Cytometry
PART A
Journal of the International Society for Analytical Cytology

Automated Learning of Generative Models for Subcellular Location: Building Blocks for Systems Biology

Ting Zhao,[1,2] Robert F. Murphy[3,4,5]*

[1]Center for Bioimage Informatics, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213

[2]Department of Biomedical Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213

[3]Molecular Biosensor and Imaging Center, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213

[4]Department of Biological Sciences, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213

[5]Department of Machine Learning, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213

Received 11 September 2007; Accepted 8 October 2007

This work was presented at the XXIII Congress of the International Society for Analytical Cytology, Quebec City, Canada, 20—24 May 2006.

Grant sponsor: NSF; Grant number: EF-0331657; Grant sponsor: NIH; Grant numbers: U54 RR022241, U54 DA021519.

*Correspondence to: Robert F. Murphy; Carnegie Mellon University, 4400 Fifth Avenue, Pittsburgh, PA 15213, USA

Email: murphy@cmu.edu

Published online 30 October 2007 in Wiley InterScience (www.interscience.wiley.com)

DOI: 10.1002/cyto.a.20487

ISAC
International Society for Analytical Cytology

• Abstract

The goal of location proteomics is the systematic and comprehensive study of protein subcellular location. We have previously developed automated, quantitative methods to identify protein subcellular location families, but there have been no effective means of communicating their patterns to integrate them with other information for building cell models. We built generative models of subcellular location that are learned from a collection of images so that they not only represent the pattern, but also capture its variation from cell to cell. Our models contain three components: a nuclear model, a cell shape model and a protein-containing object model. We built models for six patterns that consist primarily of discrete structures. To validate the generated images, we showed that they are recognized with reasonable accuracy by a classifier trained on real images. We also showed that the model parameters themselves can be used as features to discriminate the classes. The models allow the synthesis of images with the expectation that they are drawn from the same underlying statistical distribution as the images used to train them. They can potentially be combined for many proteins to yield a high resolution location map in support of systems biology. © 2007 International Society for Analytical Cytology

• Key terms

location proteomics; generative models; pattern recognition; subcellular location; shape models; medial axis models; microscope image analysis; cell modeling; systems biology

A cell is a complex system with an enormous number of different types of molecules that form a large interacting network. The growing field of systems biology seeks to understand how living systems function by modeling such networks at various levels, including the interaction of molecules in cells (1–3). Building accurate models requires not only the chemical properties of the molecules involved, but also their spatial distributions. This is especially important for proteins because the subcellular location of a protein is so critical to its function that the same protein can have different functions at different locations (4). Thus, cell models will not yield accurate predictions unless proteins are modeled at their proper locations.

However, it is not easy to integrate subcellular location information into systems biology. In some cases, cell modeling can be done at a coarse level, such as by considering each major organelle as a single compartment (5). Given that cells go to great lengths to build complex subcellular structures, however, it is unlikely that coarse modeling will be sufficient for all purposes.

Thus, we need approaches that can provide information on subcellular location with as much resolution as possible. These can be divided into *predictive* methods and *determinative* methods. There has been extensive work on prediction of subcellular location from sequence (6–9). A range of methods have been described for learning to make predictions using the sequences of proteins whose location is known, including methods that use motifs, amino acid composition, homology, and combinations thereof. The major limitation of current systems is the resolution of the subcellular location assignments in the training data. These have been at the level of a handful of major organelles, and thus current systems are unable to predict the distribution of proteins within subcellular structures. In addition, while some systems can predict that proteins are located in more than one structure, they are unable to make quantitative predictions of the distribution of proteins between those structures. Lastly, current systems cannot predict dynamic behaviors such as cycling of proteins between compartments or changes in distribution resulting from stimuli. Thus, while the sequence-based machine learning methods that have been described are in theory well suited to the problem, their utility will only be fully realized when adequate high-resolution training data are available.

That is the domain of determinative methods. Currently, the best way to obtain high-resolution location information for many proteins is to acquire images by microscopy. Although visual examination is widely used to capture information from the resulting images, a more efficient way to extract and analyze the location information is using computer vision and machine learning methods (10,11). Previously we have designed Subcellular Location Features (SLF) to describe the patterns in microscope images (12,13) and the SLF allowed us to develop methods to determine locations automatically (12,14–17). These methods can be divided into two categories, classification and clustering, where the main difference is whether the set of location patterns is predetermined. In classification, any input image will be assigned to one of the classes (e.g., major organelles) that were used to train the classifier. In contrast, clustering does not assume that the categories are known but rather finds them by grouping similar patterns. With well-designed features, each category is expected to represent a single location pattern. Using a consensus clustering approach that is designed to yield reproducible clusters, we have built a subcellular location tree for 3D images of CD-tagged 3T3 cells (16) and shown that location patterns falling in the same human-labeled category can be separated into statistically distinguishable groups. These groups can be considered as subcellular location families, by analogy with families of proteins that share similar sequence (18) or structure (19).

However, learning what patterns are possible and which proteins display them is not sufficient for integrating location information into systems biology studies. Ultimately, location information must be incorporated into cell models to capture cell behaviors that depend on proper protein locations. To this end, images of location patterns can be used directly by some simulation programs after appropriate segmentation or reconstruction (20,21). However, this approach does not readily permit the effects of variation in pattern on the results of simulations to be considered in a systematic way. In other words, in the absence of a model for the variation in a pattern, the degree to which results of simulations depend on that variation can only be assessed by performing simulations for different input images. However, if a model is available, the choice of which patterns (and how many patterns) to use for simulations can be made in a principled manner taking into account the modes of variation in the pattern. Furthermore, the use of actual images as a base for simulations does not permit multiple proteins to be included in the same simulation unless they have been simultaneously imaged (e.g., using multiple fluorescent probes). As the number of proteins to be included grows, it is increasingly unlikely that they will have been imaged in the same cell unless specific experiments are done in contemplation of simulation. Even so, the number of proteins that can be simultaneously imaged in living cells is currently less than 10. An important alternative made possible by pattern models then is to combine models from separate images. In view of the above, we describe here methods for building probabilistic models that are learned from images of a location pattern and which we propose can be used to effectively include location information in cell simulations. Since it is straightforward to collect multichannel images in which distinguishable fluorescent probes are used to detect a specific protein in parallel with reference markers (such as DNA), we design our models to utilize markers for nuclear and cell shape.

The goal of the work described here is to build models of the distribution of a protein within a given cell type that are

- *automated* in the sense that they are learned directly from a set of microscope images,
- *generative* in the sense that they are able to synthesize new examples of the pattern observed in that set,
- *statistically accurate* in the sense that they reflect the variation in the pattern from cell to cell,
- *compact* in the sense that they can be communicated with significantly fewer bits than the training data.

With these definitions, we formalize the problem as

- *Given* a set of three-channel microscope images containing information in separate channels about the position of the cell boundary (plasma membrane), the distribution of nuclear DNA, and the distribution of a specific protein,
- *Build* an automated, generative, statistically accurate, and compact model of the distribution of that protein inside a cell described by its nuclear DNA distribution and plasma membrane location.

Our approach will be to construct a nested set of conditional models. We start by using a medial axis model to represent nuclear shape and a texture model to represent DNA distribution within that shape. Using the nucleus as a starting point, we generate a cell shape model. Finally, the nuclear shape and cell shape serve as a framework for locating specific proteins. The work described here uses 2-dimensional images and models to illustrate the principles and feasibility of the approach, and it is important to note that other choices are possible for each component of the model and that no claim of optimality is made. Work on building 3-dimensional models and using other model components is in progress. In this initial work, we also consider only interphase nuclei to simplify the task; future work will focus on a dynamic model incorporating variation across the cell cycle.

Materials and Methods

For development and testing of the algorithms described here, the images from the 3D HeLa dataset described previously (14) were used (available from http://murphylab.web.cmu.edu/data/3Dhela_images.html). The dataset contains three fluorescence channels for each field, reflecting the distributions of DNA, total protein, and one of nine specific proteins. Each field has been previously segmented into single cell regions using a seeded watershed approach (14). Since the models we describe here are two-dimensional, we extracted from each 3D stack the 2D slice that contained the largest total intensity in the DNA channel. Of the 454 images in the dataset, 7 did not have a complete cell boundary in this slice. We therefore ignored these images leaving 447 images for the studies described here. Protein location models were created for six of the proteins in the dataset, including giantin, gpp130 (both Golgi proteins), LAMP2 (a lysosomal protein), a mitochondrial protein, nucleolin (a nucleolar protein) and transferrin receptor (an endosomal protein).

The algorithms used in this work were implemented using Matlab (7.0 R14, The MathWorks) and all code is available from http://murphylab.web.cmu.edu/software. While the specific algorithms for creating each component of the model are described in the Results, additional implementation details are presented here. For modeling nuclear shape, a principal axis alignment was done for each nuclear (DNA) image as described previously (22). Briefly, after thresholding using the Ridler-Calvard method, each nucleus was rotated to align its major axis and rotated an additional 180° if necessary to match the sign of the skewness along the minor axis. For modeling nuclear texture, texture synthesis toolboxes were downloaded from http://www.nealen.net/projects/texsynth/hts_code.zip and http://www.cns.nyu.edu/∼lcv/texture. The hybrid texture synthesis code was modified slightly to avoid searching patches of background pixels so that background would not be counted as a part of the texture. For modeling cell shape, the total protein image was thresholded just above the most common pixel intensity and the largest resulting object (the cell) was then filled and outlined. For estimating Gaussian object mixtures, EM code originally from the NETLAB library (http://www.ncrg.aston.ac.uk/netlab) for Matlab was used. Some modifications were made to support estimating the weighted Gaussian mixture. For classifying patterns, the code to calculate feature set SLF7DNA from the SLIC library (http://murphylab.web.cmu.edu/software) was used. The SDA implementation in SLIC was also used. Code for training and using support vector machines was obtained from the LIBSVM library (http://www.csie.ntu.edu.tw/∼cjlin/libsvm). A Gaussian kernel was used and the parameters were searched automatically for best performance on the training set in each trial.

Results

Medial Axis Model of Nuclear shape

Building a model for nuclear shape is the first step in our modeling procedure. This is important in its own right given the critical role of the nucleus in duplicating and expressing genetic information. In addition to indicating the state of the cell cycle, the size and shape of the nucleus can affect gene expression and protein synthesis (23).

Interphase nuclei typically have a shape similar to an ovoid or ellipse (Fig. 1), which is a uniaxial object. We have therefore used a method adapted from medial axis transformation (24–26) to fit the shape. For a 2D shape, the traditional definition of a medial axis is a set of the centers of the circles that support the shape. This is the same as building a Voronoi graph for all the points on the shape boundary in the 2D space. In our version of medial axis representation, we restrict the distance calculation of the Voronoi graph to one dimension, the x-axis. This can avoid generating branches, which will make a shape much more complicated to model. It also reduces the complexity of computation, as is described below.

We consider the shape of a nucleus to be described by a parametric curve $[x(t), y(t)]$. We can define another curve $h(u) = [y(t_1) + y(t_2)]/2$, where t_1 and t_2 satisfy $x(t_1) = x(t_2) = u$, $y(t_1) = \max\{y(t)|x(t) = u\}$ and $y(t_2) = \min\{y(t)|x(t) = u\}$. These definitions find the highest and lowest points along a series of lines perpendicular to the major axis of the nucleus and then averages them to find $h(u)$, which is the medial axis. The width along the medial axis is $w(u) = y(t_1) - y(t_2)$. For convenience, we normalize distances so that u is in the interval [0,1]. Given the medial axis and the width, we can easily reconstruct the shape. This definition also provides an easy way to find the medial axis in an image. Given a DNA image for a single cell region, we threshold it to yield an $M \times N$ binary digital image $f(i,j)$, where $i = 1,2,\ldots,M$, $j = 1,2,..,N$. $f(i,j) = 1$ only when the pixel (i,j) belongs to the nuclear object (i.e., is above-threshold); otherwise $f(i,j) = 0$. Then the medial axis of the nucleus in this image is defined on $\{i|\exists j, f(i,j) = 1\}$ and a point of the medial axis at position i is $(\min\{j|f(i,j) = 1\} + \max\{j|(f(i,j) = 1\})/2$. Thus the width is $w(i) = \max\{j|(f(i,j) = 1\} - \min\{j|f(i,j) = 1\}$. Figures 2a–2d illustrate the steps we have used to convert a nuclear image into a medial axis representation.

Each of the two parts of the medial axis representation, the medial axis and width, can be represented by a curve. To parameterize such a curve, we fit a fourth order B-spline with one internal point; we chose this order because it gives very good fits to the nuclear width (Fig. 2f) and reasonably accurate fits to the medial axis itself (Fig. 2e). (Some of the high frequency variation in the medial axis may be due to digitization artifacts from rotation; the spline fit results in some smoothing of this variation but may also smooth nuclear blebs.) Combining the six parameters for each of the two spline fits with a parameter for the range of the medial axis along the x-axis gives a total of 13 parameters for nuclear shape. We have observed the fitted values of the internal point to be around 0.5 for both the medial axis and width curves and to have little contribution to the variation of the shape (data not shown). Since the curves are defined over the internal [0,1], this suggests that nuclei are roughly symmetric about their center. We therefore chose to take the internal point as a constant, leaving 11 remaining free parameters.

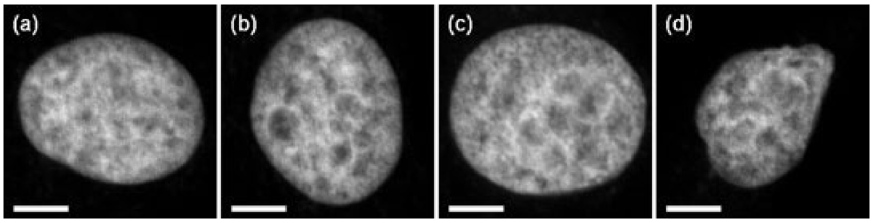

Figure 1. Examples of nuclear images. The 447 nuclei in the 3D HeLa dataset were ranked by their Mahalanobis distances to the mean value of the 11 parameters describing the shape based on medial axis. The nuclei shown are (**a**) the most typical nucleus, (**b**) the 100th most typical nucleus, (**c**) the 200th most typical nucleus and (**d**) the 400th most typical nucleus. Scale bar, 5 μm.

A statistical model is required to describe the variation of parameters from nucleus to nucleus. As an initial approach, we have chosen to use multivariate normal distributions to represent this variation. Figure 3 shows quantile–quantile plots of each parameter (a straight line on these plots indicates close agreement to a normal distribution). The range of the medial axis (Fig. 3a) and the parameters of the width curve (Figs. 3g–3k) are all fit well by normal distributions, while the parameters of the medial axis curve are not as well fit (we intend to consider other distributions in future work). For distribution estimation, we assume that the parameters of the medial axis and width are independent. This increases the flexibility of shape distribution and reduces the number of parameters. Subject to the assumption of normality for the parameters, we can capture the entire nuclear shape model using 47 values: the 11 means for each parameter and the 36 entries in the covariance matrices of the medial axis and width parameters (the medial axis has six parameters giving 21 unique entries in its covariance matrix, and the width distribution has five parameters giving 15 entries in its covariance matrix). To generate a nuclear shape, we can draw parameters from the normal distribution (Fig. 3) and then construct a shape from the drawn parameters.

Texture of Nucleus

Given an image of the DNA distribution of a cell, the nuclear texture reflects the condensation of chromosomes within the nucleus and the variation in DNA content along each chromosome. Nuclear texture analysis has shown to be significant for biomedical studies such as cell cycle examination (27) or disease diagnosis (28,29). The next task we consider is therefore building a model for chromatin texture that can be used to synthesize realistic nuclear DNA images.

We chose to apply texture modeling and synthesis techniques frequently used for natural images (30). However, these techniques work best for textures that are homogeneous within a shape, which is usually not the case in a nucleus. Among many reasons, this is because the average intensity decreases from the center of a nucleus to its boundary (Fig. 4). This is because of decreasing thickness of the three-dimensional ellipsoid nucleus near its edge and also because regions in the middle of the nucleus may contain more out-of-focus light from above and below the image plane. So before estimating a nuclear texture model, the pixel intensities must be adjusted for variation in average intensity across the nucleus. This variation can be fit well by a function derived from ellipse projection (Fig. 4): $f(d) = a + b(1 - d^2)^{1/2} + c(1 - d^2)^{1/4}$, where d is the normalized distance to the center of the nucleus. After finding the function parameters, the intensity $I(x, y)$ at the position (x, y) is scaled as $I(x, y)/f(d(x, y))$.

After intensity normalization, we used a neighbor-based method to extend the texture to remove all background (31). The texture image is then modeled by a parametric model using wavelets (32). Combining the medial axis shape model with the texture model allows us to synthesize nuclear images with proper shape and approximate chromatin texture (Fig. 5).

Shape Model of Cells

By definition, subcellular location of a protein takes place within the bounds of the plasma membrane. We therefore

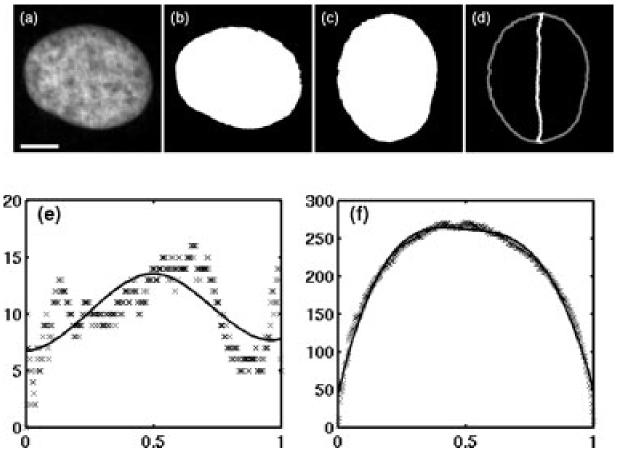

Figure 2. Example of fitting the medial axis description of a nuclear shape by B-splines. The original image (**a**) containing a nucleus was processed into a binarized image (**b**), in which the nuclear object consists of the white pixels. The nuclear object was rotated so that its major axis is vertical (**c**). It is then converted into the medial axis representation (**d**). The horizontal positions of the medial axis as a function of the fractional distance along it are shown by the symbols in (**e**), along with a B-spline fit (solid curve). The width as a function of fractional distance is shown by the symbols in (**f**), along with the corresponding fit (solid curve). Scale bar, 5 μm.

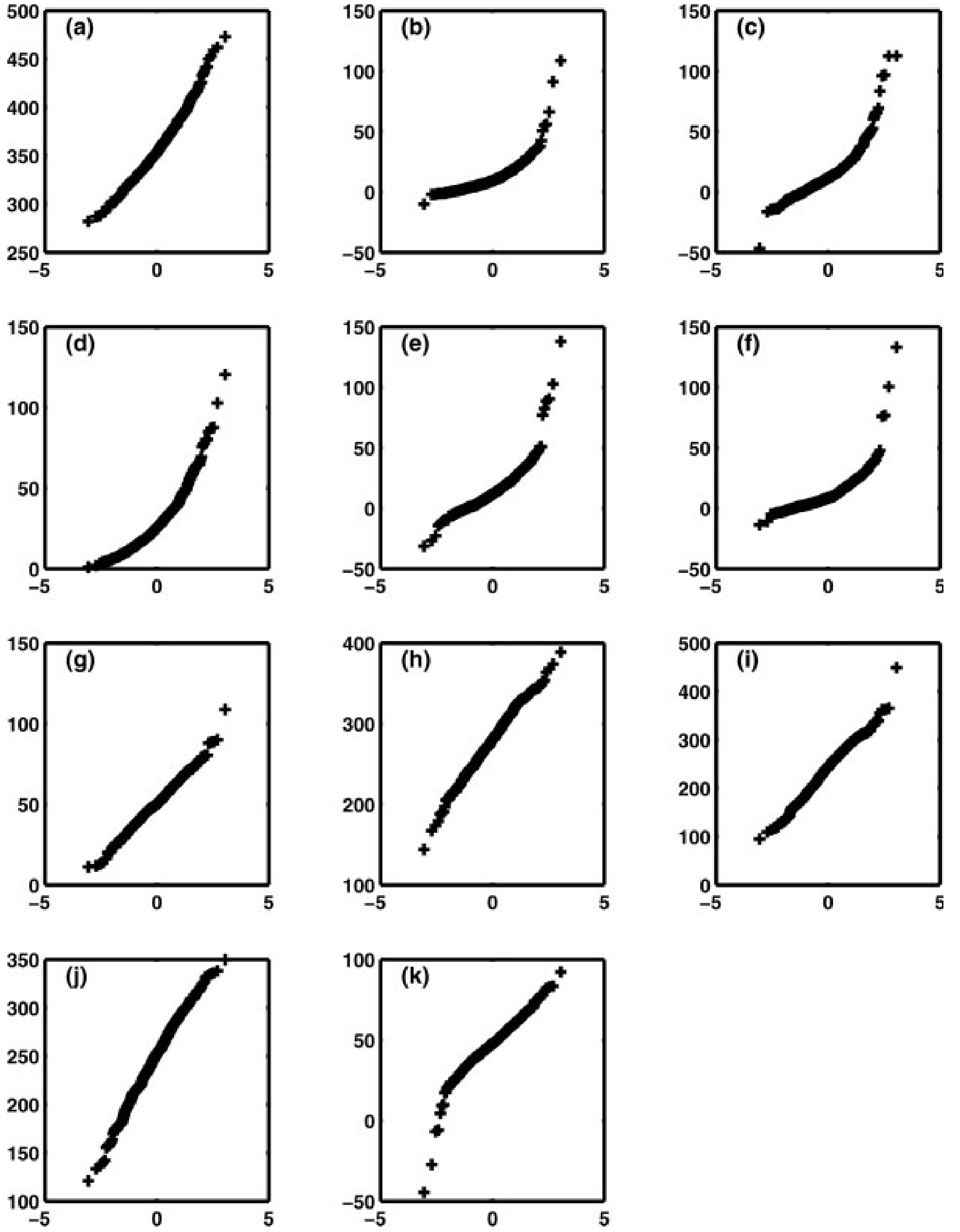

Figure 3. Estimating normality of the distributions of the parameters of the medial axis representations for all nuclei. Quantile—quantile plots comparing the distributions of each parameter across all 447 nuclei (on the vertical axis) to a Gaussian distribution (on the horizontal axis) are shown. (**a**) The length of the medial axis. (**b—f**) The five parameters of the medial axis curve. (**g—k**) The five parameters of the width curve.

next incorporate a model for cell shape. Different types of cells can have very different shapes, which are often related to their functions. For example, a neuron has a tree-like structure for signal conduction, while columnar epithelial cells are roughly rectangular so that they can be tightly connected to separate different environments. What we deal with here are shapes of cultured HeLa cells, which adhere to glass surfaces and spread their cell bodies out to take on a "fried egg" shape (Fig. 6).

Although some general shape models such as polygons (33) or active shape models (34) have been used to model cell shapes, for our purposes we wished to make the cell shape model conditional on the nuclear model described above (that is, we wished to consider the correlation between the shape of a cell and its nucleus). Figure 7 shows the histogram of the differences between nuclear major axis angle and cell major axis angle and the histogram of the distances between the nuclear

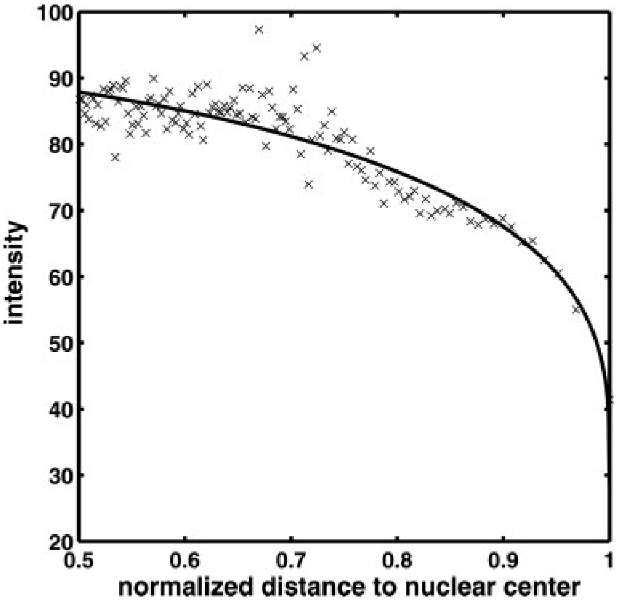

Figure 4. Capturing nuclear intensity variation. The intensity in a nucleus decreases from the center to the boundary (x) and it can be fit by a simple function as described in the text (solid line).

center and cell center. From Figure 7a we concluded that the nucleus and cell are aligned at similar orientations. Similarly, Figure 7b shows that the center of the nucleus and the center of the cell are typically close to each other, with an average distance of 2.2 μm. If we model the cell shape independently, we then need to model the correlations between nuclear and cell alignments, including positions and orientations. But if we build a cell shape model that is conditional on the nuclear shape model, this will not be necessary.

The conditional shape model we build can be illustrated in a polar coordinate system, of which the origin is at the center of the nucleus. The boundary of the nucleus and the boundary of the cell are denoted as $d_n(\theta)$ and $d_c(\theta)$ respectively, where θ is the angular coordinate and belongs to $[0,2\pi]$. Because we know $d_n(\theta)$, it would be sufficient to describe the cell shape using the radial coordinate ratio between the two shapes. We call this the shape ratio of a cell, which is also a function of angles and denoted as $r(\theta)=d_c(\theta)/d_n(\theta)$.

If we sample θ over 360° in 1 degree increments, a shape will be represented by a vector of length 360. Estimating the statistical distribution of the vectors will require much more data than we have to guarantee accuracy. To solve the problem, we used principal component analysis (PCA) to reduce the dimensions, as was done in active shape models (34). First, the average shape ratio was calculated by taking the mean of the shape ratios of all the cells. The residuals (the differences between a cell's shape ratio vector and the average shape ratio vector) were calculated for each cell. So the average ratio $\bar{r}(\theta)$ and the residual $\delta_i(\theta)$ of the ith cell are calculated as $\bar{r}(\theta) = \frac{1}{N}\sum_{i=1}^{N} r_i(\theta)$ and $\delta_i(\theta) = r_i(\theta) - \bar{r}(\theta)$, where N is the number of cells. The principal components representation of $\delta_i(\theta)$ is $\sum_{j=1}^{N}\lambda_{ij}e_j(\theta)$, where e_j is the jth principal component and λ_{ij} is its coefficient from the ith cell. The indices are arranged in decreasing order of their contribution to the overall variation. We can discard some components without losing the essential properties of the shape.

The λ_{ij} matrix can be modeled by a multivariate normal distribution, from which we can draw samples to synthesize a cell shape. In our implementation we used 10 components, which contain about 90% of the variation (data not shown). Figure 8 shows the average shape and shapes illustrating the four highest modes of shape variation. An example shape synthesized from the model with 10 components is also shown. The cell shapes represent good approximations to real cell morphologies, although fine structure in the cell boundaries is not captured well.

Protein Object Modeling

Gaussian objects. Previously we have shown that subcellular location images can be well-modeled by combinations of individual objects, which are defined as contiguous regions of non-zero pixels in a segmented image (35). We therefore focus in this paper on patterns that are comprised mainly of small, roughly ellipsoidal objects (such as lysosomes and endosomes). To model these objects as seen in 2D images, we can use 2D Gaussian distributions, $N(\mu,\Sigma)$, where $\mu = \begin{pmatrix}\mu_x \\ \mu_y\end{pmatrix}$ and $\Sigma = \begin{pmatrix}\Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy}\end{pmatrix}$. These parameters can be directly calculated from the image of a single object. However, organelles such as vesicles can aggregate or overlap to form a larger object in an image that is non-Gaussian in shape. We therefore used Gaussian mixture distributions to describe the large objects as combinations of smaller objects.

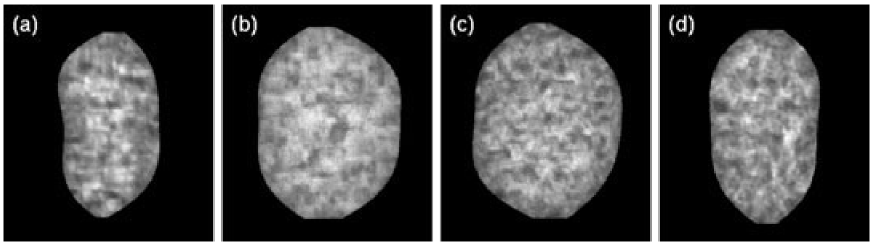

Figure 5. Examples of synthesized nuclei. Each nucleus is synthesized with two parts, shape and texture.

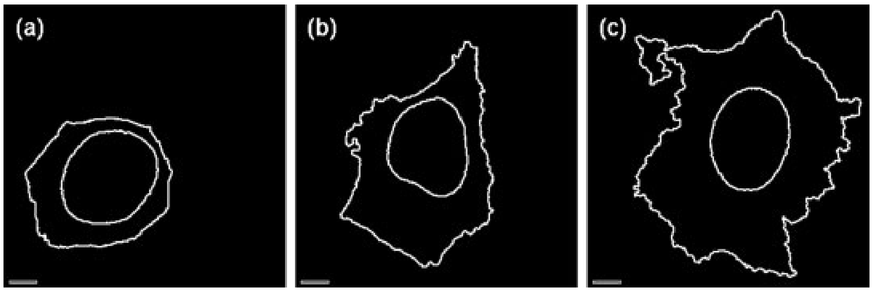

Figure 6. Examples of cells with different shapes, including a cell with the smallest cytoplasm-nucleus area ratio (**a**), a cell with the ratio closest to the average ratio (**b**), and a cell with the largest ratio (**c**). Scale bar, 5 μm.

The probability density function (PDF) of a Gaussian mixture distribution can be denoted as $f(x\,|\,\vartheta) = \sum_{k=1}^{m} p_k g(x|\mu_k, \Sigma_k)$, where $\vartheta = \{p_k, \mu_k,\ \Sigma_k\ |\ 0 \le p_k \le 1,\ k = 1, \ldots, m\}$, $\sum_{k=1}^{m} p_k = 1$ and $g(x\,|\,\mu_k, \Sigma_k)$ is the PDF of the Gaussian distribution with mean μ_k and covariance matrix $\sum_k$. In fact, the Gaussian mixture distribution can describe small objects as well because a Gaussian distribution is a Gaussian mixture distribution with one component.

The expectation-maximization (EM) algorithm can be used to estimate a Gaussian mixture distribution (36). However, the EM algorithm requires the number of components as an input. To estimate this number for an object, we used a low-band filter to smooth the object and the take the number of local maxima of the object intensities as the number of components. Since each data point has a weight, which is the intensity of the pixel ($w_i = I(x_i, y_i)$), we used the weighted EM algorithm as follows,

E step:

$$\alpha_{ik} = \frac{P_k^{(t)} g(x_i|\mu_k^{(t)}, \sum_k^{(t)})}{\sum_{k=1}^{m} P_k^{(t)} g(x_i|\mu_k^{(t)}, \sum_k^{(t)})}$$

M step:

For k from 1 to m

$$P_k^{(t+1)} = \frac{1}{\sum_{i=1}^{N} w_i} \sum_{i=1}^{N} w_i \alpha_{ik},$$

$$\mu_k^{(t+1)} = \frac{1}{\sum_{i=1}^{N} w_i \alpha_{ik}} \sum_{i=1}^{N} w_i \alpha_{ik} x_i,$$

$$\sum\nolimits_k^{(t+1)} = \frac{1}{\sum_{i=1}^{N} w_i \alpha_{ik}} \sum_{i=1}^{N} w_i \alpha_{ik} (x_i - \mu_k^{(t+1)})(x_i - \mu_k^{(t+1)})^T.$$

The solutions were obtained when the iteration of the two steps converged.

The representation of Gaussian mixture leads to a new definition of objects, which are called Gaussian objects because each object is defined as a density function of a 2D Gaussian distribution multiplied by a total intensity (this is $p_k \sum_{i=1}^{N} w_i$ for the kth component). In the general Gaussian object, all of the elements of the covariance matrices are free parameters. However, this can lead to fitted objects with very large or highly elongated shapes not typical of subcellular organelles like lysosomes and endosomes (Fig. 9). To minimize this effect, we can require each Gaussian object to be circularly

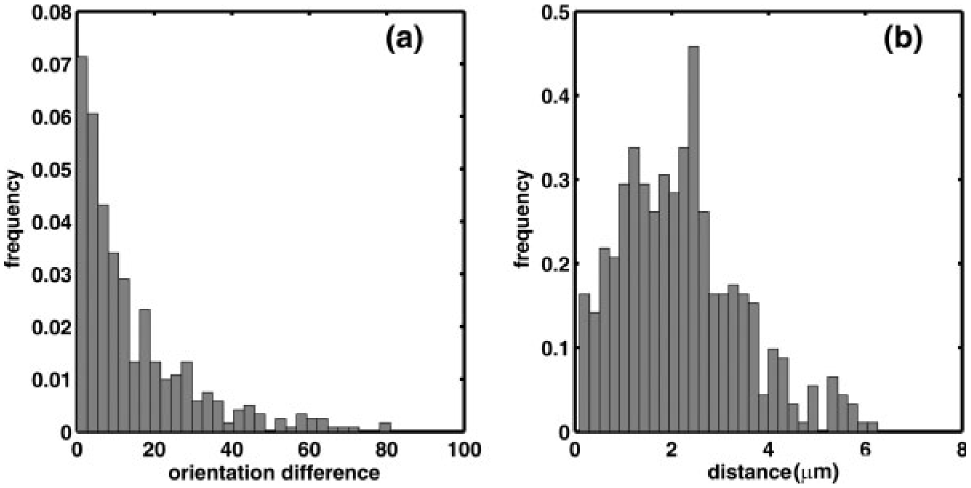

Figure 7. The correlation between the cell morphology and nuclear morphology is shown by (**a**) the histogram of the differences between nuclear angle and cell angle and (**b**) the histogram of the distances between nuclear center and cell center.

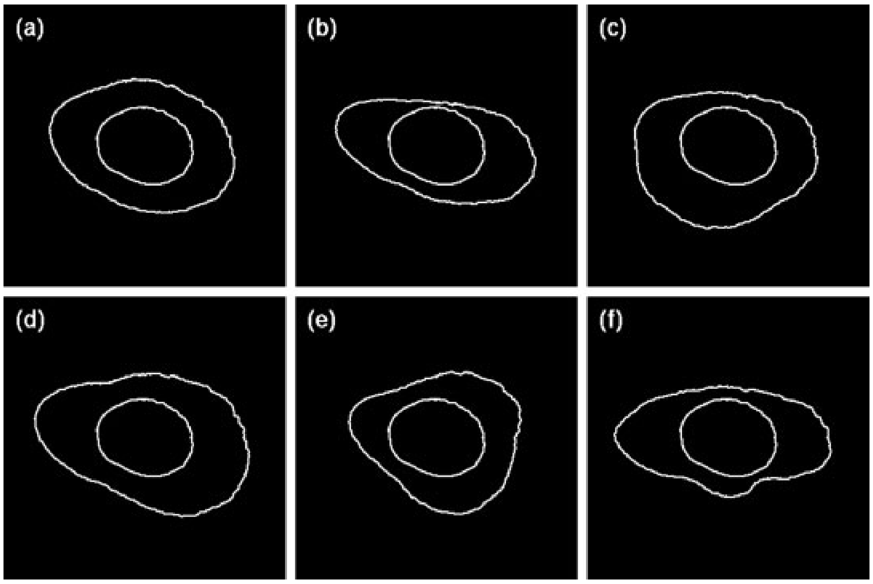

Figure 8. Illustration of the conditional cell shape model. In the cell shape model described in this paper, the shape of a given cell is described by the ratio of the cell size to the nuclear size at each of 360 angles. Variation in cell shape is captured in two parts, the average shape ratio and the variation of the differences between specific cell shape ratios and the average shape ratio. The six figures shown here are (**a**) the cell morphology corresponding to the average shape ratio, (**b—e**) the cell shapes after adding each of the first four principal components respectively, and (**f**) a cell shape synthesized with shape ratios drawn at random from the distributions. For figures (b—e), the *i*th principal component was added with the coefficient 1.5 σ_i, where σ_i is the standard deviation of the coefficient of the *i*th component (the square root of the eigenvalue of that principal component).

symmetric (by constraining the covariance matrix to have equal values along the diagonal and zero values for the off-diagonal elements). This gave better results (as judged by comparison with real images) than the full covariance matrix (data not shown).

To describe the statistics of the Gaussian objects, we found that the standard deviation of the objects (which controls their size) can be fit by an exponential distribution (Fig. 10a). In addition, the relative intensity of objects, which we define as the square root of the ratio between the intensity and variance, can also be fit by a Gaussian distribution (Fig. 10b). The distribution of the number of Gaussian objects in each cell can be fit by a Gamma distribution (Fig. 10c). To determine how many Gaussian objects exist in a cell, we draw a number from the Gamma distribution and round it to the nearest integer.

Object position model. In addition to the number and sizes of objects in a cell, the positions of these objects are important for synthesizing a realistic pattern. Proteins can be readily divided into cytoplasmic, nuclear, or membrane bound. Therefore, we modeled the positions of protein-containing objects using two parameters describing their relationship to the nuclear and plasma membranes. The first, *r*, is defined as the ratio of the distance of a given object to the nuclear membrane to the sum of that distance and the distance to the cell membrane. The second, *a*, is defined as the angle between a line from the center of the nucleus to an object's position and the major axis of the nucleus. The distance of an object to the cell boundary or nuclear boundary can be found by building distance maps. For example, to calculate the distance of an object to a nucleus, we first obtain a binary image that only contains the edge of the nucleus. Then we build a distance map in which

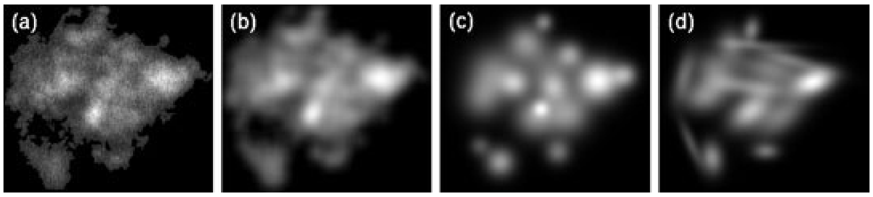

Figure 9. Example of fitting objects by 2D Gaussian mixture. (**a**) An image containing the original object is (**b**) smoothed by a Gaussian lowpass filter. Then the number of Gaussian objects is decided by the number of local maxima in the smoothed images. We can use either (**c**) spherical or (**d**) full covariance matrices while fitting the objects by the EM algorithm.

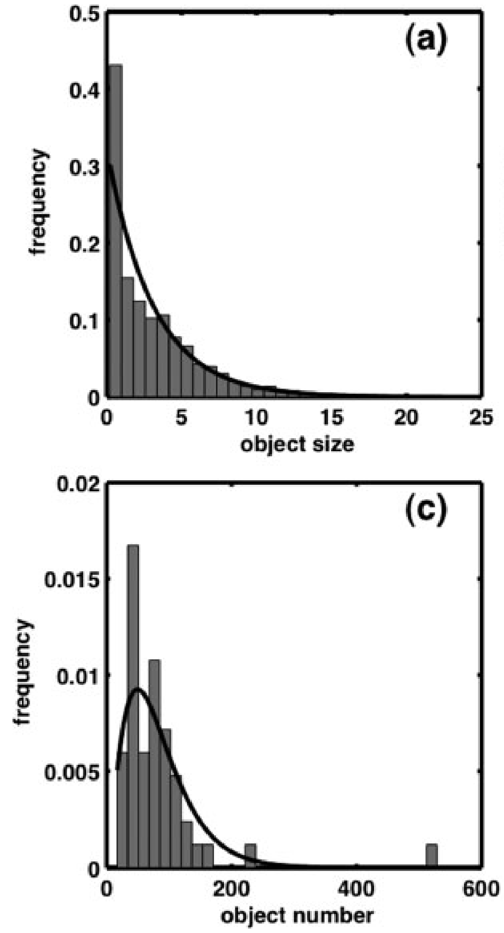

Figure 10. Statistics of Gaussian objects for lysosomal protein images. (**a**) The distribution of object sizes across all images and the corresponding fitted Gaussian distribution. (**b**) The distribution of relative intensity per object across all images and the corresponding fitted Gaussian distribution. (**c**) The distribution of number of objects per cell and the corresponding fitted Gamma distribution.

the intensity of the pixel is the smallest distance of that pixel to the nuclear edge. The distances to the cell membrane were obtained in the same way. This permits the distribution of object positions for a given cell to be converted into a distribution of *r,a* values. We model this distribution or potential (the probability that a given position is the center of an object) as

$$P(r, a) = \frac{e^{\beta_0+\beta_1 r+\beta_2 r^2+\beta_3 \sin a+\beta_4 \cos a}}{1 + e^{\beta_0+\beta_1 r+\beta_2 r^2+\beta_3 \sin a+\beta_4 \cos a}}$$

and determine the values of the parameters by logistic regression. Here the angle is transformed into the linear combination of sin and cos functions because its value is periodic, i.e. a and $a + 2\pi$ are the same angle. The parameters in this model can easily be interpreted. For example, β_1 shows whether the protein is more likely overlapping (when β_1 is positive) or not overlapping (when β_1 is negative) the nucleus. β_3 and β_4 determine whether the protein is more likely distributed along the major axis (when $|\beta_3|$ is small and $|\beta_4|$ is large) or minor axis (when $|\beta_3|$ is large and $|\beta_4|$ is small) of the nucleus.

Figure 11 shows examples of potentials learned from images of the lysosomal protein LAMP2. To use the potentials to predict object positions, we normalized them so that their sum is 1. This makes them the probabilities of an object being found at each location in the two-dimensional image grid. To obtain object locations for a synthetic image, we randomly choose the number of objects for that image from the Gamma distribution discussed above and then randomly draw that many object positions from the multinomial distribution specified by the position probabilities.

Given this model for protein object positions, we can finally synthesize location images containing all three channels; examples synthesized from the trained models for six proteins are shown in Figure 12. Many additional example images are available at http://murphylab.web.cmu.edu/data.

Evaluation of Synthesized Images

Having described how the generative model is created and how it can be used to synthesize images, the natural question is: How good are the synthesized images? We expect that the synthesized images from good generative models should be similar to real images. A simple way to verify this is to visually determine the degree of difference between the real

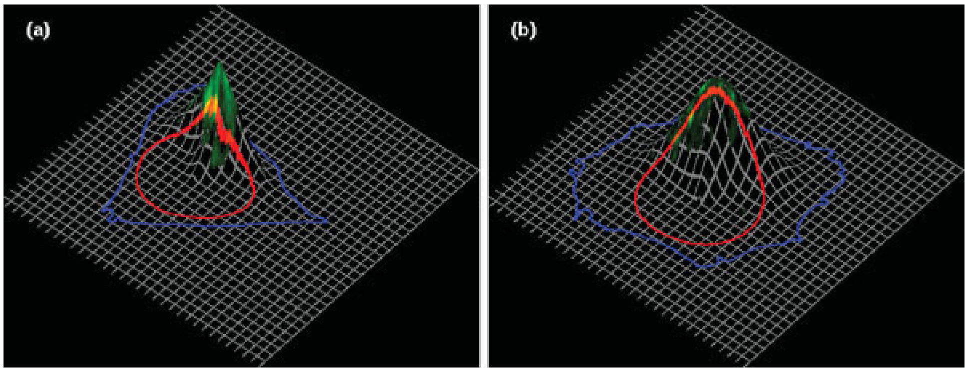

Figure 11. Illustrations of object position potentials. The estimated potentials of object positions for lysosomal proteins are shown for two cells as a surface in 3D space. The higher the pixel, the more likely it is for an object to be located at that position. Cell (blue) and nuclear (red) outlines are shown and the protein is shown in green.

and synthesized images. However, this is not a suitable approach for our generated images, which are visually distinguishable from real images because a number of sources of noise have been removed to estimate an ideal distribution. It is also difficult to make quantitative estimates of similarity using visual examination. We therefore chose to compare the images using the Subcellular Location Features, which have been shown to represent location patterns very well (13). This can be done by training a classifier on SLF of real images and then applying it to the SLF of synthesized images to see how well the synthesized images can be classified. Here we used the SLF7DNA feature set minus feature SLF7.79, the fraction of cellular fluorescence not included in objects (since the synthesized images contained no fluorescence that was not in objects). This left 89 features, including 13 texture features after downsampling to a pixel size of 1.15 μm and 32 gray levels, 49 Zernike moment features, 5 object skeleton features, 8 morphological features, 6 DNA features, 5 edge features, and 3 convex hull features (12,13). We used stepwise discriminant analysis (SDA) (37) to select the most informative features for both the real and synthesized images. SDA returned 40 features ranked in order of their ability to distinguish the classes. Support vector machines were trained on the real images using increasing numbers of these features. These were applied to the synthesized images to test how well they can be recognized. We considered the DNA pattern as a class, represented by the synthesized nuclear images.

Figure 13 shows the average accuracies of classifying real and synthesized images using various numbers of features. The average classification accuracies were calculated after merging the two Golgi proteins, giantin and gpp130, into a single class since their patterns are so similar. Variation in accuracy occurs as additional features are added, due to at least two sources. The first (small) source of variation is the sampling variation that happens between averages of random cross-validation trials. The second, larger source of variation occurs only in the case when the population of training images is not expected to be identical to that of the testing images (e.g., for the case of training using real images and testing using synthetic images). In this case, the addition of a feature that distinguishes among the classes of the real images better than among the synthetic images can lead the classifier to put weight on that feature at the expense of the previous features, and lead to a decline in performance. The decline can poten-

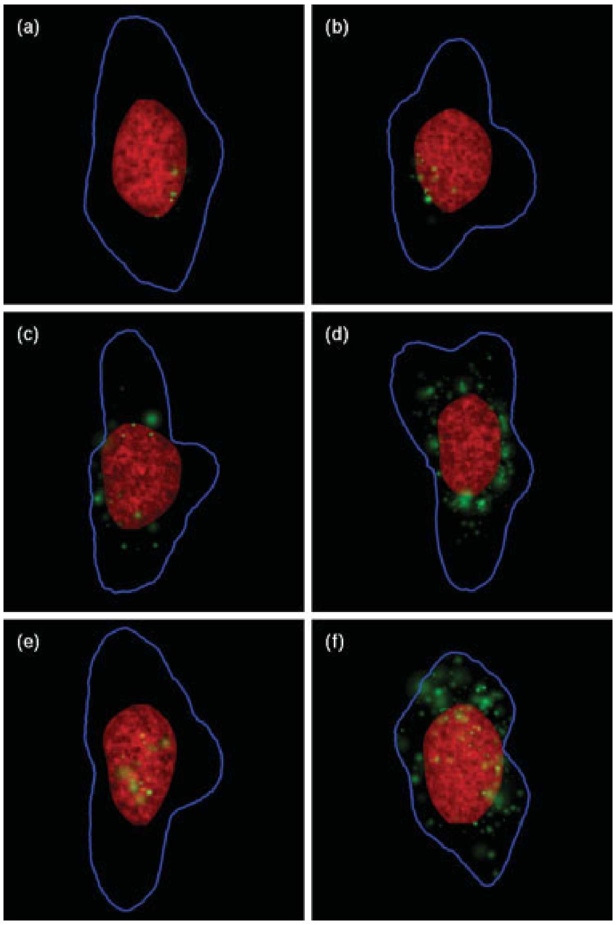

Figure 12. Synthesized images for different protein patterns. Red: nucleus; Blue: cell membrane; Green: protein. The proteins are: (**a**) giantin, (**b**) gpp130, (**c**) LAMP2 (lysosomal), (**d**) a mitochondrial protein, (**e**) nucleolin, and (**f**) transferrin receptor (endosomal).

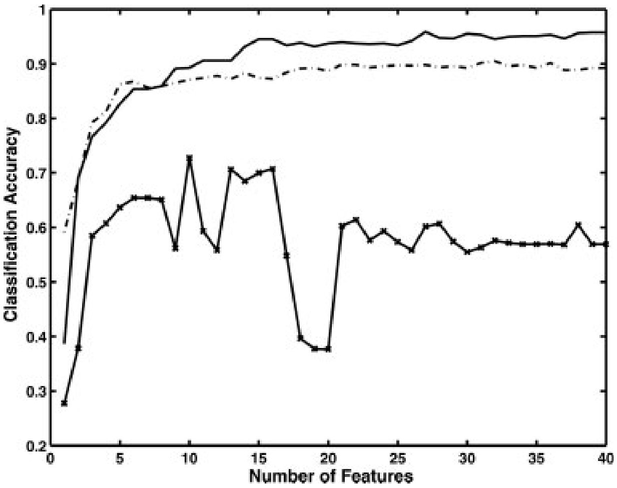

Figure 13. Evaluation of synthesized images. Classification accuracies are shown as a function of the number of features used under three conditions: classifying synthesized images by a classifier trained on real images (x), training and testing on real images using cross validation (−) and training and testing synthesized images by cross validation (−).

tially be partially reversed if a new feature is added that distinguishes among the classes well for both the real and synthetic images.

Among the feature sets giving classification accuracies on real images that were higher than 90%, a set of 16 features resulted in the best classification accuracy for the synthetic images, 71%. The confusion matrix for this case is shown in Table 1. The accuracy of classification of real images of nine classes is 95%, which means that these features contain almost all necessary information to distinguish the major patterns. The average accuracy of classification using only synthesized images is 87%. Thus the images generated by each of the models are clearly different from each other even if they are not always correctly recognized by a classifier trained on real images.

We further tested how well the model parameters could be used to discriminate real images. According to the models, each protein image has nine features, one parameter for the number of Gaussian objects, one parameter of object size distribution, two parameters of object intensity distribution and five parameters of object position model. Using just these nine features we obtained a classification accuracy for real images of 88% (Table 2), which means that the models captured most essential information to distinguish the six patterns. This is an important result in that the generative model parameters may be considered to be a more "natural" representation of the image patterns than previously described features.

Discussion

This paper presents a framework for building generative models of location patterns. The ability to represent and generate subcellular distributions for all proteins will be important for systems biology. An important aspect of our framework is that the parameters of the models are all learned from real data, enabling them to be applied to large scale projects that are analyzing thousands of proteins (38). A critical advantage of generative models over simple collections of images for the purpose of representing subcellular patterns is that correlations between components of the model (such as possible correlations between nuclear orientation and cell shape) that might be difficult to perceive from visual inspection can be identified and captured.

Beyond simply describing a system for building such models, however, we have also described an approach for the evaluation of the images generated by these models using classifiers trained on real images. This is a critical advance, since there are many possible approaches to model building that could be considered. The results in Table 1 show that most synthesized images were correctly classified and also indicate which patterns need further model improvement in future work. The only pattern classified with low accuracy is the mitochondrial pattern, for which most images were classified as the endosome pattern.

We note also that the use of generated images in simulation studies in the future will provide an additional (and potentially better) way to evaluate them: how they affect the agreement between simulation results and experimental

Table 1. Classification of synthesized images[a]

	OUTPUT OF CLASSIFIER									
TRUE CLASSIFICATION	DNA	ER	ACTIN	GIA	GPP	LYSO.	MIT.	NUC	ENDO.	TUB.
DNA	**100**	0	0	0	0	0	0	0	0	0
Gia	0	0	0	**31**	**54**	13	0	1	1	0
Gpp	0	0	0	**24**	**62**	11	0	2	1	0
Lyso.	0	0	0	7	4	**50**	7	0	32	0
Mit.	0	0	0	0	0	2	**18**	0	80	0
Nuc.	1	0	0	4	15	0	0	**80**	0	0
Endo.	0	2	0	0	0	1	2	0	**91**	4

[a] A classifier was trained using 16 features of real images. One hundred images were generated for each pattern shown in the row headings. The values shown are the percentage of synthesized images for each row that were classified as one of the 10 patterns shown in the column headings. Boldface numbers indicate the percentage of correctly classified images.

Table 2. Classification of real images based on the model parameters[a]

	OUTPUT OF CLASSIFIER					
TRUE CLASSIFICATION	GIA	GPP	LYSO	MIT.	NUC.	ENDO.
Gia	**86**	8	6	0	0	0
Gpp	6	**80**	10	2	2	0
Lyso.	2	6	**84**	4	0	4
Mit.	0	0	4	**88**	0	8
Nuc.	0	0	0	0	**100**	0
Endo.	0	0	2	10	0	**88**

[a] Generative model parameters were estimated for individual images and used to train and test classifiers using 10-fold cross-validation. The average accuracy was 88%. Boldface numbers indicate the percentage of correctly classified images.

results. Such studies will also potentially indicate directions to improve the models.

We note that lysosomes are observed to overlap the nucleus in both real images and synthesized images. This appears to be an artifact of imaging (presence in the same optical section of lysosomes above or below a section of nucleus) that is carried over into the models (since lysosomes cannot normally enter the nucleus). True three-dimensional models are required to solve this problem, and we are currently pursuing this direction. However, the 2D location models we have described are likely to be useful for those cases where a model is confined to 2D (e.g., for computational efficiency).

While our current models may be useful immediately, there are two important additional characteristics needed to build accurate cell simulations. The first is to build models that specify the location of multiple proteins (and eventually all proteins) in the same cell. Only a small number of proteins can currently be visualized in live cells using fluorescence microscopy. An important alternative is to use fixed cells and obtain correlated distributions by repeated cycles of staining, imaging and photobleaching (39). While this is thought to be able to image up to 100 proteins in the same sample, it is unlikely that it can be extended to simultaneously measure tens of thousands of proteins in the same cell (and of course it cannot be applied to living cells). Thus, methods for combining information from different cells are necessary, and generative models can play this role. Proteins can first be grouped into high-resolution subcellular location families and then a generative model can be built for each family. These can be combined to synthesize cell models showing tens of thousands of proteins under the assumption that all proteins in a family show highly-correlated distributions.

The second necessary characteristic of future models is the ability to represent changes in protein distribution over time, on time scales from below a second to greater than a year. The ability to directly acquire information on the dy-

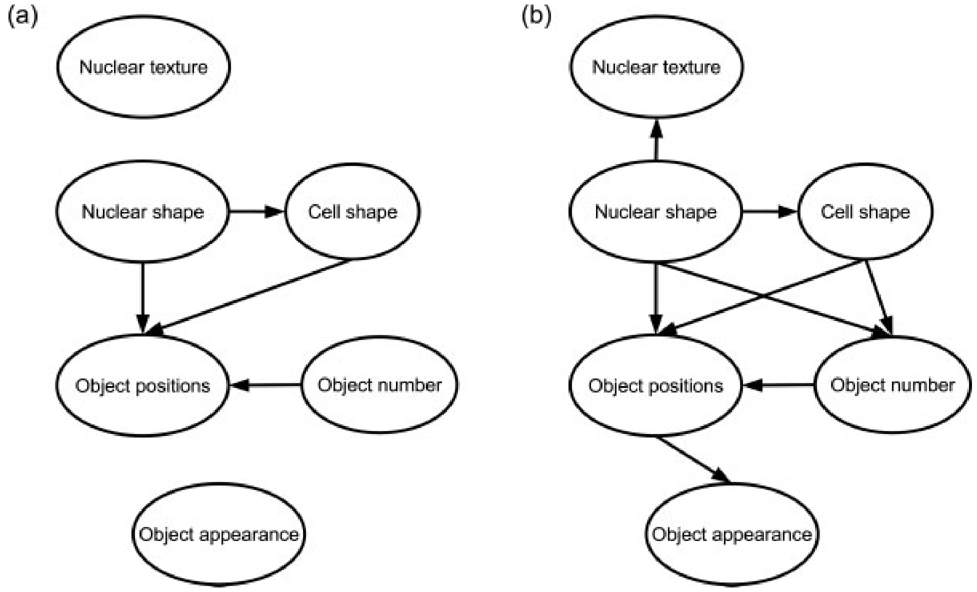

Figure 14. Description of the models as Bayesian networks. The network representing the models built in this paper is illustrated by figure (**a**). More accurate but complicated models can be obtained by adding edges to the network (**b**).

namics of protein distribution is a critical advantage of fluorescence microscopy.

In this vein, we can consider ways of representing generative models and choosing their characteristics. The models proposed in this paper can be put into a directed probabilistic graphical model framework, which is also known as Bayesian network (Fig. 14). The advantage of using a graphical model is that we can tune the model structure in a more intuitive way. In the graphical model, each node is a component of the model and each edge is the correlation between the nodes. The arrow means the direction of determination. So the procedure of model design becomes adding or removing nodes or edges. If we consider each component as a set of random variables, then the graph becomes a Bayesian network. Therefore we can use well-developed techniques for Bayesian networks to do inference and interpretation.

The goal of building the generative models is to provide an interface between location proteomics and systems biology, so we have begun implementing generative models in our Protein Subcellular Localization Image Database (PSLID, http://pslid.cbi.cmu.edu). We have also done some preliminary work to convert the models into XML format, which we hope to merge into standard cell modeling descriptions such as SBML (40) and CELLML (41). This will make our models easily transferable between programs. We expect shortly to release software to permit training of models and synthesis of images on a variety of platforms (I. Cao-Berg, T. Zhao, and R.F. Murphy, in preparation). We anticipate a wide applicability of these tools in systems biology studies, especially in simulations of cell behavior that require detailed models for subcellular location.

Acknowledgments

We thank Eric Xing and Geoffrey Gordon for helpful discussions and critical reading of this manuscript.

Literature Cited

1. Ideker T, Galitski T, Hood L. A new approach to decoding life: Systems biology. Annu Rev Genomics Hum Genet 2001;2:343–372.
2. Kitano H. Computational systems biology. Nature 2002;420:206–210.
3. Sauro HM, Hucka M, Finney A, Wellock C, Bolouri H, Doyle J, Kitano H. Next generation simulation tools: the Systems Biology Workbench and BioSPICE integration. OMICS 2003;7:355–372.
4. Faust M, Montenarh M. Subcellular localization of protein kinase CK2. A key to its function? Cell Tissue Res 2000;301:329–340.
5. Ortoleva P, Berry E, Brun Y, Fan J, Fontus M, Hubbard K, Jaqaman K, Jarymowycz L, Navid A, Sayyed-Ahmad A, Shreif Z, Stanley F, Tuncay K, Weitzke E, Wu LC. The karyote physico-chemical genomic, proteomic, metabolic cell modeling system. OMICS 2003;7:269–283.
6. Chou K-C, Elrod DW. Protein subcellular location prediction. Prot Eng 1999;12:107–118.
7. Chou KC, Cai YD. Prediction and classification of protein subcellular location-sequence-order effect and pseudo amino acid composition. J Cell Biochem 2003; 90:1250–1260.
8. Park KJ, Kanehisa M. Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. Bioinformatics 2003;19:1656–1663.
9. Pan YX, Zhang ZZ, Guo ZM, Feng GY, Huang ZD, He L. Application of pseudo amino acid composition for predicting protein subcellular location: Stochastic signal processing approach. J Prot Chem 2003;22:395–402.
10. Chen X, Velliste M, Murphy RF. Automated interpretation of subcellular patterns in fluorescence microscope images for location proteomics. Cytometry Part A 2006; 69A:631–640.
11. Glory E, Murphy RF. Automated subcellular location determination and high throughput microscopy. Developmental Cell 2007;12:7–16.
12. 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;17:1213–1223.
13. Murphy RF, Velliste M, Porreca G. Robust numerical features for description and classification of subcellular location patterns in fluorescence microscope images. J VLSI Signal Process 2003;35:311–321.
14. Velliste M, Murphy RF. Automated determination of protein subcellular locations from 3D fluorescence microscope images. In: Proceedings of the 2002 IEEE International Symposium on Biomedical Imaging, Washington, DC, 7–10 June 2002. pp 867–870.
15. 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. Proc SPIE 2003;4962:298–306.
16. Chen X, Murphy RF. Objective clustering of proteins based on subcellular location patterns. J Biomed Biotechnol 2005;2005:87–95.
17. Hu Y, Carmona J, Murphy RF. Application of temporal texture features to automated analysis of protein subcellular locations in time series fluorescence microscope images. In: Proceedings of the 2006 IEEE International Symposium on Biomedical Imaging, Arlington, VA, 6–9 April 2006. pp 1028–1031.
18. Krause A, Stoye J, Vingron M. Large scale hierarchical clustering of protein sequences. BMC Bioinformatics 2005;6:15.
19. Balaji S, Srinivasan N. Use of a database of structural alignments and phylogenetic trees in investigating the relationship between sequence and structural variability among homologous proteins. Prot Eng 2001;14:219–226.
20. Loew LM, Schaff JC. The virtual cell: A software environment for computational cell biology. Trends Biotechnol 2001;19:401–406.
21. Coggan JS, Bartol TM, Esquenazi E, Stiles JR, Lamont S, Martone ME, Berg DK, Ellisman MH, Sejnowski TJ. Evidence for ectopic neurotransmission at a neuronal synapse. Science 2005;309:446–451.
22. Huang K, Murphy RF. Boosting accuracy of automated classification of fluorescence microscope images for location proteomics. BMC Bioinformatics 2004;5:78.
23. Thomas CH, Collier JH, Sfeir CS, Healy KE. Engineering gene expression and protein synthesis by modulation of nuclear shape. Proc Natl Acad Sci USA 2002;4:1972–1977.
24. Blum H. Biological shape and visual science. J Theor Biol 1973;38:205–287.
25. Tam R, Heidrich W. Shape simplification based on the medial axis transform. In: Proceedings of the 14th IEEE Conference on Visualization, 2003; Seattle, Washington, USA. pp 481–488.
26. Hiransakolwong N, Vu K, Hua KA, Lang S-D. Shape recognition based on the medial axis approach. Proceedings of the 2004 IEEE International Conference on Multimedia Exposition, 2004; Taipei, Taiwan. pp 257–260.
27. Murata S-i, Herman P, Lakowicz JR. Texture analysis of fluorescence lifetime images of AT- and GC-rich regions in nuclei. J Histochem Cytochem 2001;49:1443–1451.
28. Palcic B. Nuclear texture: Can it be used as a surrogate endpoint biomarker? J Cellular Biochem 1994;19(Suppl):40–46.
29. Jørgensen T, Yogesan K, Tveter KJ, Skjørten F, Danielsen HE. Nuclear texture analysis: A new prognostic tool in metastatic prostate cancer. Cytometry 1998; 24:277–283.
30. Zhu SC, Wu Y, Mumford D. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. Int J Comput Vision 1998;27:107–126.
31. Nealen A, Alexa M. Hybrid texture synthesis. In: Proceedings of the 14th Eurographics Workshop Rendering, 2003; Leuven, Belgium. pp 97–105.
32. Portilla J, Simoncelli EP. A parametric texture model based on joint statistics of complex wavelet coefficients. Int J Computer Vision 2000;40:49–71.
33. Lehmussola A, Selinummi J, Ruusuvuori P, Niemistö, Yli-Harja O. Simulating fluorescent microscope images of cell populations. In: Proceedings of the 27 Annual Conference of the IEEE Engineering in Medicine and Biology Society, 2005; Shanghai, China. pp 3153–3156.
34. Cootes TF, Taylor CJ, Cooper DH, Graham J. Active shape models—Their training and application. Comput Vision Image Understanding 1995;61:38–59.
35. Zhao T, Velliste M, Boland MV, Murphy RF. Object type recognition for automated analysis of protein subcellular location. IEEE Trans Image Process 2005;14:1351–1359.
36. Bilmes J. A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussion mixture and hidden Markov models. Berkeley, CA: International Computer Science Institute; 1997. Report nr TR-97–021.
37. Huang K, Velliste M, Murphy RF. Feature reduction for improved recognition of subcellular location patterns in fluorescence microscope images. Proc SPIE 2003;4962:307–318.
38. Garcia Osuna E, Hua J, Bateman N, Zhao T, Berget P, Murphy R. Large-scale automated analysis of protein subcellular location patterns in randomly-tagged 3T3 cells. Ann Biomed Eng 2007;35:1081–1087.
39. Schubert W, Bonnekoh B, Pmmer AJ, Philipsen L, Bockelmann R, Malykh Y, Gollnick H, Friedenberger M, Bode M, Dress AWM. Analyzing proteome topology and function by automated multi-dimensional fluorescence microscopy. Nat Biotechnol 2006;24:1270–1278.
40. Hucka M, Finney A, Sauro H, Bolouri H, Doyle J, Kitano H, Arkin A, Bornstein B, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr JH, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Le Novere N, Loew LM, Lucio D, Mendes P, Minch E, Mjolsness ED, Nakayama Y, Nelson MR, Nielsen PF, Sakurada T, Schaff JC, Shapiro BE, Shimizu TS, Spence HD, Stelling J, Takahashi K, Tomita M, Wagner J, Wang J. The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. Bioinformatics 2003;19:524–531.
41. Lloyd CM, Halstead MDB, Nielsen PF. CellML: Its future, present and past. Prog Biophys Mol Biol 2004;85:433–450.