BMC Bioinformatics

BioMed Central

Methodology article

Open Access

A graphical model approach to automated classification of protein subcellular location patterns in multi-cell images

Shann-Ching Chen[1] and Robert F Murphy*[1,2]

Address: [1]Department of Biomedical Engineering and Center for Bioimage Informatics, Carnegie Mellon University, Pittsburgh, PA 15213, USA and [2]Department of Biological Sciences and Center for Automated Learning and Discovery, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Email: Shann-Ching Chen - shanncc@andrew.cmu.edu; Robert F Murphy* - murphy@cmu.edu

* Corresponding author

Published: 23 February 2006

BMC Bioinformatics 2006, **7**:90 doi:10.1186/1471-2105-7-90

Received: 28 September 2005

Accepted: 23 February 2006

This article is available from: http://www.biomedcentral.com/1471-2105/7/90

Abstract

Background: Knowledge of the subcellular location of a protein is critical to understanding how that protein works in a cell. This location is frequently determined by the interpretation of fluorescence microscope images. In recent years, automated systems have been developed for consistent and objective interpretation of such images so that the protein pattern in a single cell can be assigned to a known location category. While these systems perform with nearly perfect accuracy for single cell images of all major subcellular structures, their ability to distinguish subpatterns of an organelle (such as two Golgi proteins) is not perfect. Our goal in the work described here was to improve the ability of an automated system to decide which of two similar patterns is present in a field of cells by considering more than one cell at a time. Since cells displaying the same location pattern are often clustered together, considering multiple cells may be expected to improve discrimination between similar patterns.

Results: We describe how to take advantage of information on experimental conditions to construct a graphical representation for multiple cells in a field. Assuming that a field is composed of a small number of classes, the classification accuracy can be improved by allowing the computed probability of each pattern for each cell to be influenced by the probabilities of its neighboring cells in the model. We describe a novel way to allow this influence to occur, in which we adjust the prior probabilities of each class to reflect the patterns that are present. When this graphical model approach is used on synthetic multi-cell images in which the true class of each cell is known, we observe that the ability to distinguish similar classes is improved without suffering any degradation in ability to distinguish dissimilar classes. The computational complexity of the method is sufficiently low that improved assignments of classes can be obtained for fields of twelve cells in under 0.04 second on a 1600 megahertz processor.

Conclusion: We demonstrate that graphical models can be used to improve the accuracy of classification of subcellular patterns in multi-cell fluorescence microscope images. We also describe a novel algorithm for inferring classes from a graphical model. The performance and speed suggest that the method will be particularly valuable for analysis of images from high-throughput microscopy. We also anticipate that it will be useful for analyzing the mixtures of cell types typically present in images of tissues. Lastly, we anticipate that the method can be generalized to other problems.

Background

The location (or locations) of a protein within cells is an important attribute that can be largely independent of its structure, enzymatic activity, or level of expression. Systematic and comprehensive analysis of subcellular location is therefore needed as part of systems biology efforts to understand the behavior of all expressed proteins. Work in this area can be divided into experimental *determination* and computational *prediction*. Of course, the accuracy and utility of prediction methods is dependent on the accuracy, coverage and resolution of determination methods. This is because experimentally determined locations are the starting point for the machine learning methods at the heart of prediction systems [1-3]. Subcellular location is most frequently determined by visual interpretation of fluorescence microscope images, but such interpretations can be highly variable from observer to observer. We have therefore developed automated systems to recognize major subcellular patterns [4-6] and to learn new patterns directly from fluorescence microscope images [7,8]. These systems utilize high resolution images and have been shown to be able to distinguish similar patterns better than visual examination [9].

Automated interpretation of subcellular patterns in micrographs

The automated location determination systems consist of machine classifiers (such as neural networks or support vector machines) and sets of informative numerical features (which we term SLFs for Subcellular Location Features) to describe protein distributions in the cell. This process is illustrated in Figure 1a. The starting point is the collection of many images of two (or more) different protein patterns. Regions containing single cells are identified either automatically or manually, and background fluorescence is subtracted. A number of different types of SLF are then calculated for each cell, including morphological features that describe the number, distribution, size and shape of fluorescent objects in the cell and texture features that describe the pixel-to-pixel variation in intensity. A feature matrix is then created in which each row shows the values of each feature for a given cell along with the type (or "class") of the protein that was labeled in that cell. This matrix is used to train a classifier so that it can learn a mapping between the SLF and the classes. For each new (test) cell, the process of segmentation, background subtraction, and feature calculation is repeated, and the feature vector is supplied to the classifier to assign the cell to one of the known classes.

Using large collections of HeLa cell images containing ten distinct subcellular patterns, our systems have achieved classification accuracies as high as 92% and 98% for 2D and 3D single cell images, respectively [10,11]. The patterns of dissimilar classes can be distinguished quite well; however, there is still room to improve the classification accuracy for similar classes (such as endosomal and lysosomal proteins and different Golgi proteins).

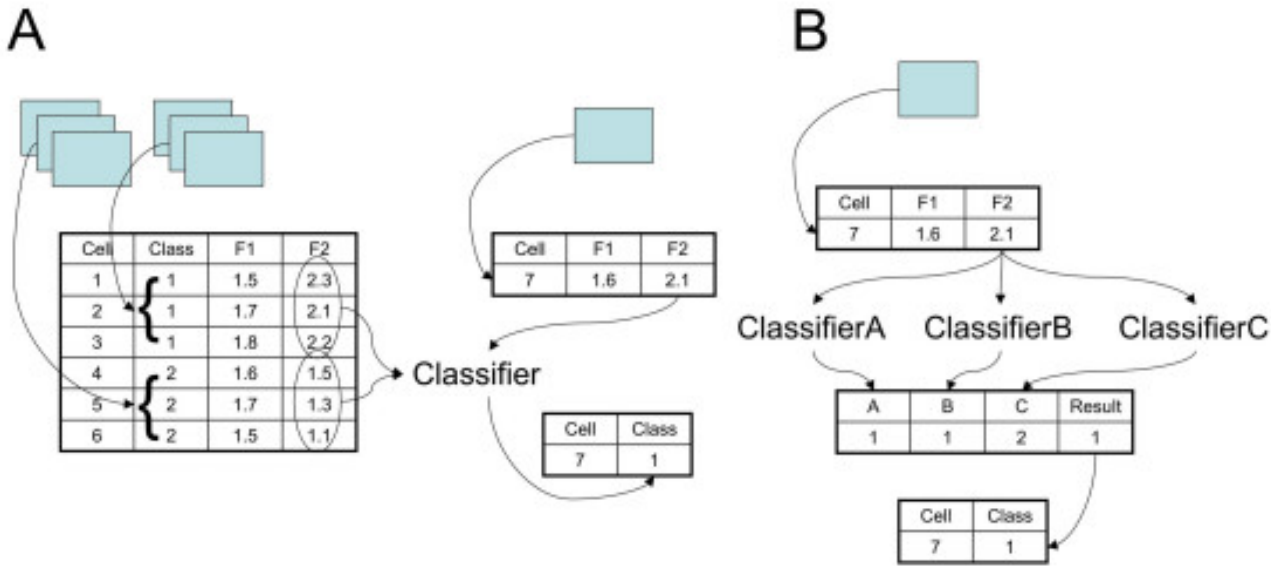

Figure 1
Illustration of classification approaches to single cells. A) Basic approach to feature-based classification of single cell images. B) Majority-voting classifier.

In order to improve the classification accuracy, one strategy is to incorporate additional or improved features and another is to combine more than one classifier using voting methods. The performance improvements we have obtained for 2D HeLa images, from 83% using a library of 84 features and a neural network classifier [6] to 92% using a library of 180 features and a majority-voting ensemble [10], resulted from implementing both of these strategies. A majority-voting ensemble combines the results from many different classifiers into a single decision, as illustrated in Figure 1b.

These improvements were obtained while considering the classification of patterns in single cells. An additional strategy is to utilize information from more than one cell from the same sample. For example, when sets of HeLa cells from the same slide were individually classified and allowed to vote for a single classification for the entire set, overall accuracy improved from 83% to 98% [6]. The penalty for this improvement is that we give up the ability to identify more than one pattern in a given set. A possible improvement on this approach is therefore to first estimate the number of classes that are present from the frequencies of the classes (by ruling out classes that have a low frequency), and then assign each cell to one of the remaining classes. (If we rule out all but one class, this approach reduces to the previous one.) So that we can decide which classes to rule out, we assume that the "true" classes are present in roughly equal proportions. In this paper, we first evaluate this simple strategy. We then describe more sophisticated approaches that construct a graphical model representing pattern information for more than one cell in a field so that improved classification accuracy can be achieved while retaining the ability to classify each cell individually (and without the assumption that classes are present in equal frequencies).

Graphical models

Graphical models have been extensively applied to problems in the computer vision field, such as image segmentation and object recognition, where the pixels in an image can be segmented or classified into two (foreground and background) or more classes [12]. Many classification problems where the labels of related objects must be consistent with each other, such as hypertext classification [13] and identification of protein functions in the protein-protein interaction network [14], can also utilize graph-based methods. To our knowledge, graphical models have not previously been applied to the recognition of subcellular patterns in multi-cell images. Large numbers of such images are increasingly being acquired both in projects aimed at determining the subcellular location of all proteins [8,15-17] and in drug screening by high-throughput microscopy [18]. Part of the motivation behind the work we describe here is the need to classify fields of cultured cells that may be expressing different tagged proteins (such fields arise when a population of cells is randomly tagged). An additional motivation is the desire to classify individual cell patterns in tissues that may consist of more than one cell type.

The problem to be solved using a graphical model is to infer the posterior probability of each class for each node (cell) using information about the likely classes of other nodes (cells). For some graphical models, an exact solution can be found using the belief propagation (BP) algorithm [19]. However, BP can only calculate the posterior probability correctly on trees or forests, that is, on graphs where there is at most one path between any two nodes. If there are loops in the graph, the junction tree algorithm [20] can be used to convert a loopy graph into a tree by clustering nodes together. Exact inference can then be achieved by applying BP on the converted tree, but the running time is exponential in the size of the largest cluster in the converted graph. We therefore need approximate inference methods for cases where the size of the largest cluster is large. A commonly used approximate method is loopy belief propagation (LBP), which iteratively applies belief propagation updates on a graph with loops. LBP often gives good approximate inference when it converges [21], and often runs very quickly, but can fail to converge on some graphs. Other approximate inference algorithms, such as variational methods [22] and Monte Carlo methods [23], are also widely used. Running times for these approximate inference methods can be prohibitive for large graphs.

A graphical model consists of an algorithm for constructing the graph itself and an algorithm for making inferences given the graph. In this paper we describe how to construct graphs for the problem of subcellular location classification, and also present a novel algorithm, which we term prior updating, that permits inferences to be made for the (often large) resulting graphs.

Results

Problem Statement: At the outset, we formalize our problem by describing our assumptions about the process used to create cell images. We assume that the process of creating a slide (or a well, plate or chamber) for imaging starts by creating a mixture of any number of cells from each of many possible classes. We further assume that cells are randomly distributed over the slide at some time t_{plate} before imaging, that the cells divide with an average generation time of t_g, and that the class of a cell is stably inherited by its daughters (the latter assumption can be relaxed slightly to allow for mutation without substantially changing our treatment). Lastly, we assume that we have accurate methods for segmenting multi-cell images into regions containing single cells, and classification methods

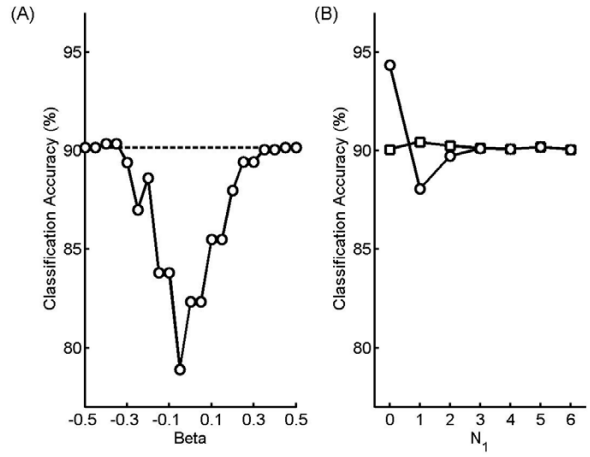

Figure 2
Classification Accuracy for simulated fields of cells using the equal-sized class model. Simulated fields consisting of N_1 cells from one class and N_2 cells from a different class were generated as described in the text, with $N_1 + N_2 = 12$. A) The average accuracy across all N_1 values are shown for the equal-sized class model (❍) as a function of the model parameter β. The accuracy of the base single cell classifier is shown by the dashed line. The best accuracy (90.4%) of the equal-sized class model is obtained when $\beta = -0.4$. B) The improvement in average classification accuracy over the base single cell classifier is shown as a function of N_1. Each point shows the average classification accuracy over 10 repeated trials for 12 cells for all possible pairs of classes for $\beta = -0.4$. The average accuracy of the equal-sized class model (❍) and base classifier (□) are shown. The classification accuracy is better than that of the base classifier only when $N_1 = 0$, the case when the set consists of just one class.

that provide a likelihood for each possible class for each segmented cell. The task is: Given an image of a field containing a number of cells meeting the assumptions above, assign a class to each cell as accurately as possible.

Equal-sized class model

As discussed above, performance of a single cell classifier on a multi-cell image can be improved if the assumption can be made that all cells in the field should show the same pattern. This can be done by assigning the most frequent class in the image to all cells [6]. While this assumption may be true in some cases, it is quite restrictive. The goal of the work in this paper is to improve the analysis of multi-cell images without the drastic assumption of homogeneity. We begin by considering a variation on this assumption, namely that each multi-cell image is composed of a small number of classes with roughly equal numbers of cells. In this case, one strategy is to decide upon the number of classes using a threshold on the observed frequencies of each class. We define $T_n = 1/(1 + n) + \beta$, where n is the number of classes and β is an adjustable parameter that ranges from -0.5 to 0.5. To find the number of classes, we find the smallest n for which the frequencies of exactly n classes are greater than T_n and record which classes those are. This definition is based on the assumption that the true classes are present in roughly equal proportion, and hence that the percentage of each should be greater than the expected frequency of a class if one more true class was present (plus a tolerance controlled by β). We consider an example to illustrate the approach. Using $\beta = 0.1$ results in $T_1 = 0.6$ and $T_2 = 0.43$. Given a field with three classes with frequencies (0.7,0.2,0.1), we would choose $n = 1$. However, if the frequencies were (0.45,0.5,0.05), $n = 2$ would be chosen. Once n is chosen, each cell in the trial field is assigned to the one of those classes that has the largest likelihood for that cell (as assigned by the single cell classifier). Note that this might not be the class with the highest likelihood if that class was not retained during the selection of the number of classes. If no n meets the criterion, we simply keep the classification results from the single cell classifier. Note that as β decreases to -0.5, we increasingly favor finding only one class, and as β approaches 0.5 we increasingly favor making no changes to the original class assignments.

Evaluation scheme

To illustrate and test approaches to multi-cell classification, we need multi-cell images in which the class of each cell is known with certainty. Since it is nearly impossible to collect such images (without, for example, using micromanipulation to spot cells on a slide), we have simulated them by combining images from a large library of single cell images (the 2D HeLa cell image collection described in the Methods). The library contains images of ten subcellular pattern classes, and to classify individual cells we have used a multi-class support vector machine classifier whose outputs were converted to probabilities for each class.

For the first tests, we created synthetic images consisting of 12 cells randomly drawn from only two classes such that the number of images in one class varied from 6 to 12. Average accuracies over 10 repeated trials were determined for the (base) single-cell classifier and for the equal-size class model described above. This process was repeated for all possible pairs of classes and for different mixtures of images from the two classes, and the average classification accuracy across all of these conditions was determined for various values of β. Figure 2a compares the overall classification accuracy across all mixtures between the base classifier and the equal-sized class model. The best average accuracy (90.4%) is obtained for $\beta = -0.4$. Figure 2b compares the classification accuracy for $\beta = -0.4$ between the base classifier and the equal-sized class

model as a function of N_1, the number of cells in one of the two classes. The classification accuracy is only better than that of the base classifier for the set consisting of only one class, but in all other cases the classification accuracies are either lower or equal. The results also indicate that cases of different mixtures need different optimal β s to achieve the best accuracy improvement (data not shown). For example, when $N_1 = 0$, the accuracy can be improved up to 9.8% over the base classifier for $\beta = -0.05$, but the average accuracy across all mixtures is much worse (78.9%). The best accuracy improvements for cases with $N_1 = (1, 5, 6)$ are (1.1%, 1.9%, 2.7%) with $\beta = (-0.15, -0.20, -0.20)$. However, for cases with $N_1 = (2, 3, 4)$, no matter how the β is tuned, the best possible average accuracy can only be the accuracy from the base classifier. This is expected since the assumption used to derive the method was that whatever classes are present are approximately equal in frequency. All these results suggest that the equal-size method should not be used when the mixture of classes is unknown.

Construction of graphical models

We next consider what information may be available about the likely class of a cell given information about its neighbors in the field, and how we can construct a graphical model to use that information. Two limit cases can be considered. These limits are based on the relative magnitudes of the constants t_{plate} and t_g defined in the problem statement above.

Feature space model

The first possibility is that t_{plate} is short relative to t_g such that cells would not have time to undergo significant cell division prior to their being imaged. In this case, the proximity of cells does not provide any information about their likely similarity (i.e., whether they are derived from the same class). The only clues that we have about the number of classes present (and the number of cells in each) are the similarities between cells in the SLF feature space. In this case, we initially construct an undirected graph in which each cell is represented by a node and edges are created between each pair of nodes with length equal to the z-scored Euclidean distance between the feature vectors of the corresponding cells.

Physical space model

If, however, the amount of time that elapses between plating and imaging is significantly greater than the generation time ($t_{plate} \gg t_g$), each original cell is expected to have divided a number of times and we may consider it likely that the class of cells adjacent to one another is the same. The rate (v_{trans}) at which daughter cells move away from each other relative to the rate at which they divide becomes the determining factor. Thus, if v_{trans} is high, we may consider physical proximity to be of little predictive value and are forced to use the feature space model described above. If, on the other hand, v_{trans} is low, we can construct an undirected graph using the Euclidean distance between the centers of cells in the field.

Pruning

Initially, the graphs for both model types are fully connected. Each edge suggests the two nodes it connects should influence each other's labels. Since we can assume that they should not influence each other if the distance between them is too large (and to improve computational efficiency), edges whose length is greater than a free parameter d_{cutoff} are removed. Note that the units of d_{cutoff} are different for the two types of models.

Inference by prior updating

Given a graphical model of either of these types, the task becomes inferring the class labels. This requires an algorithm to describe how the label at each node is influenced by information from each of its nodes. As described in the introduction, exact and even approximate inference methods can be extremely compute intensive for models with many connected nodes. Since our goal is to apply graphical models to fields with many cells, we need an efficient method for inferring the most likely class for each cell given the results of the single cell classifier for it and its neighbors. We therefore developed a new method, which we term prior updating (PU), that we believed could give improved classification accuracy in realistic compute times. The principle behind the method is simple: we allow each cell to have its own set of prior probabilities for all possible classes and adjust them to reflect the likely classes of its neighbors. We start by setting all prior probabilities equal, and then determine the posterior probability of each class for each cell using the output of the SVM classifier and Bayes rule. We then iteratively adjust the prior probabilities of all classes for each cell based on the labels of its neighbors and recalculate the posterior probabilities (Figure 3). A free parameter α controls the extent to which the prior probabilities are adjusted at each iteration (for $\alpha = 0$, no adjustment is made). The method terminates when no class labels change during an iteration. Each cell is allowed to change its label at most once, and its confidence is set to zero after the label changes. We designed this strategy because cells whose labels are easily changed are expected to have high uncertainty, and should not influence other cells after their labels change. This strategy also guarantees that the iteration will converge in constant time. Similar results are obtained if priors for each node are initialized outside the loop and if labels are allowed to change more than once (data not shown).

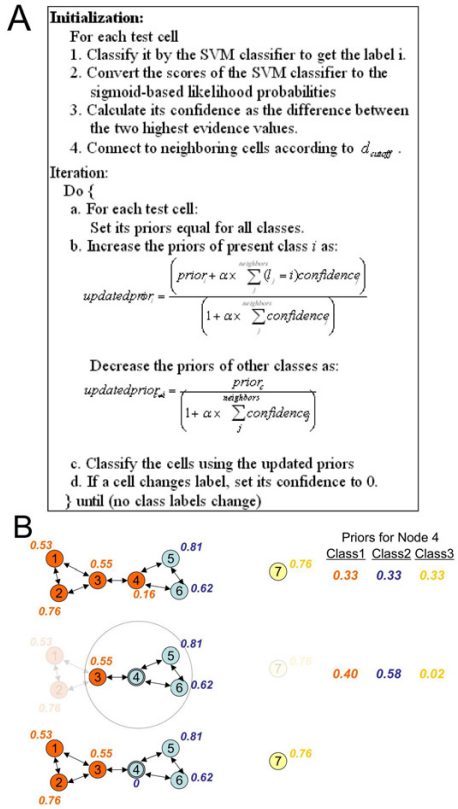

Figure 3
The prior updating algorithm. A) Pseudo-code for the algorithm is shown. A free parameter α in the updating equation is used to determine the degree of change of priors. When α is zero, the priors do not change and the graphical model results are the same as the results of the base classifier. The priors are pushed harder to the majority classes in the field as α increases. B) Illustration of the PU algorithm for a graphical network of seven cells and three classes.

Feature space model

To evaluate the accuracy of the graphical models and the prior updating method, we used synthetic multi-cell images as described above. We first consider the feature space model, which is directly comparable to the equal-size scheme since neither considers physical position of a cell. We calculated classification accuracy for various values of the two free model parameters: α and d_{cutoff}. Figure 4 shows results for fields of 6 cells each for two classes for the best d_{cutoff} for each of various values of α. The best results were obtained with $\alpha = 0.15$ and $d_{cutoff} = 8$. We evaluated three metrics: overall accuracy (average of all 10 classes), average accuracy for similar classes (the endosomal and lysosomal proteins and the two Golgi proteins), and accuracy for dissimilar classes (the remaining classes). Compared with the results for the base classifier (without inference), the accuracy of similar classes is much improved (by 9 percentage points, from 82.2% to 91.3%), and the accuracy of dissimilar classes is also improved (by 3 percentage points, from 95.3% to 98.5%). The overall accuracy is improved by over 5 percentage points (from 90.1% to 95.7%). The overall accuracy of 95.7% obtained with an SVM classifier combined with PU is higher than the best previous accuracy for the 2D HeLa collection of 92.3%, which was obtained using a much more complicated majority-voting classifier [10].

When α is zero, the priors are not updated so that cells do not influence each other. As α increases, the priors of classes that are present in the field are increased while others are decreased. As seen in Figure 4, classification accuracy also increases as α increases but roughly plateaus at α near 0.2. The results suggest that a large α usually gives good improvement in classification accuracy; however, the best α has to be found by applying cross-validation methods.

The d_{cutoff} parameter is designed to determine the neighbors of a cell. If d_{cutoff} is very small, the cell does not have any neighbors to influence and be influenced by. As d_{cutoff} gets larger, the cells start to be influenced by other similar cells, and so the classification accuracy can be improved.

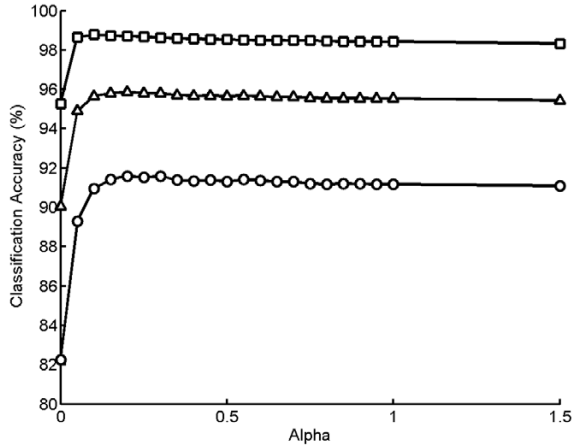

Figure 4
Improvement of classification accuracy using feature space graphical model. Each point shows the average classification accuracy over 10 repeated trials for 12 cells for fields of six cells each from two classes. The average accuracy for pairs of similar classes (❍), dissimilar classes by (□), and all classes (△) are shown. The best accuracies are obtained with $\alpha = 0.15$ and $d_{cutoff} = 8$. The accuracy of similar classes is improved by 9% (from 82.2% to 91.3%), while the accuracy of dissimilar classes is also improved 3% (from 95.3% to 98.5%). The overall accuracy is improved by the prior updating method by over 5%(from 90.1% to 95.7%).

If d_{cutoff} is set to infinity, all the cells are connected to each other in the graph and so contribute to the updates of each other's priors. In this case, some dissimilar cells will affect each other's priors and the classification accuracy could be worse than when the best d_{cutoff} is used. The best d_{cutoff} can be found by applying cross-validation methods.

Encouraged by these results, we evaluated trial fields with two classes of varying numbers of cells in the feature space field (Figure 5). For the $N_1 = 0$ case, where there is only one class of cells present in the field, the best d_{cutoff} and α are both infinite, so that all the cells can be classified into one class just as the equal-sized class scheme does. The best d_{cutoff} is 8 for all other cases. This implies that the *z*-score distances among similar cells of 2D HeLa images in the SLF16 feature space are on average less then 8, no matter how many cells the classes are composed of. The best α ranged from 0.2 to 0.5 for different cases (data not shown). The results in Figure 5 were obtained with α set to 0.5, and this value was used for all subsequent experiments. As the sizes of the two classes become more asymmetric (from $N_1 = 6$ to $N_1 = 2$ case), the accuracy improvement of similar classes still remains in the range of 8 to 9 percentage points, while the accuracy improvement of dissimilar classes decreases from 1 to 3 percentage points. This is because smaller numbers of "minority" classes affect the estimated priors to a lesser degree, and a small change in priors is more likely to affect the labels of similar classes than of dissimilar ones. For the $N_1 = 0$ and $N_1 = 1$ case, the accuracy of similar classes are higher than for the other cases, which confirms that it is easier to determine which of similar classes a cell is more likely to be when the cells are more homogeneous in the field.

Physical space model

We also evaluated a model in which the physical positions of cells in the field are used to influence classification. Synthetic fields of cells were created by simulating the processes of cell division and movement for clones derived from two cells of different classes initially at a distance *D* from each other. Figure 6 shows results for applying the graphical models on fields generated with various values of *D*. When $D = 0$, the two clones overlap in space but in most cases, the accuracies for similar and dissimilar

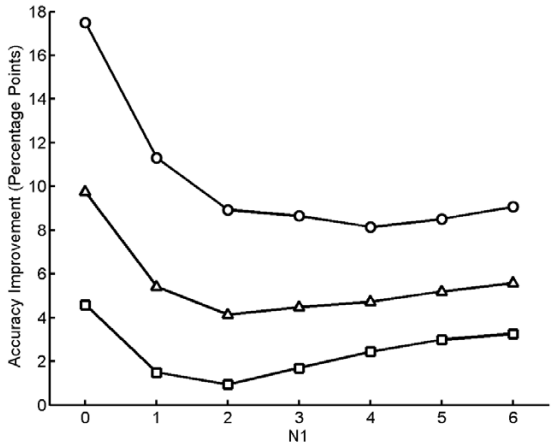

Figure 5
Improvement in classification accuracy for simulated fields of cells using a feature space graphical model. Simulated fields consisting of N_1 cells from one class and N_2 cells from a different class were generated as described in the text. A class label was assigned to each cell in the simulation using the feature space graphical model described in the text. The improvement in average classification accuracy over the base single cell classifier is shown as a function of N_1, where $N_1+N_2 = 12$. Each point shows the average classification accuracy over 10 repeated trials for 12 cells for all possible pairs of classes. The average accuracy for pairs of similar classes (❍), dissimilar classes by (□), and all classes (△) are shown. Results except for $N_1 = 0$ are for a d_{cutoff} value of 8, the best value of those tested.

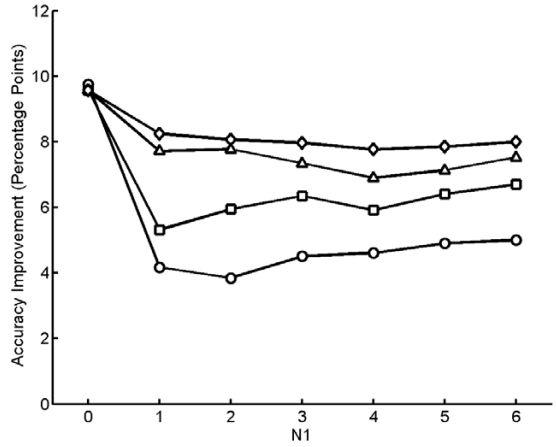

Figure 6
Improvement in classification accuracy for simulated fields of cells using a physical space graphical model. Simulated fields containing clones of cells consisting of N_1 cells from one class and N_2 cells from a different class were created as described in the text for various values of D, the distance between the initial cells of each class. A class label was assigned to each cell in the simulation using the physical space graphical model described in the text. The improvement in average classification accuracy over the base single cell classifier is shown as a function of N_1, where $N_1+N_2 = 12$. Each point shows the average classification accuracy over 10 repeated trials for 12 cells for all possible pairs of classes for fields generated with D = 0 (❍), D = 6 (□), D = 12 (△), and D = 400 (◇). Results except for $N_1 = 0$ are for a d_{cutoff} value of 6, the best value of those tested. Note that, as expected, the accuracy improves with increasing D.

classes are still improved over the base classifier. This is expected, since this case is very similar to the feature space model evaluated above. The classification accuracy improves as the separation of the two clones increases ($D > 0$), also as may be expected. The results demonstrate the important conclusion that our graphical models can result in significant improvement in classification accuracy for the task of classifying a mixed population of cells under a variety of test conditions.

Multiple classes test

Given that our method performed well for multiple cells from two different classes, we next examined the cases where more than two classes of cells were present. We therefore performed experiments using the feature space model on one to five classes with each class having six cells. The results are shown in Table 1. As the number of classes increases, the overall accuracy decreases from around 9 percentage points *above* to around 2 percentage points *below* that of the base classifier. Since it is more likely that there are cells from both of two similar classes in the field as the number of classes increases, this is expected. The observation that the transition from improvement to degradation occurs after 4 out of 10 classes are present loosely suggest that the maximum number of classes that can be simultaneously present in a field and still see improvement from a graphical model is around 40% of the number of possible classes.

Effect of training set size

We also examined the effect of training set size on the prior updating scheme as a way of examining the improvement possible for a less accurate base classifier. Various training set sizes were used to train base classifier SVMs and then these were applied to fields of two classes of equal sizes. The results in Table 2 show that the base SVM classifier decreases its accuracy with fewer training data (as expected), but that the prior updating scheme can still improve its accuracy by between 5 and 8 percentage points. The smaller the amount of the training data, the more the prior updating method can improve and compensate the accuracy. It is especially impressive that the combination of the prior updating scheme and the SVM classifier with only 10 training data per class can do a similar job to the SVM classifier alone with 50 training data per class. The results also indicate that at least for this subcellular pattern classification task, the SVM classifier joined with the prior updating scheme does not need a lot of training data in order to attain a fair classification performance.

Discussion

Our work has particular implications for classification of patterns in images obtained by high-throughput microscopy. Since high-throughput systems typically use low magnification, the number of cells per field is often high and the accuracy of single-cell classifiers is usually not perfect. By applying this method on multi-cell images made of real single cells and synthesized locations, we are able to verify that our scheme can be used for such systems to achieve significantly better performance.

Since we have proposed a new approximate inference algorithm, it is important to identify when this method works better than other approximate inference methods. This method is very fast compared to previously described graphical model algorithms: its runtime is linearly proportional to the number of cells in each trial field and to the number of classes it needs to choose from. Whether this method has better classification performance under different circumstances will be examined in future work. We anticipate that the method can be made more general so that it can be used for other applications, both for biomedical applications like classification of cell types in tissue images and for other applications like Internet link analysis.

Conclusion

This paper addresses a supervised learning problem in the domain of protein subcellular location determination. We have proposed a novel graphical representation where multiple cells in a field influence each other. Assuming

Table 1: Results for multiple classes.

	Classification Accuracy (%)		
No. of classes	Similar Classes	Dissimilar Classes	All Classes
1	96.7	99.8	98.6
2	91.3	98.5	95.6
3	86.4	97.3	92.9
4	82.0	96.0	90.4
5	78.2	94.8	88.1
Base Classifier	82.2	95.3	90.1

The accuracy of classification using feature space graphical models were evaluated for all possible mixtures composed of various numbers of classes drawn from ten classes. Results shown are averages over 10 trials.

Table 2: Results for different training set sizes.

No. of training data	Classification Accuracy (%)					
	Similar Classes		Dissimilar Classes		All Classes	
	No updating	With updating	No updating	With updating	No updating	With updating
50	82.2	91.3	95.3	98.5	90.1	95.6
40	80.8	90.2	94.9	98.3	89.2	95.1
30	78.9	88.9	94.2	98.4	88.1	94.6
20	76.3	87.5	93.2	98.0	86.4	93.8
10	71.2	80.8	90.6	96.6	82.9	90.3

Base SVM classifiers were trained using various numbers of cells for each of ten classes. Graphical models using those base classifiers were then tested on fields containing six cells each from two classes and results were averaged over 10 trials each for all possible pairs.

that these cells are only composed of a small number of classes, the classification accuracies are improved by manipulating the prior distributions of classes. The improvement is largest for groups of classes which would be difficult for the base classifier to distinguish from one another.

We have also shown the robustness of our prior updating scheme. The accuracies for different classes were always improved under different assumptions about the distribution of cells in the field, different sizes of the two classes of cells present in the field, different numbers of classes, and different training set sizes.

The results are very encouraging since the prior updating method improves the overall accuracy from the base classifier by around 5 percentage points and the accuracy of similar classes by around 9 percentage points. The combination of the prior updating method and the base single cell classifier outperforms the majority voting classifier that with an accuracy of 92.3% had the best prior reported performance on this dataset [10].

Methods

2D HeLa cell image collection

The 2D HeLa cell image collection was created by introducing antibodies and molecular probes against proteins in major subcellular organelles [6]. This data set contains 862 single-cell images consisting of ten classes, each of which contains from 73 to 98 images. Figure 7 shows typical images from each class. Every image has a resolution of 382 × 512 pixels and each pixel represents 0.23 × 0.23 μm in the sample plane. In parallel to each protein image, an image of the DNA distribution was obtained using a DNA-specific fluorescent probe. These parallel images provide a common reference framework for describing the distribution of each protein.

Subcellular Location Features (SLF)

We have developed several sets of informative features to describe protein subcellular patterns. These features, termed Subcellular Location Features (SLFs), are of several types, including Zernike moment features, Haralick texture features, morphological features and wavelet features. The details for different versions of SLFs are reviewed in [10]. The best classification results obtained to date for the 2D HeLa dataset were with feature set SLF16 [10], and we have therefore used the SLF16 feature set in this work. Each cell in the dataset is thus represented by a feature vector x of length $d = 47$.

Bayesian decision theory

Bayesian decision theory is a fundamental statistical approach to pattern classification problems [24]. The Bayes formula can be expressed as:

$$p(w_j \mid x) = \frac{p(x \mid w_j)p(w_j)}{p(x)}$$

where w_j is the class with index j, $p(w_j)$, termed the prior probability, is the probability of class j being observed in the absence of any other information, $p(x \mid w_j)$, termed the likelihood probability, is the probably density function for an observed feature vector x given that the class is w_j, $p(w_j \mid x)$, termed the posterior probability, is the probability of the class being w_j given that x has been observed, and $p(x)$, termed the evidence, is just a normalization to guarantee that the posterior probabilities sum to one. For n classes, the evidence can be formulated as

$$p(x) = \sum_{j=1}^{n} p(x \mid w_j)p(w_j).$$

A probabilistic classifier assigns an observation x to class i if

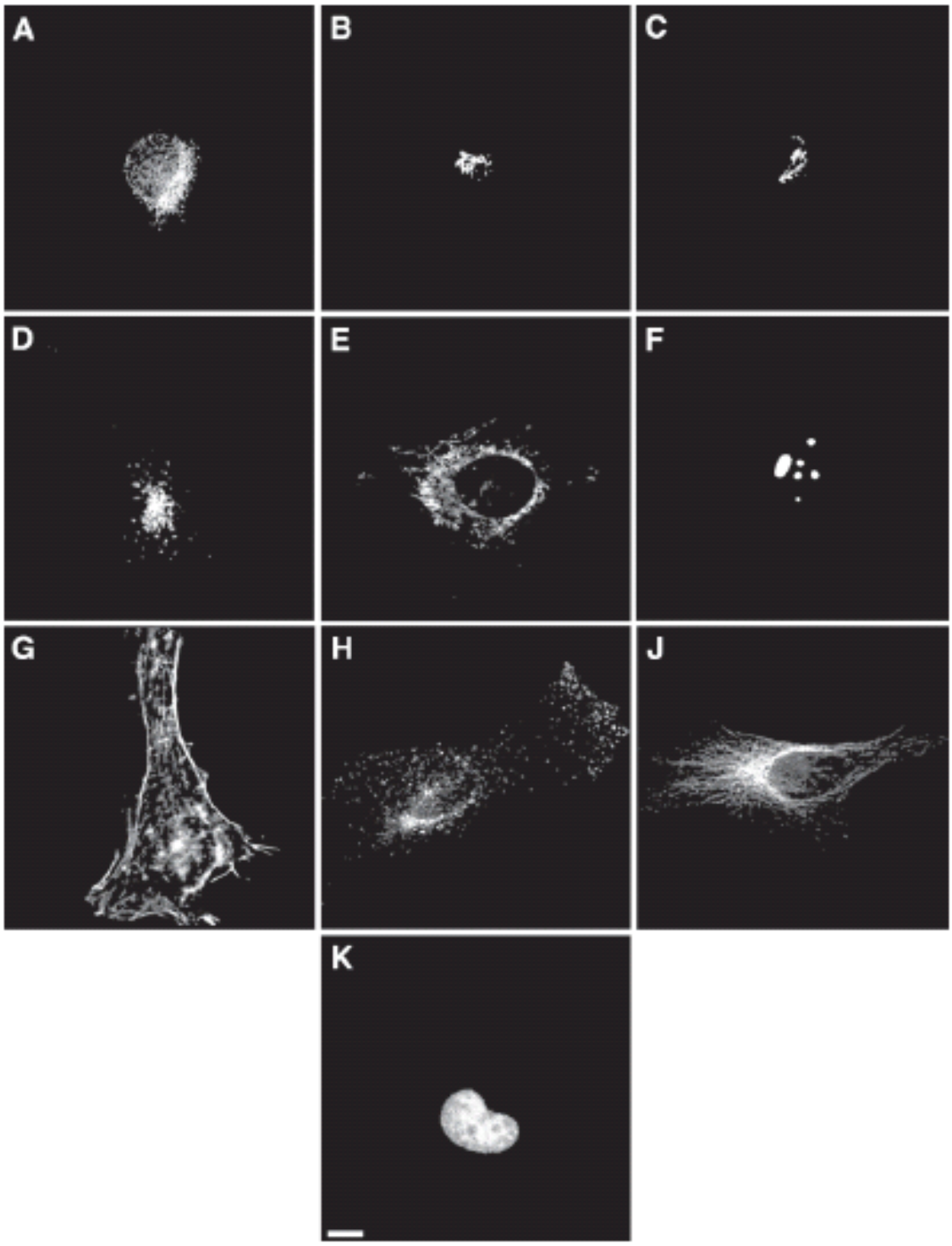

Figure 7
Typical images from the 2-D HeLa cell image collection used in this study. Images are shown for cells labelled with antibodies against an ER protein (A), the Golgi protein giantin (B), the Golgi protein GPP130 (C), the lysosomal protein LAMP2 (D), a mitochondrial protein (E), the nucleolar protein nucleolin (F), transferring receptor (H), and the cytoskeletal protein tubulin (J). Images are also shown for filamentous actin labelled with rhodamine-phalloidin (G) and DNA labelled with DAPI (K). Scale bar = 10 μm. From [6].

$$p(w_i \mid x) > p(w_j \mid x) \quad \forall j \neq i$$

That is, the classifier assigns x to the class with the maximum posterior probability.

In our previous work, each cell was classified independently. Since the priors were not known in advance, they were assumed to be equal. In this case, the classification with the "Maximum a Posteriori Probability" (MAP) is equivalent to the "Maximum Likelihood" (ML).

Classifier – Support Vector Machine

Support Vector Machines (SVM) were originally designed for binary classification by finding a maximum margin hyperplane between two classes [25]. They can be extended to solve multi-class classification problems by combining several binary classifiers. There are several commonly used methods, such as one-against-all, one-against-one, and directed acyclic graph. Here we adapt the one-against-all method [26,27], which constructs n SVM classifiers where n is the number of classes. The i_{th} SVM is trained using all of the examples in the i_{th} class with positive labels and all others with negative labels. The test example is fed into these n SVMs and the one with the highest output score is selected as the final class. Each SVM used an exponential radial basis function kernel with $C = 20$ and $\sigma = 7$, where C mediates the trade-off between maximizing the margin and minimizing the training error, and σ is the parameter in the expression:

$$K(x, y) = e^{\frac{-(x-y)^2}{2\sigma^2}}$$

The kernel function K is a distance function for two feature vectors x and y. The multi-class SVM produces uncalibrated scores that are expected to be positively correlated with the confidence of the assignment but which are not directly comparable between classes. Thus, we use a sigmoid function to calibrate the output scores of the SVM. The parameters of the function can be found by minimizing the negative log likelihood of the training data [28]. The resulting probabilities are then comparable between different classes. We associate with each node an evidence vector consisting of the probabilities for each class and a label corresponding to the class with largest evidence. The confidence of this label is defined as the difference between the two highest class probabilities.

Creation of synthetic multi-cell images

To synthesize multi-cell images, we used the 2D HeLa image set composed of 10 classes of major subcellular location patterns (described above). To meet the assumptions that cells are only composed of a small number of classes, we constructed trial fields consisting of cells drawn from all possible pairs of the 10 classes in the 2D HeLa dataset. For each trial, N_1 and N_2 cells were randomly picked from two different classes with total number of 12 cells. Separate trials were conducted for N_1 from 0 to 6.

For cross-validation, we split the data into five folds: one fold for the testing pool and the other four folds for the training pool. In the training pool, 50 images from each class were randomly chosen and for each trial, N_1 and N_2 cells were randomly picked from all possible pairs of classes out of the testing pool. Each of the five folds was in turn used for testing and the remaining four for training a multi-class SVM classifier. The classification accuracies were averaged for each pair of classes over all five folds. Some of the images are used neither for training nor for testing in any one fold, but the testing images may be used more than once overall due to lack of data. Because of this reuse, this evaluation method is similar to the usual five-fold cross validation procedure but not the same. In expectation it will report the correct accuracy for the classifier, but the variance of its reported accuracy is difficult to compute. To reduce this variance as much as possible we average 10 trials by randomly assigning images in the testing and training pools.

Since the 2D HeLa images were originally collected for single cells without recording their position on the slide, we simulated the positions of the cells according to a simple model of cell growth and movement. The pseudocode of the simulation is shown in Figure 8. Once the positions were simulated, a randomly-chosen cell from a specified class of the HeLa images was assigned to each position. To simulate the presence of more than one class on a slide, two (or more) simulated clones from different classes were generated with a separation parameter D representing the distance between their origins. An example for two clones of six cells each is shown in Figure 9, with edges drawn between cells that are less than 6 units apart (i.e., $d_{cutoff} = 6$).

Code availability

The data and source code used for the work described in this paper is available from http://murphylab.web.cmu.edu/software.

Authors' contributions

SCC participated in the design of, and carried out, all experiments, and drafted the manuscript. RFM conceived of the approach, participated in its design and coordination, and helped extensively with writing the manuscript. Both authors read and approved the final manuscript.

Acknowledgements

We thank Geoffrey Gordon for helpful discussions and critical reading of the manuscript. This work was supported in part by NIH grant R01

Begin with one cell placed at the origin of the 2D plane.
Do {
For each cell:
Duplicate it with waiting time $t_d = \frac{e^{-\lambda}}{\lambda_1}$ and
place the daughter at distance $d_{child} = u[d_1, d_1]$
Move it with waiting time $t_m = \frac{e^{-\lambda_2}}{\lambda_2}$ and
translation $d_{trans} = u[d_2, d_2]$
} until (the number of cells is N)

Figure 8
Algorithm for simulating cell fields. The algorithm simulates the formation of a clone of *N* cells from a single cell and incorporates cell growth and movement. *u[d,d]* represents a two dimensional uniform distribution from *-d* to *d* (e.g., a cell can move to anywhere within the square with length of the side equals to *2d*). d_1 and d_2 describe how much cells spread apart after cell division. t_d corresponds to the average generation time of t_g, and t_m indicates the average time a cell moves. If the d_1 and d_2 are the same, large t_d and small t_m will result a more compact colony, while small t_d and large t_m will result a sparser colony.

GM068845, NSF grant EF-0331657, and a research grant from the Commonwealth of Pennsylvania Tobacco Settlement Fund.

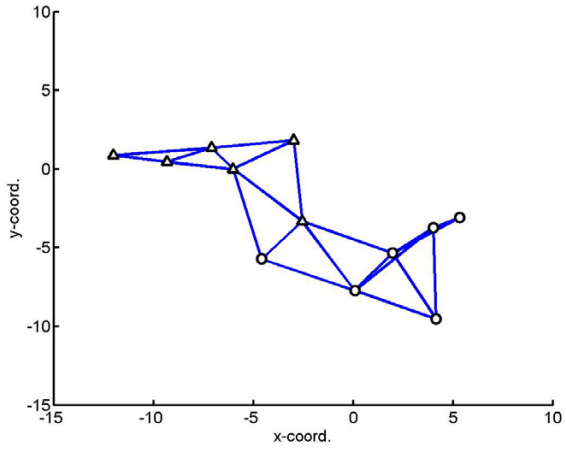

Figure 9
Simulation of cell positions for two classes. Two simulated clones from different classes were generated with a separation parameter *D* defining the distance between the initial cell positions. An example of the distribution for two simulated clones of six cells each is shown for D = 12. Edges connect cells that are less than 6 units apart. Note that some of these edges connect cells from different classes.

References

1. Park KJ, Kanehisa M: **Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs.** *Bioinformatics* 2003, **19(13):**1656-1663.
2. Chou KC, Cai YD: **Prediction and classification of protein subcellular location-sequence-order effect and pseudo amino acid composition.** *J Cell Biochem* 2003, **90(6):**1250-1260.
3. Lu Z, Szafron D, Greiner R, Lu P, Wishart DS, Poulin B, Anvik J, Macdonell C, Eisner R: **Predicting subcellular localization of proteins using machine-learned classifiers.** *Bioinformatics* 2004, **20(4):**547-556.
4. Boland MV, Markey MK, Murphy RF: **Automated recognition of patterns characteristic of subcellular structures in fluorescence microscopy images.** *Cytometry* 1998, **33(3):**366-375.
5. Murphy RF, Boland MV, Velliste M: **Towards a Systematics for Protein Subcellular Location: Quantitative Description of Protein Localization Patterns and Automated Analysis of Fluorescence Microscope Images.** *Proc Int Conf Intell Syst Mol Biol* 2000, **8:**251-259.
6. Boland MV, Murphy RF: **A Neural Network Classifier Capable of Recognizing the Patterns of all Major Subcellular Structures in Fluorescence Microscope Images of HeLa Cells.** *Bioinformatics* 2001, **17(12):**1213-1223.
7. Chen X, Murphy RF: **Objective Clustering of Proteins Based on Subcellular Location Patterns.** *J Biomed Biotechnol* 2005, **2005(2):**87-95.
8. Chen X, Velliste M, Weinstein S, Jarvik JW, Murphy RF: **Location proteomics - Building subcellular location trees from high resolution 3D fluorescence microscope images of randomly-tagged proteins.** In *Proc SPIE Volume 4962*. San Jose, CA, U. S. A. ; 2003:298-306.
9. Murphy RF, Velliste M, Porreca G: **Robust Numerical Features for Description and Classification of Subcellular Location Patterns in Fluorescence Microscope Images.** *J VLSI Sig Proc* 2003, **35(3):**311-321.
10. Huang K, Murphy RF: **Boosting Accuracy of Automated Classification of Fluorescence Microscope Images for Location Proteomics.** *BMC Bioinformatics* 2004, **5:**78.
11. Chen X, Murphy RF: **Robust Classification of Subcellular Location Patterns in High Resolution 3D Fluorescence Microscopy Images.** In *Proc 26th Intl Conf IEEE Eng Med Biol Soc* San Francisco, CA ; 2004:1632-1635.
12. Felzenszwalb PF, Huttenlocher DP: **Efficient Belief Propagation for Early Vision.** *Proc 2004 IEEE Conf on Computer Vision Pattern Recognition* 2004, **1:**261-268.
13. Taskar B, Abbeel P, Koller D: **Discriminative Probabilistic Models for Relational Data.** *Uncertainty in Artificial Intelligence* 2002:485-492.
14. Vazquez A, Flammini A, Maritan A, Vespignani A: **Global protein function prediction from protein-protein interaction networks.** *Nat Biotechnol* 2003, **21(6):**697-700.
15. Kumar A, Agarwal S, Heyman JA, Matson S, Heidtman M, Piccirillo S, Umansky L, Drawid A, Jansen R, Liu Y, Cheung KH, Miller P, Gerstein M, Roeder GS, Snyder M: **Subcellular localization of the yeast proteome.** *Genes Develop* 2002, **16(6):**707-719.
16. Ghaemmaghami S, Huh WK, Bower K, Howson RW, Belle A, Dephoure N, O'Shea EK, Weissman JS: **Global analysis of protein expression in yeast.** *Nature* 2003, **425(6959):**737-741.
17. Conrad C, Erfle H, Warnat P, Daigle N, Lorch T, Ellenberg J, Pepperkok R, Eils R: **Automatic Identification of Subcellular Phenotypes on Human Cell Arrays.** *Genome Res* 2004, **14(6):**1130-1136.
18. Perlman ZE, Slack MD, Feng Y, Mitchison TJ, Wu LF, Altschuler SJ: **Multidimensional Drug Profiling by Automated Microscopy.** *Science* 2004, **306(5699):**1194-1198.
19. Pearl J: **Probabilistic Reasoning in Intelligent Systems.** Morgan Kaufmann; 1988.
20. Huang C, Darwiche A: **Inference in belief networks: a procedural guide.** *Intl J Approximate Reasoning* 1996, **15(3):**225-263.
21. Murphy K, Weiss Y, Jordan M: **Loopy Belief Propagation for Approximate Inference - an Empirical Study.** *Uncertainty in Artificial Intelligence* 1999:467-475.
22. Jordan MI, Ghahramani Z, Jaakkola TS, Saul LK: **An Introduction to Variational Methods for Graphical Models.** *Machine Learning* 1998, **37(2):**183-233.

23. Mackay DJC: **Introduction to Monte Carlo methods.** In *Learning in graphical models* Cambridge, MA , MIT Press; 1998:175-204.
24. Duda RO, Hart PE: **Pattern Classification and Scene Analysis.** New York , John Wiley & Sons; 1973:482.
25. Cortes C, Vapnik V: **Support vector networks.** *Machine Learning* 1995, **20:**1-25.
26. Vapnik V: **Statistical Learning Theory.** New York City , Wiley; 1998.
27. Hsu CW, Lin CJ: **A comparison of methods for multi-class support vector machines.** *IEEE Transactions on Neural Networks* 2002, **13:**415-425.
28. Platt J: **Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods.** *Advances in Large Margin Classifiers, MIT Press* 1999:61-74.