BIOINFORMATICS ORIGINAL PAPER

Vol. 26 no. 13 2010, pages 1630–1636
doi:10.1093/bioinformatics/btq239

Systems biology

Advance Access publication May 19, 2010

Automated analysis of protein subcellular location in time series images

Yanhua Hu[1,2], Elvira Osuna-Highley[1,3], Juchang Hua[1,2,4], Theodore Scott Nowicki[1,2], Robert Stolz[5], Camille McKayle[5] and Robert F. Murphy[1,2,3,4,6,*]

[1]Center for Bioimage Informatics, [2]Department of Biological Sciences, [3]Department of Biomedical Engineering, [4]Department of Machine Learning, Carnegie Mellon University, Pittsburgh, PA 15213, [5]Division of Science and Mathematics, University of the Virgin Islands, St Thomas, VI 00803 and [6]Lane Center for Computational Biology, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Associate Editor: Burkhard Rost

ABSTRACT

Motivation: Image analysis, machine learning and statistical modeling have become well established for the automatic recognition and comparison of the subcellular locations of proteins in microscope images. By using a comprehensive set of features describing static images, major subcellular patterns can be distinguished with near perfect accuracy. We now extend this work to time series images, which contain both spatial and temporal information. The goal is to use temporal features to improve recognition of protein patterns that are not fully distinguishable by their static features alone.

Results: We have adopted and designed five sets of features for capturing temporal behavior in 2D time series images, based on object tracking, temporal texture, normal flow, Fourier transforms and autoregression. Classification accuracy on an image collection for 12 fluorescently tagged proteins was increased when temporal features were used in addition to static features. Temporal texture, normal flow and Fourier transform features were most effective at increasing classification accuracy. We therefore extended these three feature sets to 3D time series images, but observed no significant improvement over results for 2D images. The methods for 2D and 3D temporal pattern analysis do not require segmentation of images into single cell regions, and are suitable for automated high-throughput microscopy applications.

Availability: Images, source code and results will be available upon publication at http://murphylab.web.cmu.edu/software

Contact: murphy@cmu.edu

Received on November 20, 2009; revised on April 3, 2010; accepted on April 27, 2010

1 INTRODUCTION

Subcellular distribution is an important characteristic of a protein because location is intimately related to its function. The most common method used to find a protein's subcellular location is to fluorescently tag the protein, take images by microscopy and then visually analyze the images. Previous work has shown that automatic image analysis can outperform visual examination, providing higher sensitivity to subtle differences (Murphy *et al.*, 2003). Proteins from major subcellular locations can be differentiated from each other with over 90% accuracy (Huang and Murphy, 2004). However, previous work has been done using static images and the challenge now is to extend these approaches to time series images that better reflect protein behavior in living cells. One objective is to distinguish proteins that have similar static but different temporal patterns. This is of obvious biological importance since many proteins change their location over time in order to carry out their functions. For example, helicases localize to the nucleus during the G_1 phase of the cell cycle to repair endogenous DNA damage, and exit the nucleus during S phase (Gu, 2004).

For automated analysis of protein subcellular locations, machine learning tools for classification and clustering are well established (Glory and Murphy, 2007). A key component is feature design: good features are numerical representations of the patterns that capture the differences between classes or clusters. The goal for temporal pattern analysis is therefore to design good features from time series images that distinguish different protein movements.

Automated analysis of biological time series images has received significant attention in recent years, but most movies are of low resolution at the organ or cell level (Liebling *et al.*, 2006; Souvenir *et al.*, 2008), and only a few deal with high resolution microscopy images that record protein movement within a single cell (Danuser and Waterman-Storer, 2006; Sigal *et al.*, 2006; Zhou *et al.*, 2009). We have carefully studied applicable theories and algorithms from the computer vision field and adopted/designed five sets of temporal features for application to subcellular pattern analysis for microscopy images.

The first, and perhaps most straightforward, approach to characterizing temporal behavior is to track individual objects in a cell and calculate features to describe these tracks. For example, speed, the most common numerical description of movement, requires measuring the total distance the object travels along its way. However, some proteins do not form rigid objects and it is therefore unclear what the targets for tracking should be. The best example is cytoplasmic proteins: even when they show changing fluorescence distributions, there is typically no easily characterized movement of objects. Such 'complex and nonrigid' motion that has statistical regularity was defined as 'temporal texture' (Nelson and Polana, 1992). In that work, the normal flow on each pixel was used

[*]To whom correspondence should be addressed.

to represent movement along the image gradient and compared to uniform direction flow. The concept of ‘temporal texture’ has also been applied to adapt Haralick texture features for static images to movies (Bouthemy and Fablet, 1998) or to develop a series of features based on temporal slices (Ngo *et al.*, 2002). We therefore chose co-occurrence based temporal texture features (Hu *et al.*, 2006) and normal flow features as our second and third temporal feature sets. To design other temporal features without defining an entity for tracking, we have also explored image intensity changes in the frequency domain (Fourier transform features) and analyzed static feature changes over time (autoregression features).

We evaluated the ability of these feature types to improve our ability to discriminate protein patterns in a collection of 4D images (three spatial dimensions over time) for 12 cell lines expressing fluorescently tagged proteins.

2 METHODS

2.1 Image acquisition

NIH 3T3 cell lines previously generated as part of a proteome-scale tagging project were used in this study (Chen *et al.*, 2003; Garcia Osuna *et al.*, 2007; Jarvik *et al.*, 2002). In each cell line, a particular protein is tagged by CD tagging (Jarvik *et al.*, 1996) which introduces a green fluorescent protein (GFP) tag into the gene coding for the protein using self-inactivating retroviral vectors.

Cells were plated on glass-bottomed culture dishes in Dulbecco’s modified Eagle’s media. After 48 h, the media were changed to Opti-MEM to avoid the interference by phenol red and provide pH stability. Three-dimensional movies of GFP tagged proteins in 3T3 cells were taken by a spinning disk confocal microscope. The imaging system consists of a LaserPhysics Reliant 100 s 488 Argon laser, a Yokogawa CSU10 Confocal Scanner Unit and an Olympus IX50 microscope with a 60× 1.4NA objective. Images (1280 × 1024) were collected with a Roper Scientific/Photometrics CoolSnap HQ Cooled CCD camera, with a final resolution of 0.11 microns per pixel in the sample plane. A stack of 15 images was collected using a 3 s exposure time and a spacing of 0.5 μ between slices. Stacks were taken every 45 s, with the total period of time for each movie varying depending on the extent of fluorescence photobleaching. Before starting acquisition for each field, the focus was manually adjusted using the GFP fluorescence channel to ensure that cells were centered in the stack.

Time series images were taken for 12 cell lines, each having a different protein labeled with GFP. The proteins (genes) were as follows: cytochrome b-5 reductase (dia1) and annexin A5 (anxa5) in cytoplasm, serum deprivation response protein (sdpr) in vesicles and cytoplasm, adipose differentiation-related protein (adfp) in vesicles, ADP-ATP translocase 23 (timm23), ATP synthase (atp5a1) and mitochondrial stress-70 protein (hspa9a) in mitochondria, catalase (cat) in mitochondria and vesicles, glucose transporter 1 (glut1) in plasma membrane and t-complex testis expressed 1 (tctex1), alpha-actinin-4 (actn4), caldesmon 1 (cald1) in cytoskeleton. Figure 1 illustrates their 2D static patterns and Figure 2 shows a sample movie for ATP synthase. The full movies are available as described in the Abstract.

2.2 Image preprocessing

Background fluorescence was removed by subtracting the most common pixel value in the image. This was based on the assumption that an image contains more pixels outside the cell than inside it and that background is roughly uniform, both of which hold true for our images. Then a threshold was chosen by an automated method (Ridler and Calvard, 1978), and pixels below the threshold were set to zero. For simplicity, thresholded images were used to calculate all features, although thresholding is not required for calculating texture features. No segmentation was performed because no nuclear channel image was available to provide a reference for automatic segmentation algorithms.

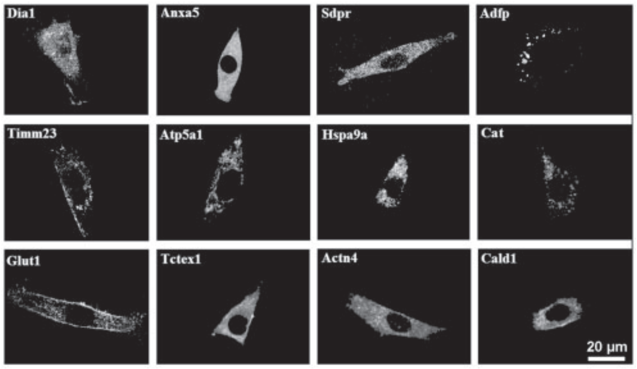

Fig. 1. Sample fluorescence microscope images of 12 GFP-tagged proteins. Each image is the most typical center slice at the first time point among all 3D time series images of the protein, selected by TypIC (Markey *et al.*, 1999) which finds the image closest to the mean in feature space. Images are preprocessed with background removal and thresholding. For display purposes, only a rectangular region containing a single cell is shown.

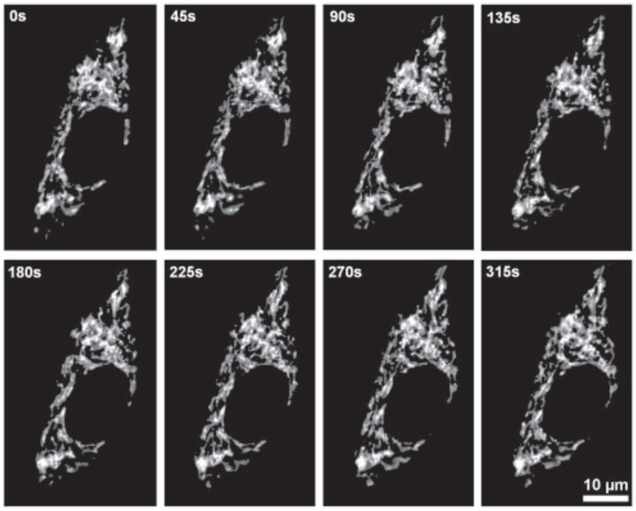

Fig. 2. Movie of ATP synthase. The first 8 center slices across 315 s are shown. Images are preprocessed by background removal, thresholding and photobleaching correction. For display purposes, only a rectangular region containing a single cell is shown.

A major problem with time series images is that photobleaching can occur over time. To minimize the influence of photobleaching, pixel intensities of images at each time point were stretched so that the 95% quantile (and above) was set equal to 256. We chose to stretch the 95% quantile instead of the brightest pixel, because this is less sensitive to noise from artifacts such as fluorescent debris. Figure 3 illustrates the extent of bleaching and correction. The original intensity (dotted line) dropped to 50% after 10 time points. After photobleaching correction, the total intensity (solid line) at the 10th time point is about 80% of the original. Although the correction cannot fully restore the total intensity, it largely reduces the influence of photobleaching. The other advantage of stretching the intensity to a fixed maximum for all proteins is that effect of protein abundance or brightness of images is largely removed.

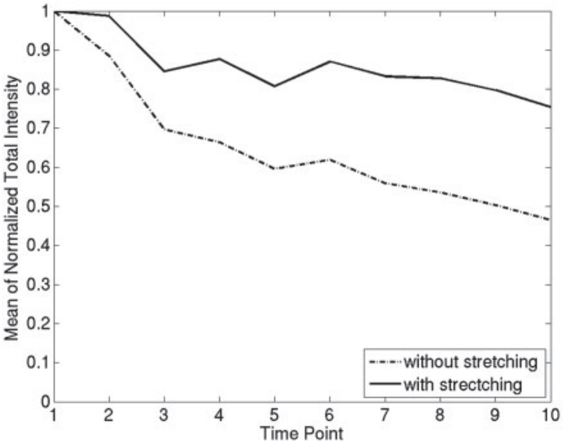

Fig. 3. Comparison of total pixel intensity changes with (solid line) versus without (dotted line) photobleaching correction. Intensities are normalized to one at the initial time point.

2.3 2D and 3D static feature calculation

In order to set the reference for comparison, both 2D and 3D static features were calculated. 2D Static features were calculated using the center slice at the first time point. Twenty-one of the features from SLF21 (Huang and Murphy, 2004) were used. They include 8 morphological (SLF21.3-SLF1.5 and SLF21.9-SLF21.13) and 13 texture features (SLF21.66-SLF21.78) that are based on the whole image field and do not depend on cell segmentation. Similarly, 3D static features were calculated by using the 3D image at the first time point. Thirty-three features that also do not require cell segmentation were chosen from SLF11 (Chen *et al.*, 2003). They include 5 morphological, 2 edge and 26 texture features.

2.4 2D Temporal feature calculation

We collected 3D time series images, in order to have as much information as possible on protein behavior. However, we started our exploration with 2D temporal patterns, by using the center slices over time. We used only the first eight time points in each movie, limited by the shortest movie in our image collection. The total period analyzed is therefore 6 min.

2.4.1 Object tracking features Calculating features based on object tracking requires several steps. First, we identify all the objects in the image at each time point, and then we compare and match these objects from one image to the next. After solving the position of each object in different time points, we have the trajectories of objects and we design features based on these object trajectories.

Objects are defined as continuous pixels that are above threshold. Over time, objects change their positions, shapes and total intensities. The assumption of tracking is that although one object might change its properties, the change is small. The similarity between an object at time point t_1 and itself in the next time point t_2 is stronger than the similarity between any other objects at t_1 with this object at t_2. Features are designed to describe properties of objects and used to calculate similarities. They are as follows:

Object center in the x coordinate (X)

Object center in the y coordinate (Y)

Object size in pixels (S)

Object total intensity (I)

The distance between the object j at time point t_1 and the object k at time point t_2 is calculated by the distance of their features:

$$\sqrt{\left(\frac{X_j-X_k}{\mathrm{std}(X_{t_1}-X_{t_2})}\right)^2+\left(\frac{Y_j-Y_k}{\mathrm{std}(Y_{t_1}-Y_{t_2})}\right)^2+\left(\frac{S_j-S_k}{\mathrm{std}(S_{t_1}-S_{t_2})}\right)^2+\left(\frac{I_j-I_k}{\mathrm{std}(I_{t_1}-I_{t_2})}\right)^2}$$

The distance is a normalized Euclidean distance so that features with large values do not dominate the result. Distances were calculated between all objects in each frame and all objects in the next frame, but object pairs that exceeded a distance threshold of 4 were discarded.

The next step is to compare the distance between all object pairs across neighboring images, and find the matching with the minimal sum of distances. Assume there are $N1$ objects in the image at t_1, and $N2$ objects in the image at t_2. $N1$ and $N2$ are not necessarily equal because objects can appear or disappear due to movement. The matching is a bipartite matching or an assignment problem: $N1$ people available for $N2$ jobs with different costs, where each person can do only one job and each job only needs one person. The best assignment is the one that minimizes the total cost. The naive way to compare different matchings is to calculate total distance for all possible matches, the computation of which is the factorial of the minimum of $N1$ and $N2$. We used the Hungarian Algorithm, a classic algorithm that reduces the computation to complexity $O((N1+N2)^3)$ (Kuhn, 1955).

After matching we know the position of each object in every time point and we can build a trajectory of each object. From these, we calculated:

Speed: the distance traveled over adjacent time points, divided by the time interval.

Velocity: the distance between an object's position when it first appears and its position when it disappears (or at the end of the movie), divided by the total time. Velocity differs from speed in that it only reflects the absolute distance that the object has traveled. If the object travels back and forth around the same area, the speed is high but velocity is low.

Acceleration: the change of speed over adjacent time points, divided by the time interval.

Angles between successive movements: the angle (0–180°) between the two directions the object travels in adjacent images.

We used the mean and variance of these four features across the time series as the final features of the image (eight in total).

2.4.2 Temporal texture features Temporal texture features were calculated as described previously (Hu *et al.*, 2006). These were inspired by the Haralick texture features, which capture the intensity correlation of neighboring pixels in space (Haralick, 1979), and designed to capture the value correlation of neighboring pixels in time. We build the temporal gray-level co-occurrence matrix, whose element in row i and column j is the frequency that a pixel with value i changes to value j in the same position at the next time point. Co-occurrence matrices capture information of protein movement, for example, for proteins that show no movement between the two time points, the temporal co-occurrence matrix contains only zeros except on the diagonal.

We calculated 13 statistics described by Haralick (Haralick, 1979) based on the co-occurrence matrix for each pair of images separated by a certain time interval. We used the mean and variance across the time series as the final features of the image. By varying the time interval, we collected 130 features in total: means and variances of the 13 statistics for images 45, 90, 135, 180 and 225 s apart.

2.4.3 Normal flow features In a series of images taken over time, the intensity of each pixel in each image I (x, y, t) is a function of position x, y and time t. To understand the movement, we need to infer from the observed I (x, y, t), the direction and velocity of each pixel. We use u to represent the vector of movement in the x direction and v to represent the vector of movement in the y direction.

$$u=\mathrm{d}x/\mathrm{d}t \text{ and } v=\mathrm{d}y/\mathrm{d}t$$

Optical flow is a motion field with vectors (u, v) at each pixel. From the definition, we know (u, v) is pointing to the direction that the pixel moves and the length of the vector is the speed. Let us consider a pixel I $(x, y, t-1)$ that moves to a new position at t. We here assume that the brightness of this pixel does not change:

$$I(x,y,t-1) = I(x+u(x,y), y+v(x,y), t)$$

We further assume that the movement is not large. So the right-hand side formula can be approximated with first order:

$$I(x,y,t-1) \approx I(x,y,t) + u(x,y)\frac{\partial I}{\partial x} + v(x,y)\frac{\partial I}{\partial y}$$

Rearranging the equation we have:

$$u(x,y)\tfrac{\partial I}{\partial x} + v(x,y)\tfrac{\partial I}{\partial y} + [I(x,y,t) - I(x,y,t-1)] = 0$$
$$uI_x + vI_y + I_t = 0$$

This is the fundamental equation of motion. I_x and I_y are the components of the gradient, which, together with the difference of the intensity I_t, can be directly calculated from the two images. However we are left with one equation with two unknowns (u and v). This is the aperture problem: motion along the edge or perpendicular to the gradient can never be recovered. Meanwhile the motion (u, v) along the gradient can be calculated:

$$u = \frac{-I_t I_x}{I_x^2 + I_y^2},\ v = \frac{-I_t I_y}{I_x^2 + I_y^2}$$

and the speed of a pixel along the gradient is

$$U_n = \frac{I_t}{\sqrt{I_x^2 + I_y^2}}$$

The flow field containing flows at the gradient direction is called normal flow. Using it, the following 34 features were calculated:

1–13	Haralick texture features of U_n
14	Mean of U_n
15	SD of U_n
16	Mean/SD of U_n
17–29	Haralick texture features of binned direction
30	Difference of direction from uniform distribution
31	Mean positive divergence
32	Mean negative divergence
33	Mean positive curl angular velocity
34	Mean negative curl angular velocity

For the texture features of direction, directions were binned into eight groups. We used 1–8 to represent vectors within the range of 0–44°, 45–89°, 90–134°, 135–179°, 180–224°, 225–269°, 270–314° and 315–360°, respectively. Divergence and curl angular velocity are defined as:

$$\mathrm{div} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \quad \mathrm{cav} = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$$

These 34 features were then computed across all images separated by a specific interval, and the mean and variance across the series were used as the final features. This process was carried out for spacing of 45 s (adjacent time points), 90 s (every other time point), 135 s, 180 s and 225 s. The result is 34*2*5 = 340 features in total.

2.4.4 Fourier transform features The Fourier analysis is the classic approach to studying signal changes over time. Studying signals in the frequency domain is commonly used because of its convenience for signal processing: complicated convolution in the spatial domain is just simple multiplication in the frequency domain. The Fourier transform returns a spectrum where each coefficient corresponds to the strength of a specific frequency in the input signal.

To calculate the Fourier transform features for time series images, each pixel at the same position over time for the first eight time points was considered as one time series signal. The first five Fourier coefficients were collected and eight basic statistics of each coefficient over the whole image were calculated as the final temporal features (40 in total).

1	5% percentile of non-zero coefficients
2	25% percentile of non-zero coefficients
3	50% percentile of non-zero coefficients
4	75% percentile of non-zero coefficients
5	95% percentile of non-zero coefficients
6	Mean of non-zero coefficients
7	SD of non-zero coefficients
8	Mean/SD of non-zero coefficients

2.4.5 Autoregression features Three properties that define temporal texture have been described: 'complex and nonrigid', 'indeterminate spatial and temporal extent' and 'statistical regularity' (Nelson and Polana, 1992). A direct way to calculate statistical regularity is to compute the characteristics of images over time and build a regression model. A linear spatiotemporal autoregressive model based on pixel intensities has been used to recognize and build temporal textures (Szumme and Picard, 1996). Instead of modeling pixel value changes, we chose to model the features of static images over time. There are several advantages of doing this: features are less affected by noise in pixel values, the feature vector (21) is much shorter than the number of pixels (1280*1024), and combined with our understanding of the static features, the nature of the motion can be better appreciated.

An autoregressive (AR) model depends only on the previous outputs of the system. The following equation describes a linear AR model for feature X at time t as a function of feature X at earlier time points:

$$X_t = c + \sum_{i=1}^{P} \varphi_i X_{t-i} + \varepsilon_t$$

where c is a constant, P controls how many time points in the past are used in modeling and ε is estimation error. The ψ_i are the parameters we seek to estimate.

We choose to vary P-values from 2 to 4, yielding 2, 3 and 4 ψ values. AR was done on each of the 21 2D static features, resulting in (2 + 3 + 4)*21 = 189 features.

2.5 3D temporal feature calculation

We expanded the three sets of 2D temporal features to 3D temporal features. Again, only the first 8 time points in each movie were used.

3D temporal texture features were calculated in a similar way as for 2D, except voxels instead of pixels were used when building co-occurence matrices. A total of 130 features that are based on mean and variance of 13 statistics for five different time intervals, the same as for 2D temporal texture features, were calculated.

3D normal flow features are defined similarly to 2D. Normal flow on x, y and z directions can be calculated with the following equations:

$$u = \frac{-I_t I_x}{I_x^2 + I_y^2 + I_z^2} \quad v = \frac{-I_t I_y}{I_x^2 + I_y^2 + I_z^2} \quad w = \frac{-I_t I_z}{I_x^2 + I_y^2 + I_z^2}$$

where I_t is the voxel intensity difference across time points and I_x, I_y and I_z are the gradient projected on the x ,y and z directions. The speed of a voxel along the gradient is

$$U_n = \frac{I_t}{\sqrt{I_x^2 + I_y^2 + I_z^2}}$$

Thirty-three normal flow features were calculated for each pair of 3D images, 16 of which are based on U_n, 13 based on Haralick texture features of binned directions and 4 based on divergence and curl (extending the definitions above to three dimensions). A total of 330 features, including mean and variance of features for five different time intervals, were calculated, similarly to the 2D normal flow features.

3D Fourier transform features were calculated similar as in 2D, except that voxels at the same position over time were considered as one time series signal. Total of 40 features including 8 statistics of 5 coefficients, same as 2D, were calculated.

2.6 Feature selection

The large numbers of features described above could potentially overwhelm the ability of a classifier to identify meaningful decision boundaries. Therefore, we used stepwise discriminant analysis (SDA) to select the features that have the greatest power to discriminate the classes. While many feature selection methods have been described, SDA has performed well in previous studies of subcellular pattern classification (Huang *et al.*, 2003).

2.7 Supervised learning (Classification)

A well established SVM package LIBSVM (Chang and Lin, 2001) was used for classification. LIBSVM uses the pair-wise algorithm to classify multiple classes and it has support for choosing optimal parameters. Ten-fold cross-validation was used to evaluate the classification accuracy: the images for each class were randomly divided into 10 groups of equal sizes. Nine of them were used to train a classifier and the remaining one was used for testing. Both feature selection and parameter tuning were done on the training set: SDA selected features were used to adjust the parameters until the classifier reached the best accuracy within the training set. The parameters were the radius of the radial basis function kernel (which was varied from 2^{-5} to 2^3) and the penalty for overlap between two classes (which was varied from 2^{-5} to 2^{15}). After training, the features for the test images were supplied to the classifier, and prediction was made on each image. By comparing the prediction with the class labels of the testing set, we obtain the classification accuracy. This process was repeated for each test fold, and the final overall accuracy is the average accuracy from the 10 tests. Since SVM can inherently handle a large number of correlated features without overfitting, we also repeated each classification without SDA feature selection.

3 RESULTS

By comparing classification accuracy using individual feature sets and combinations of them, we achieved two goals: evaluating the power of each feature set and finding the best way of differentiating our 12 proteins of interest. Three-dimensional time series images are 4D images. Feature sets we have acquired are based on 2D (2D static features of first center slice), 3D across z (3D static features), 3D across time (2D temporal features of center slices) and 4D images (3D temporal features).

3.1 Classification using 2D static features

To set a reference for comparison, classification results of proteins in 3T3 cells were obtained using 21 2D static features. The overall accuracy was 63% with SDA feature selection and 66% without SDA. The confusion matrix without SDA is shown in Table 1. We can see that Cald1 has the lowest classification accuracy (25%).

3.2 Classification using 2D temporal feature sets

To evaluate each temporal feature set, we trained classifiers using it with and without the static feature set. When object tracking features were combined with the 2D static features, the average classification accuracies were 61 and 68%, with or without SDA, showing a very small improvement over static pattern classification.

From previous work we know when calculating static Haralick texture features, one can change the gray level or resize the image to different resolution and get different classification results (Murphy *et al.*, 2003). Since the same effect would be expected in the temporal domain, we evaluated temporal texture features calculated using different pixel sizes and numbers of gray levels. Classification accuracies are summarized in Table 2. Accuracies fall within the range of 52–71% with SDA feature selection and 66–77% without SDA. The best accuracy of 77% was obtained for 64 gray levels and no downsampling. Apparently, accuracy is higher without SDA, and SVM inherently handled redundancy of features. To further test the power of temporal texture features, the same classification procedure was done without static features. The overall

Table 2. Summary of classification accuracy of 21 static combined with 130 temporal texture features, with different gray level and resolution

Resolution (micron/pixel)	Gray level		
	16	64	256
0.11	64, 77	71, 77	70, 76
0.22	68, 70	63, 70	67, 68
0.66	64, 67	58, 69	52, 69
1.1	66, 67	62, 70	64, 66

The number before comma is the accuracy in percentage with SDA feature selection, and the number after comma is the accuracy in percentage without SDA.

Table 1. Confusion matrix for classification using only 2D static features

True classes	Prediction by classification											
	Dia1	Anxa5	Sdpr	Adfp	Timm23	Atp5a1	Hspa9a	Cat	Glut1	Tctex1	Actn4	Cald1
Dia1	45	0	35	5	5	0	5	0	5	0	0	0
Anxa5	5	61	0	0	0	0	0	0	0	0	33	0
Sdpr	26	4	56	0	4	0	0	0	0	0	8	0
Adfp	0	1	0	90	1	1	0	3	0	0	0	0
Timm23	2	2	2	2	72	0	0	12	2	0	2	0
Atp5a1	10	0	5	0	0	65	10	0	0	0	0	10
Hspa9a	4	0	0	4	8	4	70	0	0	0	8	0
Cat	0	0	0	0	43	6	6	37	6	0	0	0
Glut1	0	0	5	11	11	11	0	5	52	0	0	0
Tctex1	0	6	0	0	0	0	0	0	0	93	0	0
Actn4	0	20	3	3	0	0	3	0	0	3	58	6
Cald1	0	0	0	0	6	12	12	0	0	6	37	25

The overall accuracy is 66%.

Table 3. Confusion matrix for the best classification accuracy of 12 protein patterns

True classes	Prediction by classification											
	Dia1	Anxa5	Sdpr	Adfp	Timm23	Atp5a1	Hspa9a	Cat	Glut1	Tctex1	Actn4	Cald1
Dia1	60	0	15	0	5	10	5	0	5	0	0	0
Anxa5	0	83	0	0	0	0	0	0	0	0	16	0
Sdpr	4	0	95	0	0	0	0	0	0	0	0	0
Adfp	0	0	0	96	0	1	0	0	0	0	1	0
Timm23	0	0	0	2	65	0	7	12	12	0	0	0
Atp5a1	10	0	5	10	0	75	0	0	0	0	0	0
Hspa9a	0	0	4	0	12	0	79	0	0	0	4	0
Cat	0	0	0	6	50	6	0	37	0	0	0	0
Glut1	11	0	0	0	11	11	0	0	64	0	0	0
Tctex1	0	0	0	0	0	0	0	0	0	100	0	0
Actn4	0	6	0	3	0	0	0	0	0	0	82	6
Cald1	0	0	0	0	0	0	6	0	0	6	31	56

Features used are 2D static, temporal texture, normal flow and Fourier transform features. The overall accuracy is 78%.

classification accuracy was 67% with SDA and 66% without SDA. The results show that temporal texture features themselves capture a significant amount of information.

We next evaluated the normal flow features. Three hundred and forty normal flow features were combined with the 2D static features. Since the normal flow features include texture features calculated on U_n and the normal flow direction, a similar exploration of number of gray levels and resolution was done. Classification accuracy ranged from 66% to 75% with SDA and 59% to 73% without SDA (data not shown). The best accuracy of 75% was achieved with the original resolution (0.11 μm/pixel) and 64 gray levels, with SDA. When the same classification procedure was done without static features, the overall classification accuracy was 75% with SDA and 74% without SDA. This indicates that these features were more informative than temporal texture features.

When 40 Fourier transform features were combined with the static features, the overall classification accuracy was 69% with SDA and 67% without SDA, a slight increase over using static features alone (data not shown). When using Fourier transform features only, the overall accuracy was 60 and 57% with and without SDA (data not shown).

We next combined 189 autoregression features with the static features. The overall classification accuracy was 59% with SDA and 48% without SDA (data not shown). These are both lower than using static features alone.

From this analysis of each temporal feature set, we found that temporal texture features, normal flow features and Fourier transform features are capable of increasing classification accuracy. We combined these 3 sets of temporal features to search for the best way to differentiate the 12 proteins in 2D time series images. If combined with 2D static features, the overall accuracy is 75% with SDA and 78% without SDA. With temporal features only, very good classification accuracy of 73 and 77% (with and without SDA) can be achieved, proving again the power of the temporal features. The confusion matrix of the best classification is shown in Table 3. Comparing the results with Table 1, the overall accuracy is 12% higher than using just 2D static features. Tctex1 was now classified perfectly. The lowest accuracy of 25% for cald was increased to 56%, and accuracy for Sdpr increased from 56% to 95%. Two other proteins, Anxa5 and Actn4, also have over 20% improvement in accuracy. While the accuracy with SDA selection was lower than without, it is interesting to note that SDA selected 7 static, 15 temporal texture, 13 normal flow and 4 Fourier transform features, from the total of 531 features. This suggests that all of these feature types provide useful information.

3.3 Classification using 3D static features

Static features based on 3D images capture information of images with one more dimension than 2D images. Since we were not able to perform automatic segmentation due to lack of nuclear channel, we selected those features that do not compare to center of cells, thus do not require segmentation. Classification accuracy using 33 field level 3D static features was 74% with SDA, and 71% without SDA. It is 6% higher than using 2D static features and 4% lower than using combined 2D static and temporal features. When 3D static and 2D static features were combined, classification accuracy was 73 and 70%, with and without SDA. As might be expected, the 3D features apparently contain all of the information in the 2D features, and adding redundant 2D static features only made classifier learning more difficult.

3.4 Classification using 3D temporal feature set

Similar to the way we tested 2D temporal features, we combined 3D temporal features with 3D static features to see if the classification accuracy was improved. The result is disappointing. Classification accuracies with or without SDA for 3D temporal textures were 74 and 75%, for 3D normal flow were 71 and 72% and for 3D Fourier transform were 73 and 71%. Only 3D temporal texture features, combined with 3D static features, performed slightly better than using 3D static features alone. None of the accuracies were higher than the accuracy we achieved combining 2D temporal features with 2D static features. It appears that calculating features over the full 3D images diluted the critical information in the central slices.

4 CONCLUSION AND DISCUSSIONS

Location proteomics as a branch of proteomics study has grown over the last 10 years. To systematically analyze large amounts of data, automatic algorithms have been developed. This article presented our effort to extend the analysis of static patterns with an additional dimension: the temporal domain. Time series microscopy image of 12 proteins were collected, 5 sets of 2D temporal features and 3 sets of 3D temporal features were implemented and classifications were performed to validate their usefulness. The best 2D temporal feature sets, in the order of their ability to improve classification accuracy, were normal flow, temporal texture and Fourier transform features. Combining 2D static features with 3 sets of 2D temporal feature sets gave the best accuracy of 78%, compared with 66% for static features alone. Accuracy using 3D static and/or temporal features was lower than for 2D features.

If limited acquisition time requires deciding whether to collect 3D static images or 2D time series images, our results suggest that 2D time series images have higher potential of delivering better differentiation. Although not all of the 2D or 3D temporal feature sets improved classification accuracy for our dataset, we still presented them here because they may be useful for future datasets.

While each protein in a proteome is unique, its location patterns might not be. Thus while increasing accuracy of distinguishing the 12 proteins is an indication of feature value, the ability to distinguish them all perfectly is not expected. If proteins interact or colocalize with each other, they cannot be differentiated either by static or temporal pattern. In our dataset, alpha-actinin-4 (actn4) and caldesmon 1 (cald1) both bind to actin, and over 30% of caldesmon 1 is misclassified as alpha-actinin-4. These two proteins are always observed in the same cluster using cluster analysis (data not shown). Similarly, ADP-ATP translocase 23 (timm23) and catalase (cat), which have both been described as mitochondrial proteins, are difficult to distinguish (50% of catalase is misclassified as ADP-ATP translocase 23). The results suggest that there are only 10 distinguishable patterns in our 12 protein set, and this agrees with prior knowledge about these proteins.

For each classification, we compared accuracy with or without SDA feature selection, because SVM is known to be highly robust with large number of correlated features. Our result shows when the number of features is large, 340 normal flow and 189 autoregression, SDA outperforms no feature selection. When the number of features is small, 21 static and 130 temporal textures, SVM does well without SDA feature selection. Many different feature selection algorithms and classifiers could be tried in order to achieve higher classification accuracy, but such an analysis is beyond the scope of this study.

Since temporal texture features and Fourier transform features can be calculated within 25 and 50 s for each time series, they are readily applicable to many high throughput applications. On the other hand, calculating normal flow features, object tracking features and autoregression features take 8, 22 and 2 min per time series, respectively.

Given the dramatic increase in automated microscopy over the past decade, we anticipate that methods for analyzing temporal changes in protein patterns such as those we have described here will be of significant utility both for basic research in systems biology and for drug screening and development purposes.

ACKNOWLEDGEMENTS

This publication does not necessarily reflect the views of the National Science Foundation.

Funding: National Science Foundation (grants EF-0331657 to R.F.M.; HBCU-UP grant HRD-9979896 to LaVerne Ragster); National Institutes of Health (grants R01 GM075205 to R.F.M. U54 RR022241 to Alan Waggoner; and MBRS-RISE R25 GM061325 to Teresa Turner); National Science Foundation's IR/D program (to C.M., a Program Officer from 2005 to 2008).

Conflict of Interest: none declared.

REFERENCES

Bouthemy,P. and Fablet,R. (1998) Motion characterization from temporal cooccurrences of local motion-based measures for video indexing. In *International Conference on Pattern Recognition (ICPR'98)*, pp. 905–908.

Chang,C.-C. and Lin,C.-J. (2001) LIBSVM: a library for support vector machines.

Chen,X. *et al.* (2003) Location proteomics - Building subcellular location trees from high resolution 3D fluorescence microscope images of randomly-tagged proteins. *Proc. SPIE*, **4962**, 298–306.

Danuser,G. and Waterman-Storer,C.M. (2006) Quantitative fluorescent speckle microscopy of cytoskeleton dynamics. *Annu. Rev. Biophys. Biomol. Struct.*, **35**, 361–387.

Garcia Osuna,E. *et al.* (2007) Large-scale automated analysis of location patterns in randomly tagged 3T3 cells. *Ann. Biomed. Eng.*, **35**, 1081–1087.

Glory,E. and Murphy,R.F. (2007) Automated subcellular location determination and high throughput microscopy. *Dev. Cell* **12**, 7–16.

Gu,J. *et al.* (2004) Cell cycle-dependent regulation of a human DNA helicase that localizes in DNA damage foci. *Mol. Biol. Cell*, **15**, 3320–3332.

Haralick,R.M. (1979) Statistical and structural approaches to texture. *Proc. IEEE*, **67**, 786–804.

Hu,Y. *et al.* (2006) Application of temporal texture features to automated analysis of protein subcellular locations in time series fluorescence microscope images. In *2006 IEEE International Symposium on Biomedical Imaging*, pp. 1028–1031.

Huang,K. and Murphy,R.F. (2004) Automated classification of subcellular patterns in multicell images without segmentation into single cells. In *2004 IEEE International Symposium on Biomedical Imaging*, pp. 1139–1142.

Huang,K. and Murphy,R.F. (2004) Boosting accuracy of automated classification of fluorescence microscope images for location proteomics. *BMC Bioinformatics*, **5**, 78.

Huang,K. *et al.* (2003) Feature reduction for improved recognition of subcellular location patterns in fluorescence microscope images. *Proc. SPIE*, **4962**, 307–318.

Jarvik,J.W. *et al.* (1996) CD-Tagging: a new approach to gene and protein discovery and analysis. *BioTechniques*, **20**, 896–904.

Jarvik,J.W. *et al.* (2002) In vivo functional proteomics: mammalian genome annotation using CD-tagging. *BioTechniques*, **33**, 852–867.

Kuhn,H.W. (1955) The hungarian method for the assignment problem. *Naval Res. Logistic Quart.*, **2**, 83–97.

Liebling,M. *et al.* (2006) Nonuniform temporal alignment of slice sequences for four-dimensional imaging of cyclically deforming embryonic structures. In *IEEE International Symposium on Biomedical Imaging*. Arlington, VA, pp. 1156–1159.

Markey,M.K. *et al.* (1999) Towards objective selection of representative microscope images. *Biophys. J.*, **76**, 2230–2237.

Murphy,R.F. *et al.* (2003) Robust numerical features for description and classification of subcellular location patterns in fluorescence microscope images. *J. VLSI Sig. Proc.*, **35**, 311–321.

Nelson,R.C. and Polana,R. (1992) Qualitative recognition of motion using temporal texture. *CVGIP Image Understanding*, **56**, 78–89.

Ngo,C.W. *et al.* (2002) Motion retrieval by temporal slices analysis. *Proc. Int. Conf. Pattern Recognition*, **4**, 64–67.

Ridler,T.W. and Calvard,S. (1978) Picture thresholding using an iterative selection method. *IEEE Trans. Syst. Man Cybernet.*, **SMC-8**, 630–632.

Sigal,A. *et al.* (2006) Dynamic proteomics in individual human cells uncovers widespread cell-cycle dependence of nuclear proteins. *Nat. Methods*, **3**, 525–531.

Souvenir,R. *et al.* (2008) Cell motin analysis without explicit tracking. *Computer Vision and Pattern Recognition*. Anchorage, AK, 1–7.

Szummer,M. and Picard,R.W. (1996) Temporal Texture Modeling. *IEEE Intl. Conf. On Image Processing*, **3**, 823–826.

Zhou,X. *et al.* (2009) A novel cell segmentation method and cell phase identification using Markov model. *IEEE Trans. Inf. Technol. Biomed.*, **13**, 152–157.