Cytometry
PART A
Journal of the International Society for Advancement of Cytometry

Image-derived, Three-dimensional Generative Models of Cellular Organization

Tao Peng,[1,2] Robert F. Murphy[1,2,3,4,5]*

[1]Center for Bioimage Informatics, and Department of Biomedical Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, Pennsylvania 15213

[2]School of Computer Science, Lane Center for Computational Biology, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, Pennsylvania 15213

[3]Department of Biological Sciences, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, Pennsylvania 15213

[4]Department of Machine Learning, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, Pennsylvania 15213

[5]School of Life Sciences, Freiburg Institute for Advanced Studies, Albert Ludwig University of Freiburg, Freiburg 79104, Germany

Received 2 December 2010; Revision Received 4 March 2011; Accepted 14 March 2011

Grant sponsor: National Institutes of Health; Grant number: R01 GM 075205;

*Correspondence to: Robert F. Murphy, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

Email: murphy@cmu.edu

Published online 6 April 2011 in Wiley Online Library (wileyonlinelibrary.com)

DOI: 10.1002/cyto.a.21066

Abstract

Given the importance of subcellular location to protein function, computational simulations of cell behaviors will ultimately require the ability to model the distributions of proteins within organelles and other structures. Toward this end, statistical learning methods have previously been used to build models of sets of two-dimensional microscope images, where each set contains multiple images for a single subcellular location pattern. The model learned from each set of images not only represents the pattern but also captures the variation in that pattern from cell to cell. The models consist of submodels for nuclear shape, cell shape, organelle size and shape, and organelle distribution relative to nuclear and cell boundaries, and allow synthesis of images with the expectation that they are drawn from the same underlying statistical distribution as the images used to train them. Here we extend this generative models approach to three dimensions using a similar framework, permitting protein subcellular locations to be described more accurately. Models of different patterns can be combined to yield a synthetic multi-channel image containing as many proteins as desired, something that is difficult to obtain by direct microscope imaging for more than a few proteins. In addition, the model parameters represent a more compact and interpretable way of communicating subcellular patterns than descriptive image features and may be particularly effective for automated identification of changes in subcellular organization caused by perturbagens. © 2011 International Society for Advancement of Cytometry

Key terms

generative models; machine learning; subcellular location; microscope image analysis; cell shape; nuclear shape; location proteomics

Introduction

Proteins function in different cellular or subcellular compartments as part of complex systems. In systems biology, investigating and modeling these complex systems from different aspects and at various levels is hoped to lead to a mechanistic understanding of cell behavior (1,2). Extensive efforts have been made toward proteome-scale determination of protein sequence, structure, abundance and interactions, and tremendous progress has been achieved. Much less information is available about protein location within cells, with descriptions using words (such as GO terms) being the main approach used to represent this important concept. More detailed and comprehensive approaches to learning and describing the spatial distributions of proteins at different levels of accuracy will be critical for systems models.

Development of modern microscopy technology makes observation of protein localization possible both in vitro and in vivo with high throughput. However, traditional visual analysis to recognize protein localizations can be a key barrier for converting large sets of images to useful descriptions of protein locations. To overcome this difficulty, machine learning methods and digital image processing tools have been combined to develop systems that automatically recognize protein subcellular location patterns (3). Here "pattern" designates the subcellular distribution of a pro-

ISAC
International Society for Advancement of Cytometry

tein or of a set of proteins whose distributions are statistically indistinguishable. The most critical component of these systems is sets of numerical features to describe protein subcellular location patterns in 2D or 3D images. With these features, the feasibility of classifying major protein subcellular location patterns with high accuracy and efficiency compared with visual analysis has been demonstrated (4,5).

However, recognition of location patterns provides only limited information. For example, describing a protein location as "nucleus" in a given cell type under a given condition provides no detail on how it is distributed within the nucleus (and of course no information on the size or shape of nuclei in that cell type). Similarly, recognition based approaches can describe a protein's "relocation from organelle A to B" but communicate no information about how this process happens spatially and geometrically. Thus, beyond simply recognizing subcellular location patterns, an important goal is to be able to build models to capture the essence and variation of a specific pattern.

Zhao and Murphy (6) describe the first system for constructing generative models of subcellular patterns in 2D images, providing a framework in which cell structure and subcellular location patterns can be represented and communicated. In this work, images are viewed as the manifestation of a set of random variables and image synthesis or generation is viewed as a stepwise random process. A statistical generative model is the combination of all distributions of these random variables. Building a generative model for images in the form of a joint distribution of all pixels in an image is too computationally expensive (and potentially underdetermined) to be practical. Therefore, methods of computational geometry and data analysis were explored to compromise between complexity and accuracy of the model. 2D fluorescence images of cells were represented by three major components: nucleus, cell membrane, and protein objects distributed inside these compartments. All three components were represented by small sets of parameters (much fewer than the number of image pixels) from which the key features related to protein locations in the original image can be reconstructed with reasonable accuracy. The three components were modeled conditionally on each other, e.g., the output of the model of organelle position takes as inputs the instances drawn from models of cell and nuclear shape.

While this initial approach is useful for the vast majority of fluorescence microscope images that are acquired in only 2D, they represent a significant simplification of actual cell organization in 3D. Therefore, in this article, we describe extending the generative modeling and simulation framework to 3D.

Materials and Methods

For the studies described here, we used the same 3D HeLa dataset (5) used previously for testing 2D models (the previous studies used only the central slide of each 3D stack). The dataset is available at http://murphylab.web.cmu.edu/data and contains three fluorescent channels for each field: a DNA channel that reflects nuclear shape and chromosome texture; a total protein channel that reflects cell shape; and an antibody channel that reflects the distribution of a particular protein. Each field has been previously segmented into single cell regions using a seeded watershed approach. The entire 3D stacks of 447 images were included to build the 3D nuclear and cell shape models. Protein location models were created for four "vesicle-like" protein location patterns—lysosomes, mitochondria, endosomes, and nucleoli, each with around 50 training 3D image stacks. Each stack contains 14 to 29 slices containing 1024 × 1024 voxels. The voxel spacing is 0.049 μm × 0.049 μm × 0.203 μm.

The 3D generative model algorithm used in this work was built upon the toolbox for 2D generative models (available from http://murphylab.web.cmu.edu/software) and implemented using MATLAB (version 7.8.0). Code and trained models will be available upon publication from the same site. Methods and algorithms to create each component of the model are described in Results. Before the modeling step, each image slice was thresholded using the Ridler-Calvard method (7). This was done rather than using a global threshold because lower slices in the 3D stacks are subject to photobleaching. Nuclear and cell shapes were rotated to have the major axis aligned using a principal axis alignment method (8). The Matlab spline toolbox (version 3.3.6) was used for nuclear shape modeling. The EM code used in protein object modeling from the NETLAB library (http://www.ncrg.aston.ac.uk/netlab) was modified to estimate weighted Gaussian mixtures with interval inputs (instead of points).

Results

Spline Surface Model of 3D Nuclear Shape

We begin by building a 3D nuclear shape model. In the previous 2D model, a nuclear shape was represented by curves describing the medial axis and the width along it. These two curves were each parametrically approximated by fourth order B-spline curves. The medial axis representation has also been applied in modeling 3D shapes, such as pancreas (9). However, it is not easy to find a concise and proper representation for 3D nuclear shape with this method, especially for nuclei of cultured cells, which are usually flat (Fig. 1a).

We therefore consider the 3D shape of a nucleus to be described by a parametric surface $[x(\varphi,z),y(\varphi,z),z(\varphi,z)]$. The definition becomes much clearer in a cylindrical coordinate system

$$\begin{cases} x(\varphi,z) = r(\varphi,z)\cos\varphi \\ y(\varphi,z) = r(\varphi,z)\sin\varphi \\ z(\varphi,z) = z \end{cases} \quad \begin{matrix} 0 \le \varphi < 2\pi \\ z_{\min} \le z \le z_{\max} \end{matrix} \qquad (1)$$

The cylindrical representation is an "unwrapping" of the side surface of a nuclear shape to a surface function $r(\varphi,z)$ with rectangular support (Fig. 1b). The conversion from Cartesian to cylindrical coordinate systems included resampling of the digitized image.

As previously done for 2D models, we used splines to parameterize the 3D nuclear shape as they were observed to

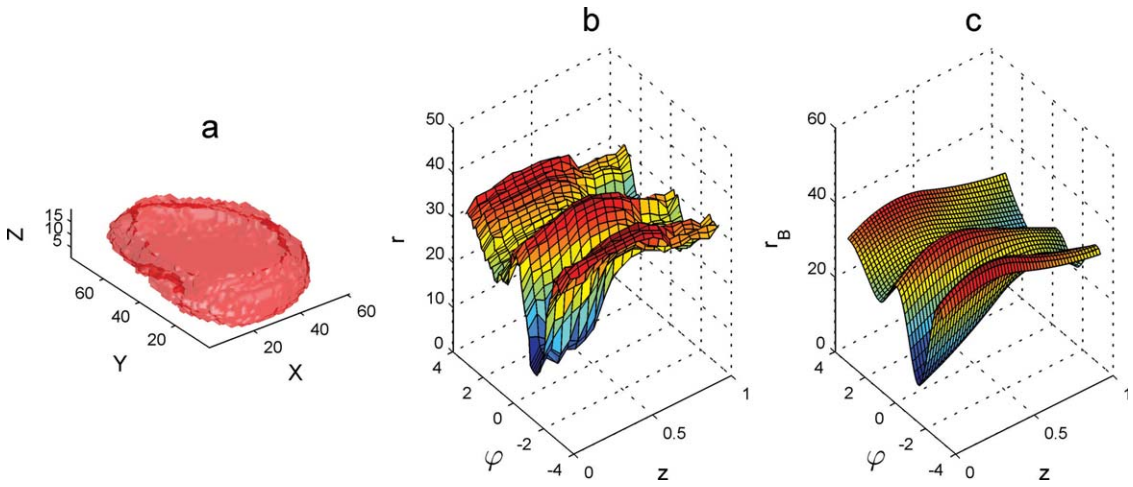

Figure 1. Nuclear shape representation. **(a)** Surface plot of a 3D HeLa cell nucleus. **(b)** Unfolded surface of the nuclear shape in a cylindrical coordinate system. The surface plot shows the radius r as a function of azimuth φ and height z. **(c)** B-spline surface fitted to the unfolded nuclear surface.

give very good fits. A tensor product spline surface was used to fit the surface function $r(\varphi,z)$.

$$r(\varphi, z) \sim r_B(\varphi, z) = \sum_{i=0}^{m} \sum_{j=0}^{n} \tau_{i,j} N_{i,p}(\varphi) N_{j,q}(z) \qquad (2)$$

where m and n are the number of control points for φ and z, respectively.

In the tensor form, $N_{i,p}(\varphi)$ and $N_{j,q}(z)$ are B-spline basis functions of degree p and q, respectively; $\tau_{i,j}$ are the control points or coefficients of the basis functions. Parameterization of the surface function $r(\varphi,z)$ is achieved by least-square approximation

$$\tau = \arg\min_{\tau} \sum_{i=0}^{N_1} \sum_{j=0}^{N_2} \left[r_B(\varphi_i, z_j) - r(\varphi_i, z_j) \right]^2 \qquad (3)$$

where N_1 and N_2 are the number of angles and slices in the digitized image, respectively. $r_B(\varphi_i,z_j)$ is the radius calculated from the fitted spline surface function at (φ_i,z_j), where real value $r(\varphi_i,z_j)$ is observed. To achieve good fit with reasonable complexity, and to eliminate the effect of high frequency digitization artifacts, we pick the orders of the B-spline basis functions as $p = 4$ and $q = 3$ for the HeLa nuclei (Fig. 1c), with $m = 8$ and $n = 3$. We observed the fitted values of the internal knot points to be around (1/4,3/8,1/2,5/8,3/4) along the azimuth coordinate and 1/2 along the height coordinate and to have little contribution to the variation of the spline surface. The positions were therefore set to be constants.

Fitting the "unwrapped" surface ignores the continuous condition at $\varphi_0 = 0$ and $\varphi_{N_1} = 2\pi$.

$$\frac{\partial^l}{\partial \varphi^l} r_B(\varphi_0, z) = \frac{\partial^l}{\partial \varphi^l} r_B(\varphi_{N_1}, z), \quad \forall z \in [z_0, z_{N_2}], l = 0, \ldots, p-1 \qquad (4)$$

Fujioka and Kano (10) have proved that continuity is maintained if and only if the coefficients τ satisfies

$$\tau_{0,j} = \tau_{m,j} \qquad j = 0, \ldots, n \qquad (5)$$

We append these constraints to the least-square approximation problem in Eq. (3) to guarantee periodicity. This reduces the number of free parameters from $(m + 1) \times (n + 1) = 36$ to 32. Adding a parameter for the height, each nuclear shape is therefore represented by a total of 33 free parameters. All nuclei are rotated and mirrored (if necessary) according to the central slice so that they all have same "elongating" and "bending" direction. Statistical learning is then performed on the shape parameters extracted from these nuclei images to describe the variation of the shapes from nucleus to nucleus. We chose a multivariate normal distribution to model the variation of the parameters. As shown in Figure 2, comparison of the data empirical distribution with the fitted normal distribution validates this choice. The nearness of each plot to the diagonal line indicates close agreement to theoretical normal distributions. The mean and median of p -values under Kolmogorov-Smirnov normality tests for all parameters are 0.28 and 0.19, respectively, also confirming the choice. Thus, we conclude that the nuclear shape of HeLa cells can be captured by a statistical model with 562 values: a length 32 vector τ for the means of the spline coefficients and the unique elements of the symmetric 32×32 covariance matrix, plus the mean and standard deviation of the height.

To synthesize a nuclear shape, we draw the parameter values from the trained distributions and then construct a new shape from the parameters.

Eigen Shape Model of Cells

The next step is to construct a model of the cell shape (or plasma membrane). Compared with the nuclear boundary, the cell boundary has more local morphological flexibility and much larger variation from cell to cell, making it hard to be parameterized to a statistically simple representation.

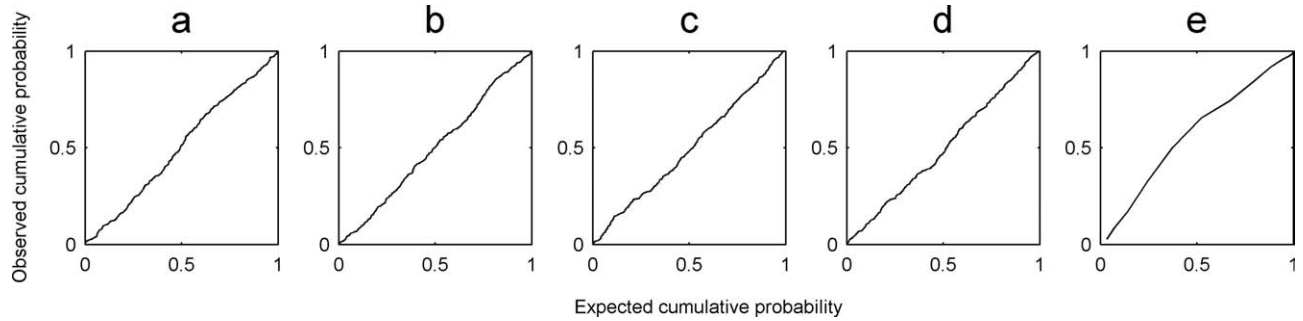

Figure 2. Normality plots of nuclear shape parameters. **(a–d)** P-P plot of randomly selected spline surface coefficients (empirical cdf versus fitted normal cdf). **(e)** P-P plot of the nuclear height.

However, it is obvious that the cell shape is conditioned on the nuclear shape (at least the nucleus is inside the cell). Instead of building a parametric cell shape model and a correlation model separately, we adopt the approach previously used in the 2D model: learning the ratio of the radius of the cell to that of the nucleus. As for the nucleus, we first define a polygonal representation of the cell shape in a cylindrical coordinate system, denoted as $r_c(\varphi,z)$. The radius ratios are then expressed as a function of φ and z: $R(\varphi,z) = r(\theta,z)/r_c(\theta,z)$. We sampled φ over 360° in 1 degree increments, and sampled z over 18 slices. This was the average number of slices per cell; all cells were resampled to this value along the z dimension. The ratio representation contains $360 \times 18 = 6{,}120$ values.

Direct statistical estimation is impossible for the 6,120-dimensional vector with only hundreds of samples and is not necessary to guarantee accuracy. Instead, we applied principal component analysis (PCA) to the ratio vectors after centering them by subtraction of the mean vector. Instead of performing PCA directly on the original covariance matrix, we use a modified PCA method to find significant modes and coefficients quickly by applying eigen analysis on $\frac{1}{N} R^T R$, where R is an N-column matrix with each column a centered ratio vector. This modified PCA method is a more efficient approach for eigen decomposition with far more original data dimension than number of samples, based on previous discussions (11,12).

With 20 most significant principal modes, the shape ratio can be reconstructed substantially (Fig. 3). Moreover, the coefficients λ_{ij} of each mode appear to be normally distributed and independent from each other (Fig. 4). The mean and median of p-values under Kolmogorov-Smirnov normality tests for all parameters were 0.29 and 0.22, respectively. The cell shape model then contains 21 6,120-dimensional constant vectors for the mean and principal modes, and 20 variances of the coefficients (the coefficient means are zero).

To generate an instance of cell shape, we first synthesize an instance of nuclear shape as described in the previous section. We then sample values for each principal component from its normal distribution, generate centered ratio vectors

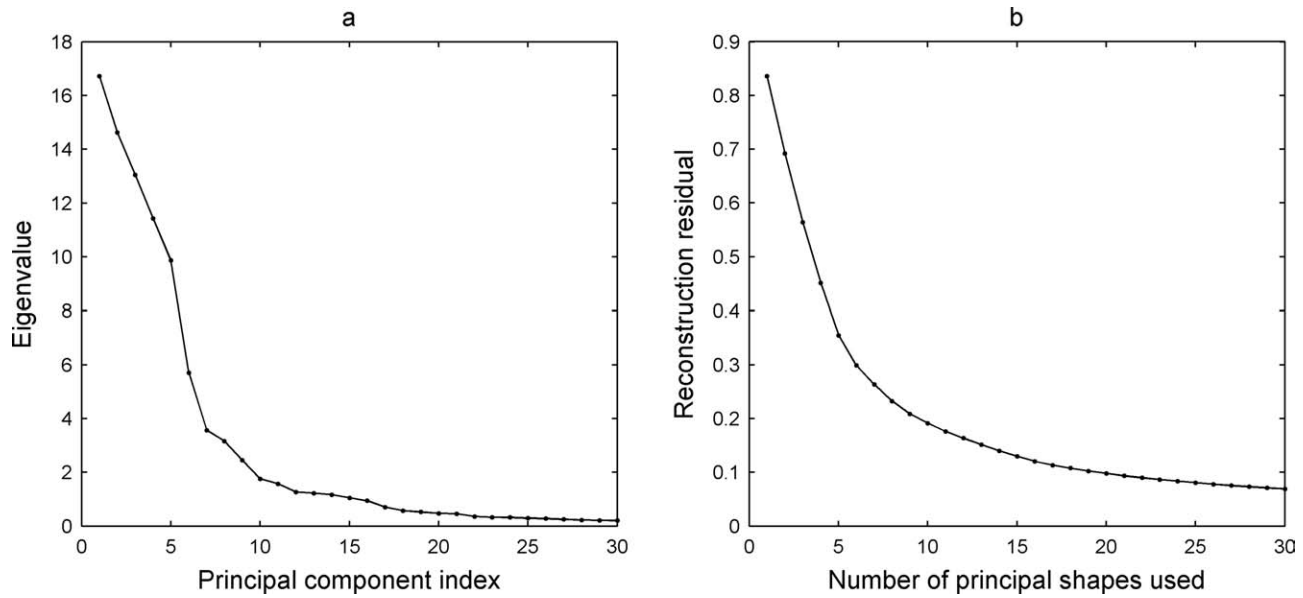

Figure 3. PCA on cell shape representation. **(a)** The eigen value spectrum (truncated to the first 30). **(b)** Reconstruction residual of cell shapes as a function of number of principal modes used. The residual is the ratio of the sum of eigen values of not included modes to the sum of all eigen values. With 20 modes, the residual drops under 0.1.

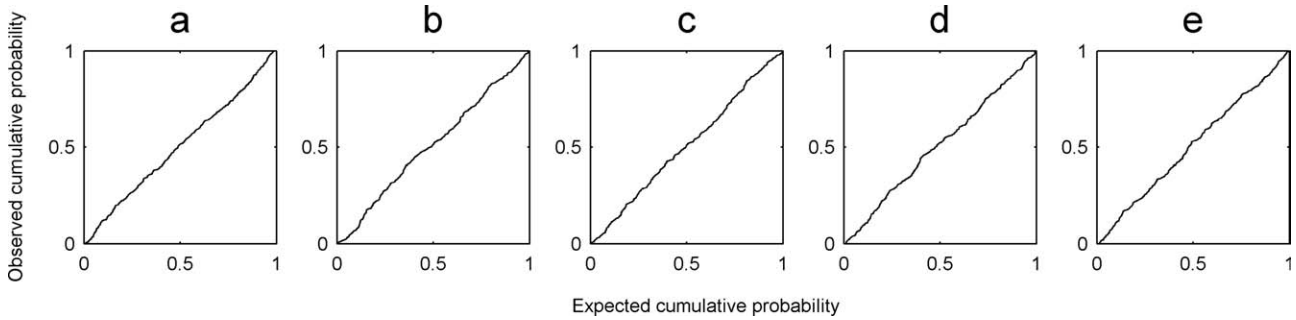

Figure 4. Normality plots of nuclear shape parameters. **(a–d)** P-P plot of randomly selected spline surface coefficients (empirical cdf versus fitted normal cdf). **(e)** P-P plot of the nuclear height.

by multiplying by the saved principal modes, and add back the mean ratio vector. The result has a fixed height of 18 slices. We next sample a cell height from its normal distribution, stretch the synthesized nuclear shape to correspond to this height, apply the ratios at every (θ,z), and form a cell shape by connecting line segments between neighboring surface points. The cell shape is stretched back so that the nuclear height corresponds to its original height. Note that this assumes that the nucleus and cell cover the same number of slices; this was observed to be the case for HeLa cells which have little cytoplasmic space above or below the nucleus (relative to the slice thickness).

Protein Object Model

Object-based protein pattern representation. Protein subcellular location patterns have been successfully represented using "objects" in a number of previous studies, such as location pattern recognition and complex pattern unmixing (4,13–15). "Objects" are defined as contiguous regions of nonzero pixels. Many cell organelles have roughly ellipsoidal

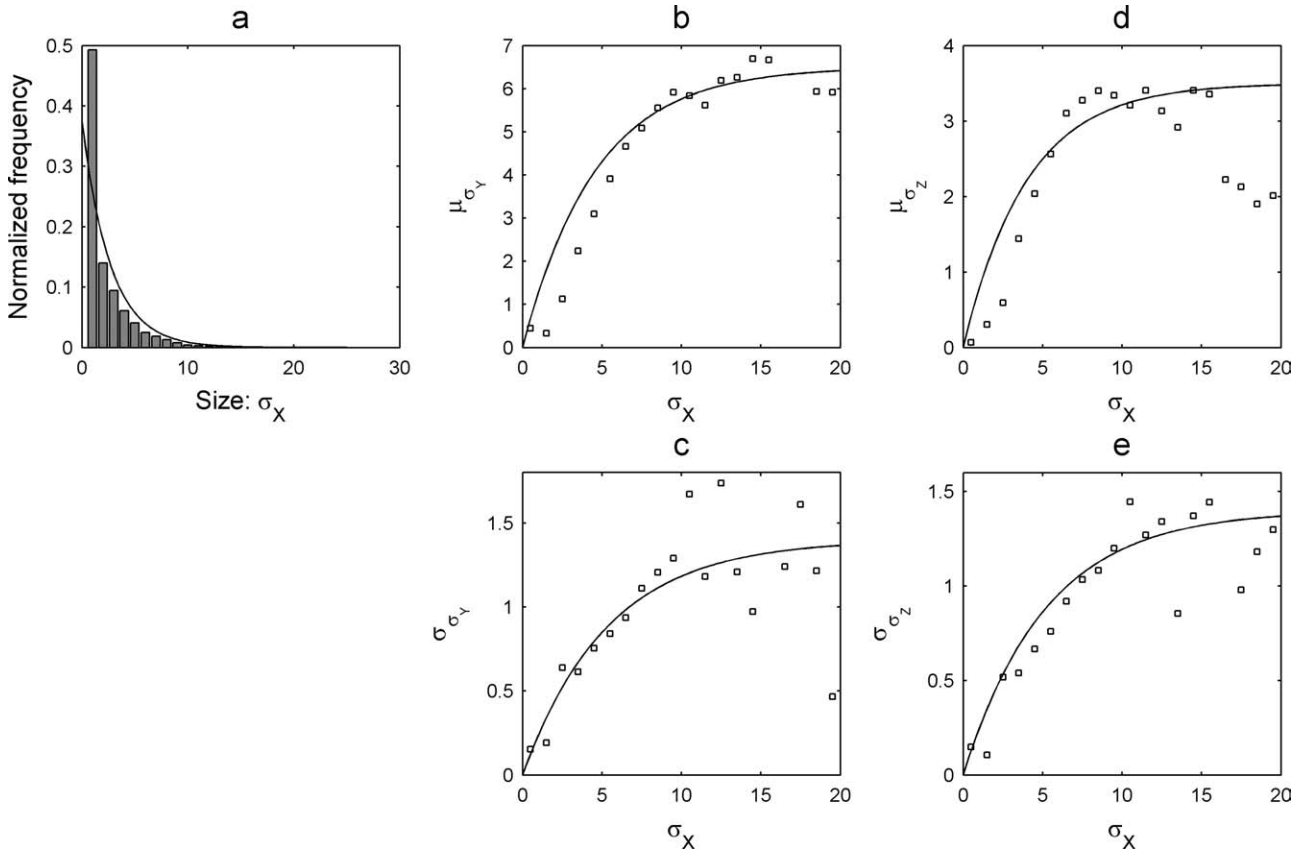

Figure 5. Fitting the distributions of object size parameters. **(a)** The histogram of the Gaussian object parameter σ_X is shown, along with an exponential fit (solid line) to the distribution. **(b,d)** Conditional mean of σ_Y **(b)** and σ_Z (d) as a function of σ_X (squares), and results of parametric fitting (solid line). **(c,e)** Conditional standard deviation of σ_Y (c) and σ_Z (e) as a function of σ_X (squares), and results of parametric fitting (solid line).

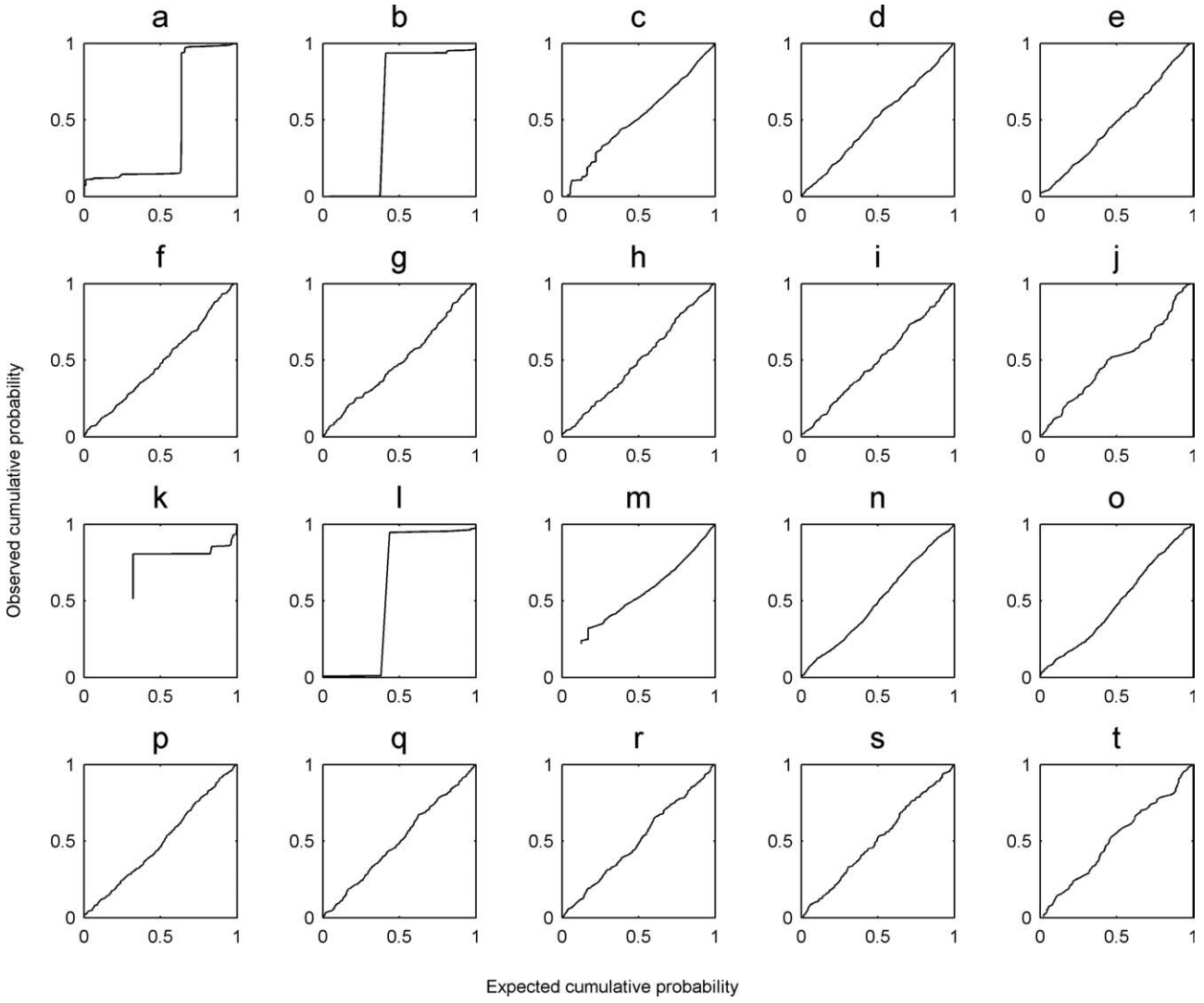

Figure 6. Normality plots of the conditional distributions of object dimensions. **(a–j)** P-P plot of σ_Y versus corresponding σ_X for different ranges of x in increments of 0.1. **(k–t)** P-P plot of σ_Z with corresponding σ_X in different ranges. The plots for the first two ranges (a,b,k,l) deviate from the diagonal because in these intervals σ's are rather small, approaching the voxel limit.

shapes and appear to be ellipsoids with intensity clustered in the center in the images. Zhao and Murphy (6) therefore used 2D Gaussian distribution functions to fit objects in 2D images and modeled each ellipsoidal family as the distribution of the Gaussian distribution parameters. In this article, we also focus on modeling protein location patterns consisting of mostly ellipsoidal objects in 3D images by extending the methods of Gaussian objects learning into 3D.

As in 2D images, ellipsoidal objects in 3D images may aggregate to form larger objects to which a single Gaussian gives a poor fit. Using the expectation-maximization (EM) algorithm, we can estimate parameters of each Gaussian component in an aggregated object as a Gaussian mixture. The probability density function (PDF) of a Gaussian mixture distribution can be written as

$$g_m(\mathbf{x}) = \sum_{k=1}^{m} p_k g(\mathbf{x}|\mu_k, \Sigma_k) \tag{6}$$

where $g(\mathbf{x}|\mu_k,\Sigma_k)$ is a Gaussian PDF over three variables $\mathbf{x} = (x,y,z)$ with mean μ_k and covariance matrix Σ_k, and p_k is a probabilistic weight of the k th Gaussian component in the mixture ($\sum_{k=1}^{m} p_k = 1$).

As in the 2D case, we weight each data point with the intensity value of the voxel ($w_i = I(x_i,y_i,z_i)$). The E-step of the algorithm to fit the Gaussian mixture stays the same for 3D objects. $\alpha_i k$ is the expected probability that point $\mathbf{x}_i$ is associated with component k. The superior (t) added to the mixture model PDF parameters means the values at step t of the iterative EM process. The Gaussian parameter for each object is fitted at convergence.

E-step:

$$\alpha_{ik} = \frac{p_k^{(t)} g\left(\mathbf{x}_i|\mu_k^{(t)}, \Sigma_k^{(t)}\right)}{\sum_{k=1}^{m} p_k^{(t)} g\left(\mathbf{x}_i|\mu_k^{(t)}, \Sigma_k^{(t)}\right)} \tag{7}$$

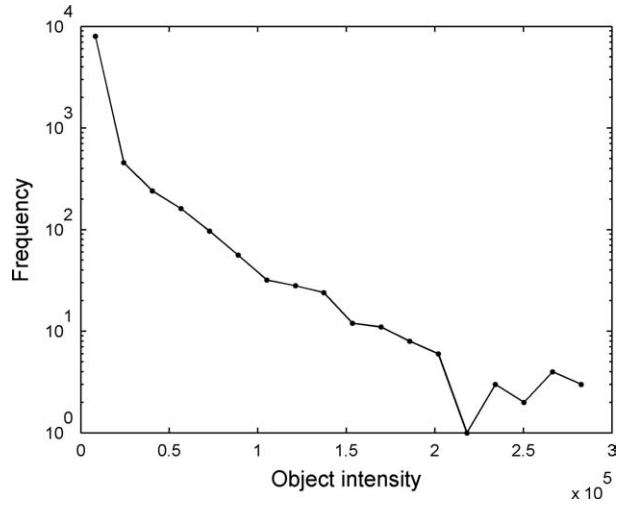

Figure 7. Distribution of the total intensity of Gaussian objects. The points are close to a straight line, supporting the use of an exponential distribution to represent the distribution.

In the previous 2D model, pixels were treated as points. However, taking account of the digitization effect, the pixels and voxels are better treated as intervals. Therefore, the Gaussian parameters estimation in the M-step should be slightly changed. The variances should increase by 1/12 to avoid zero variance at certain dimensions (1/12 is the variance of a standard continuous uniform distribution).

M-step:

$$p_k^{(t+1)} = \frac{1}{\sum_{i=1}^{N} w_i} \sum_{i=1}^{N} w_i \alpha_{ik} \tag{8}$$

$$\mu_k^{(t+1)} = \frac{1}{\sum_{i=1}^{N} w_i \alpha_{ik}} \sum_{i=1}^{N} w_i \alpha_{ik} \mathbf{x}_i \tag{9}$$

$$\Sigma_k^{(t+1)} = \frac{1}{\sum_{i=1}^{N} w_i \alpha_{ik}^{(t)}} \sum_{i=1}^{N} w_i \alpha_{ik}^{(t)} \left(\mathbf{x}_i - \mu_k^{(t+1)}\right)\left(\mathbf{x}_i - \mu_k^{(t+1)}\right)^T + \tfrac{1}{12}\mathbf{I} \tag{10}$$

We then perform statistical learning, mostly distribution fitting with maximum likelihood estimation, on the decomposed Gaussian function parameters. The centers (mean) of the Gaussian objects are used to learn the object position model, which is described in the next section. The covariance matrix Σ is directly related to the size of the object and we use it to train an object size model. The ellipsoid Gaussian objects are rotated to have their major and minor axis aligned to the Cartesian coordinates (the rotation transformation can be achieved by eigen decomposition of Σ). Each object is then represented in size by three standard deviations σ_X, σ_Y, and σ_Z in descending order. As the variances along different coordinates are highly dependent, we use a simple Bayesian structure to fit their distribution: $\sigma_Y \leftarrow \sigma_X \rightarrow \sigma_Z$. Figure 5a shows standard deviation on the major axis is well fit by an exponential distribution. Conditional distribution of σ_Y or σ_Z over σ_X are also fit by normal distributions (normality of the conditional distributions of $P(\sigma_Y|\sigma_X)$ or $P(\sigma_Z|\sigma_X)$ is shown in Fig. 6). The mean and median of *P*-values under Kolmogorov-Smirnov normality tests for all parameters are both 0.37. Figures 5b–5e show the dependency of the normal parameters on σ_X, along with their parametric fitting by

$$\begin{cases} \bar{\sigma}_{Y(Z)} = \alpha_1\left(1 - e^{-\beta_1 \sigma_X}\right) \\ \mathbf{stdev}\left(\sigma_{Y(z)}\right) = \alpha_2\left(1 - e^{-\beta_2 \sigma_X}\right) \end{cases} \quad \sigma_{Y(z)} \le \sigma_X \tag{11}$$

The other important parameter to describe a single Gaussian object is its intensity. It is defined to be the coefficient between fitted object and the Gaussian function, which is normalized. γ_i for an object is estimated from the product of total intensity of the aggregated object it resides in and the fraction this object in the whole mixture $I_i(\mathbf{x}) = \gamma_i \cdot g(\mathbf{x}_i)$. Again, we choose an exponential distribution to fit the intensity coefficient γ_i (Fig. 7).

To synthesize an individual Gaussian object, we simply generate the σ, γ values and use them to synthesize a 3D Gaussian function $\gamma \cdot g(\mathbf{x}|\mathbf{0},\Sigma)$. The Gaussian function is digitized and added to the 3D image at a specific position as determined below.

Probabilistic protein location model. We modeled the positions of protein objects relative to the nuclear and plasma membranes. The model we used here is a direct extension of the 2D protein object position model used previously.

We parameterize the position of each object by three variables: *s*, ratio of the distance of a given object's center to the nuclear surface over the sum of that distance and the distance to the cell surface; and θ and φ, the inclination angle and azimuth angle of the vector from the nuclear center to the object center. Using this parameterization, a distribution was formed in which all points where an object occurs were set to 1 and all other positions set to zero. After normalizing for the total number of objects, the potential (the probability that a given

Table 1. Position parameter β for different location patterns

	β_0	β_1	β_2	β_3	β_4	β_5
Lysosome	−5.7032	1.2905	−3.3514	0.1812	0.3240	−1.2133
Mitochondria	−4.3154	1.8065	−2.6727	0.0666	−0.0431	−0.0142
Nucleolus	−13.8989	−26.6740	−24.2650	0.1145	−0.1033	−1.2589
Endosome	−4.5109	0.4816	−0.7073	0.0336	−0.0010	−1.1416

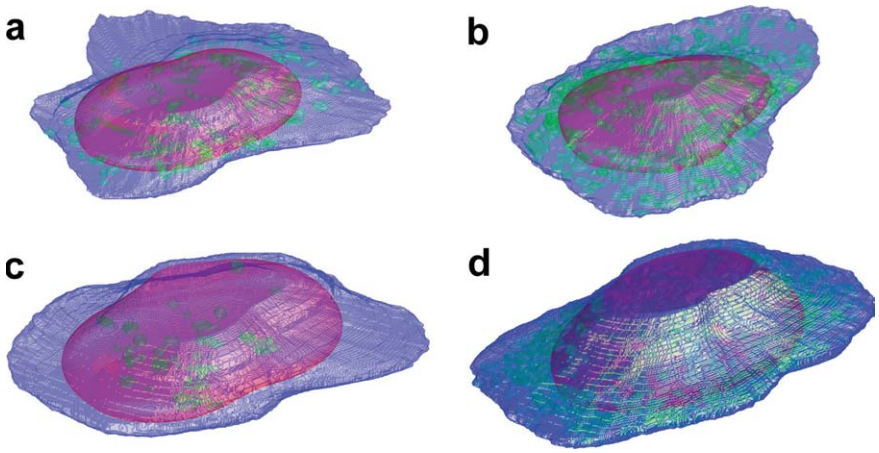

Figure 8. Synthesized images displayed in pseudo color surfaces for different protein location patterns (green), with nuclear (red) and cell shapes (blue). **(a)** Lysosome, **(b)** Mitochondria, **(c)** Nucleolus, and **(d)** Endosome.

position is the center of an object) can be fitted by a logistic regression function

$$P(s,\varphi,\theta)=\frac{e^{\beta_0+\beta_1 s+\beta_2 s^2+\beta_3\cos\varphi\sin\theta+\beta_4\sin\varphi\sin\theta+\beta_5\cos\theta}}{1+e^{\beta_0+\beta_1 s+\beta_2 s^2+\beta_3\cos\varphi\sin\theta+\beta_4\sin\varphi\sin\theta+\beta_5\cos\theta}} \quad (12)$$

Here, the parameters of β have different interpretations for object location "preference." For example, β_1 shows whether the protein is more likely to distribute near the nucleus or near the cell membrane. β_3, β_4, and β_5 determines the protein angular preference. Table 1 shows learned β for different location patterns. As expected, nucleoli show a preference to be inside the nuclear membrane (negative β_1).

At this point, the only aspects of the model that are not statistically modeled are the number of Gaussian objects and the direction of alignment of each of them to the central axis. These parameters did not show an easily fitted distribution (data not shown) and therefore for image generation values for them are randomly sampled from the empirical distributions. Given synthesized nuclear and cell shapes, Gaussian objects and positions, we can synthesize full images depicting protein subcellular location patterns. Figure 8 shows synthesized location images with three channels of the four patterns. Additional images can be generated using the tools and models available at http://murphylab.web.cmu.edu/software.

Discussion

The work described here enables significantly more accurate and realistic descriptions and simulations than prior work. Nonetheless, much further work is required to further refine and extend these models. In particular, methods are needed to capture patterns not well represented by discrete objects. Work on learning generative models for microtubule distributions represents an initial step in this direction (16).

As in any parametric model, our approach includes a number of choices for what parameters to use and how to capture their distributions (mostly using normal and exponential distributions). Their generalizability to new cell types is therefore unknown. For such applications in the future, it will therefore be important to revisit these choices, and perhaps ultimately to make them using large collections of images for many cell types. In cases where parametric models do not perform well, alternatives such as the instance-based diffeomorphic shape generation models described previously (17) may be considered.

With our approach, images can be generated that appear to the eye to capture the essential features of protein location patterns. These synthesized images are suitable for simulations (using systems like Virtual Cell (18) and MCell (19)), and their utility can ultimately be evaluated (and hopefully increased) through simulations using them to predict cell behaviors. It is important to note that in line with their anticipated use in simulations, our models are abstractions not intended to generate images that appear similar to actual microscope images. However, it is a straightforward matter to generate such images by convolution of the idealized image with the point-spread function for a particular microscope (20).

While our 2D and 3D systems are the only ones described to date that construct generative models of organelles within the context of cell and nuclear organization, other investigators have described approaches for modeling or simulating nuclear or organelle size and shape (20,21). Models resulting from such work can readily be incorporated into the conditional model framework we have described.

Literature Cited

1. Ideker T, Galitski T, Hood L. A new approach to decoding life. Annu Rev Genomics Hum Genet 2001;2:343–372.
2. Kitano H. Computational systems biology. Nature 2002;420:206–210.
3. Glory E, Murphy RF. Automated subcellular location determination and high throughput microscopy. Dev Cell 2007;12:7–16.
4. Boland MV, Murphy RF. A neural network classifier capable of recongnizing all major subcellular structures in fluorescence microscope images of hela cells. Bioinformatics 2001;17:1213–1223.
5. Velliste M, Murphy RF. Automated determination of protein subcellular locations from 3D fluorescence microscope images. Washington, DC: Proc IEEE Int Symp Biomed Imaging 2002; pp 867–870.
6. Zhao T, Murphy RF. Automated learning of generative models for subcellular location: building blocks for systems biology. Cytometry Part A 2007;71A:978–990.

7. Ridler TW, Calvard S. Picture thresholding using an iterative selection method. IEEE Trans Syst Man Cybernet 1978;SMC-8:630–632.
8. Huang K, Murphy RF. Boosting accuracy of automated classification of fluorescence microscope images for location proteomics. BMC Bioinform 2004;5:78.
9. Fletcher PT, Lu C, Pizer SM, Joshi S. Principal geodesic analysis for the study of nonlinear statistics of shape. IEEE Trans Med Imaging 2004;23:995–1005.
10. Fujioka H, Kano H. Periodic smoothing spline surface and its application to dynamic contour modeling of wet material objects. IEEE Trans Syst Man Cybernet Part A 2009;39:251–261.
11. Turk MA, Pentland AP. Eigenfaces for recognition. J Cogn Neurosci 1991;3:71–86.
12. Cootes TF, Taylor CJ, Cooper DH, Graham J. Active shape models: Their training and application. Comput Vis Image Underst 1995;61:38–59.
13. Zhao T, Velliste M, Boland MV, Murphy RF. Object type recognition for automated analysis of protein subcellular location. IEEE Trans Image Proc 2005;14:1351–1359.
14. Peng T, Bonamy GM, Glory E, Rines DR, Chanda SK, Murphy RF. Determining the distribution of probes between different subcellular locations through automated unmixing of subcellular patterns. Proc Natl Acad Sci USA 2010; 107:2944–2949.
15. Coelho LP, Peng T, Murphy RF. Quantifying the distribution of probes between subcellular locations using unsupervised pattern unmixing. Bioinformatics 2010;26:i7–i12.
16. Shariff A, Murphy RF, Rohde GK. A generative model of microtubule distributions, and indirect estimation of its parameters from fluorescence microscopy images. Cytometry Part A 2010;77A:457–466.
17. Peng T, Wang W, Rohde GK, Murphy RF. Instance-based generative biological shape modeling. Boston, MA: Proc IEEE Int Symp Biomed Imaging 2009; pp 690–693.
18. Loew LM, Schaff JC. The virtual cell: A software environment for computational cell biology. Trends Biotechnol 2001;19:401–406.
19. Stiles JR, Bartol TM Jr, Salpeter EE, Salpeter MM. Monte carlo simulation of neuro-transmitter release using mcell, a general simulator of cellular physiological processes. In: Bower JM, editor. Proc Comput Neurosci 1998; NY: Plenum Press, pp 279–284.
20. Svoboda D, Kozubek M, Stejskal S. Generation of digital phantoms of cell nuclei and simulation of image formation in 3d image cytometry. Cytometry Part A 2009;75A: 494–509.
21. Helmuth JA, Burckhardt CJ, Greber UF, Sbalzarini IF. Shape reconstruction of subcellular structures from live cell fluorescence microscopy images. J Struct Biol 2009; 167:1–10.