieve superior performance. This is consistent with the conclusion reached by a recent systematic evaluation that SIFT-based descriptors perform best (Mikolajczyk and Schmid, 2005). It can also be observed from Tables 5 and 6 that the spin image descriptor (Lazebnik *et al.*, 2005) also achieves competitive performance. This is consistent with the fact that the spin image descriptor is designed to capture texture information, and such information is critical in pattern discrimination. In terms of grid size, kernels constructed on a grid size of 16 pixels consistently outperform those constructed on a 32-pixel grid. However, computational resource limitations prevent us from decreasing the grid size further.