

Geometries for Spatial Cell Simulations

Prepared for "Modeling Biomolecular Site Dynamics: Methods and Protocols" in Springer's Methods in Molecular Biology series (W. S. Hlavacek, ed.)

CellOrganizer: Learning and Using Cell Geometries for Spatial Cell Simulations

Timothy D. Majarian^{1,2#}, Ivan Cao-Berg¹, Xiongtao Ruan¹, and Robert F. Murphy^{1-4*}

¹Computational Biology Department, ²Department of Biological Sciences, ³Department of Biomedical Engineering, ⁴Machine Learning Department, Carnegie Mellon University

*Corresponding Author: murphy@cmu.edu

#Present address: Broad Institute of MIT and Harvard

Summary:

This chapter describes the procedures necessary to create generative models of the spatial organization of cells directly from microscope images and use them to automatically provide geometries for spatial simulations of cell processes and behaviors. Such models capture the statistical variation in the overall cell architecture as well as the number, shape, size and spatial distribution of organelles and other structures. The different steps described include preparing images, learning models, evaluating model quality, creating sampled cell geometries by various methods, and combining those geometries with biochemical model specifications to enable simulations.

Key words: generative model, spatial organization, biochemical simulation

1. Introduction:

A major goal of systems biology is to describe the functional network of dynamic protein interactions within a given cell and perform subsequent simulations on that network. To this end, toolkits that support biochemical reaction models such as MCell, Smoldyn, and VCell have been developed [1-3]. Lacking from these models are complex, statistically accurate spatial information; they rely heavily on simplified or tediously generated compartment models, often simple geometries, that may not represent the true morphological and spatial heterogeneity within a cell population. The accuracy and generalizability of these biochemical models could be greatly improved by utilization of learned cell geometries given that *in vivo* spatial organization of proteins and their containing structures ultimately influences network dynamics [4-6]. Here we describe protocols using the CellOrganizer platform on Galaxy, a system for learning generative models of cell organization and geometry directly from microscopy images in an easy to use, interactive graphical interface. CellOrganizer can build models that capture the statistical variation in key aspects of cell morphology and organelle distribution. It allows for synthesis of realistic, representative cell geometries in various importable formats useful for subsequent biochemical simulation.

Modeling in CellOrganizer begins with a collection of cellular images, usually fluorescently tagged for a number of proteins. Once cell regions are identified and segmented, each cell image is parameterized depending on the model type. For nuclear and cell shape, three main model classes exist in CellOrganizer: diffeomorphic, PCA and medial axis/ratio models. Vesicular organelles are modeled through Gaussian mixture models, while cytoskeletal components are modeled through a growth-based network model. See [7-12] for in-depth descriptions of each model.

CellOrganizer is primarily software that requires users to have a working knowledge of Matlab that may be prohibitive to those without basic programming acumen. With the aim of developing approachable computational tools for image analysis, we recently deployed CellOrganizer through Galaxy, a widely-used workflow management system for data-driven biomedical research [13]. Galaxy provides a clean, convenient web-based graphical user interface that allows users to upload their own data, choose specific tools, design workflows, setup parameters and run pipelines automatically [13]. Moreover, all analysis is performed in the server we provide, eliminating the need for users to install Matlab locally.

Here we provide protocols for building modeling pipelines for cellular image analysis, including all steps from uploading image datasets to the Galaxy server to downloading synthetic geometries reflecting the original cell population.

2. Materials

Prior to using CellOrganizer for Galaxy, users will need to obtain an account on a Galaxy server that has both CellOrganizer and the Galaxy tools installed. If your

Geometries for Spatial Cell Simulations

institution has a Galaxy server, you can contact the administrator to ask about installing CellOrganizer and obtaining an account (instructions can be found at <http://www.cellorganizer.org/galaxy>).

Alternatively, you can find a list of Galaxy servers that support CellOrganizer at <http://www.cellorganizer.org/galaxy-servers>.

3. Methods

3.1. Image Preparation

1. CellOrganizer requires images in OME-TIFF format that contain single cells. The OME format is composed of two parts: pixel data and metadata. The pixel data includes all image channels while the metadata contains descriptors and properties of the image. Most metadata fields are optional, however, the pixel length in the sample plane must be specified in order to use images with CellOrganizer. Once cell images are collected, they must be segmented and converted to the OME-TIFF format.
2. Segmentation can be performed with built in tools for Matlab and ImageJ among others. Seeded watershed algorithms are usually the preferred methods for cell segmentation with both Matlab and ImageJ.
3. To use ImageJ, download and install the distribution from (<http://imagej.net>) and follow the installation instructions [14].
4. Once installed, the MorphoLibJ plugin is required for segmentation. From the Help menu, select Update. This will bring up the Updater window. Click Manage Update sites.
5. Once the dialog box appears, scroll down to IJPB-plugins. Select the check-box and then close the window.
6. Apply the changes and restart ImageJ.
7. Once restarted, load the images to be segmented and select Classic Watershed in the MorphoLibJ menu. (see Note 1)
8. For more details on segmentation with MorphoLibJ in ImageJ, see (http://imagej.net/Classic_Watershed) [15].
9. In Matlab, the *watershed* function performs watershed transform. For more information on how to use the function, type *help watershed* in the Matlab command window. This function is part of Matlab's Image Processing Toolbox.
10. To convert images to OME-TIFF format, Bio-Formats can be downloaded from the Open Microscopy webpage (<https://www.openmicroscopy.org/>). We suggest using python-bioformats, a Python wrapper for the toolkit (<https://pypi.python.org/pypi/python-bioformats>), but you can build OME-TIFFs using C++ bindings as well as the toolboxes for Matlab and Octave. Please refer to the Bio-Formats documentation for details on how to convert images.
11. If you are going to use the Python wrapper to convert images into a single OME-TIFF, then you will need to write a script that builds a container for storing both pixels and metadata. Example scripts can be found at (<http://cellorganizer.org>).

Geometries for Spatial Cell Simulations

12. Most fields are optional in the data model, however, the images must contain information about the size of the sample region corresponding to one pixel, i.e. you must populate the fields `PhysicalSizeX`, `PhysicalSizeY` and `PhysicalSizeZ` as well their respective units. Often, this information is automatically populated by microscopes in their respective proprietary formats. When converting from one of these proprietary formats to an OME-TIFF, Bio-Formats should populate these fields. If not, then you should do it manually. Please refer to the CellOrganizer website for examples.

3.2. Uploading images

1. From the CellOrganizer for Galaxy homepage, select *Get Data* from the top-left side menu.
2. Then select *Upload File from your computer* to open the tool.
3. From the tabs at the top of the tool window, click *Collection*.
4. From the dropdown menu *Collection Type*, select *List*.
5. From the dropdown menu *File Type*, select *mat*.
6. Click on *Choose local files* and navigate to your local directory containing the image files. Select the OME-TIFF files created in the previous steps.
7. Click **Start**. This will begin uploading the images.
8. After the images finish uploading, click **Build** and name your dataset. This will create an image dataset in your history (see Note 2),

3.3. Building a workflow for training a diffeomorphic model

1. From the CellOrganizer for Galaxy homepage, select *Workflow* at the top of the page. This will bring up your list of saved workflows, if any exist.
2. Near the right side of the screen, create a new workflow by clicking the **Create new workflow** button. On the next screen, provide the name 'Diffeomorphic framework training' for the new workflow (See Note 3).
3. Once on the workflow canvas, the main Tools menu can be seen on the left side of the page (see Figure 1). This contains all of the CellOrganizer widgets for data import, model training, synthesis, and visualization. From the *Inputs* section of the tools menu, click *Input dataset collection*. A widget will appear in the workflow canvas; this will be the starting block for any and all workflows in CellOrganizer.
4. From the *Training* section of the Tools menu, select the *Trains a generative model* widget to add it to your canvas.
5. To connect the input data to the model training widget, click and drag the '>' icon on the right side of the input dataset block to the '>' on the left side of the training block. The path will turn green if the two data types are compatible.
6. To save the workflow, click the small gear above the canvas and select **Save**. This will allow you to edit and run the workflow in the future. (See Figure 1)
7. From the gear menu, select **Run**. This will bring up the job submission page containing the various options for each widget in the workflow (See Note 4).
8. Select the input dataset(s) for model training by using the drop-down menu.

Geometries for Spatial Cell Simulations

9. To set training options, click the *Train generative model* box highlighted in tan.
10. First, provide a name for the model matching your image dataset.
11. Change the cellular components option to 'Nuclear and cell shape (framework)'.
12. Select the model dimensionality.
13. Provide the integer index for the DNA and cell image channels.
14. For nuclear and cell model class, choose 'framework' to build a model where cell and nuclear shape are interdependent.
15. Next, select the nuclear and cell model types. Choose 'diffeomorphic' for both.
16. Models can be trained at various resolutions by setting different downsampling rates. Higher downsampling rate yields a lower resolution model yet faster training speed. Provide a desired downsampling rate for training. You can set this in the '*General options*' box.
17. You can use the '*Advanced options*' box to add optional parameters. Default parameters values are set when training of this model class and type. Provide some documentation for the model in the box '*Documentation*'. This information will be saved with the model file as a model descriptor.
18. Finally, click **Run** workflow. On the right side of the screen, your history will be populated with multiple jobs, each corresponding to a widget in the workflow.
19. To check on a job's status or review standard input and output, select the job in your history and click the 'i' in the lower left. This will bring up a detailed summary page with relevant information for the job (see Figure 2).
20. When training is complete, the training job in your history will turn green and the output can be saved by clicking the disk icon.
21. To visualize the trained model, select the *Display shape space plot from a diffeomorphic model* tool from the *Useful tools for models* section of the Tools menu.
22. Choose the trained diffeomorphic model as your input dataset. Use the default options and select **Execute**.
23. Once finished, click the eye icon on the job in your history. This will open a page for viewing the trained shape space (see Note 5).

3.4. Building a workflow for training a vesicle model

1. Create a new workflow for vesicular model training, adding the same inputs and training widgets as in 'Diffeomorphic framework training'.
2. To better grasp the parameters of the trained model, add the *Print information about a generative model file* widget from the *Useful tools for models* subsection and connect the *Trains a generative model* widget. This will generate a summary of the trained vesicular model with figures for each fitted parameter.
3. Save the workflow and select run from the gear drop-down menu.
4. On the next page, change the input dataset to your image dataset containing a vesicular protein channel.
5. Click the section for *Train generative model* highlighted in tan.
6. Change the channel indices to match the image dataset format.

Geometries for Spatial Cell Simulations

7. Under '*Select the nuclear components desired for modeling*', select 'Nuclear shape, cell shape and protein pattern' to train the model for cell and nuclear shape and vesicular protein pattern.
8. Select the model dimensionality.
9. Provide a name for the model.
10. Select options for nuclear and cell name, type, and class.
11. Protein model class should be set to 'vesicle'.
12. Under '*Select a protein model type*', ensure that 'Gaussian mixture model' is selected. Choose an option for the protein location.
13. If desired, document the model and run the workflow.
14. Once finished, save the trained vesicle model by clicking the disk icon on the model training job in your history (See Note 6).
15. To view the model summary, click the eye icon on the *Print information summary* job in your history. This will bring up a set of figures showing the fitted parameters of the newly trained model.

3.5. Synthesizing an instance from a trained model

1. Navigate to the workflow page and create a new workflow.
2. Add an *Input dataset collection* widget. This will be a model trained in CellOrganizer that will be used for sampling.
3. Under the *Synthesis* subcategory in the Tools menu, add the *Generates a synthetic image from a valid model* widget.
4. Connect the input dataset collection to model1 in the *Generate synthetic image* widget (see Note 7).
5. If an image output is desired, various tools for visualization are available under the *Useful tools for images* subcategory in the Tools menu. These tools modify the OME-TIFF output and convert to PNG format for web viewing (see Note 8).
6. Add the *Generates a surface plot from a 3D OME.TIFF images*, *Makes an RGB projection from an OME.TIFF*, and *Makes a projection from an OME.TIFF* widgets. Connect the output to each new widget.
7. Save the workflow and select run from the gear menu.
8. On the job submission page, select the model(s) that will be sampled from as the input dataset(s).
9. Next, select the structures to be synthesized from the drop-down menu.
10. If using a diffeomorphic model, multiple sampling methods are possible. Using the '*Advanced options*' box, you can add a random walk method and a number steps to sample from the trained shape space.
11. Select the output format desired; each instance can be output in four distinct formats: OME-TIFF, indexed tiff, Wavefront OBJ, and SBML Level 3 Spatial files. For this workflow, select OME-TIFF.
12. In each visualization tool, use the default options and run the workflow.
13. Once finished, instances can be downloaded and saved by clicking the disk icon on the job in the history.
14. The visualization outputs can be viewed by click the eye icon on their respective jobs in the history.

3.6. Using synthetic geometries and trained models for Biochemical Simulation

3.6.1. VCell

1. Once a synthetic instance is generated, it can then be imported into various programs that support the creation of well-defined compartmental geometries. For 2D simulations, an indexed image can be exported for use in VCell [1].
2. Under the *Useful tools for images* category in the Tools menu, select *Export to VCell*. This function takes a 2D or 3D OME-TIFF and converts it to an indexed tiff. If the image is 3D, it is converted to 2D by doing a projection before creating the indexed image.
3. In the tool options box, select a previously generated synthetic image and click **Execute**.
4. Once the conversion is finished, the output image can be saved to a local drive by clicking the disk icon under the job in the history.

3.6.2. CellBlender

1. 3D biochemical simulations can also be performed using platforms like CellBlender in conjunction with CellOrganizer and BioNetGen [3, 16, 17]. CellBlender supports both Wavefront OBJ and SBML Level 3 Spatial files, both of which can be generated by CellOrganizer.
2. During synthesis, add the output options for Wavefront OBJ files or SBML-Spatial files and set to *true* to generate geometries from a trained model. The default value on these options is *false*.
3. These geometries can then be saved to a local drive by clicking the disk icon under the synthesis job in the history.

3.6.3. SBML and SBML Spatial

1. The Systems Biology Markup Language (SBML) is a widely used format for biochemical modeling. Recently, the language has been extended to support spatial information through the SBML-Spatial specification [18]. To integrate biochemical models with spatial information, CellOrganizer can output a synthetic instance following the SBML specification. These can then be used with simulation systems that support SBML.
2. Using the *Generates a synthetic image from a valid model* tool, add the output option for SBML-Spatial and set to *true* to generate a geometry. The default value on this option is *false*.
3. The SBML-spatial instance can then be imported into other programs supporting the format.

Notes:

1. Seeded or Marker-controlled watershed segmentation can often yield more accurate results. Regions in an image strained for the nucleus are used as the 'seeds' or 'markers' for the watershed transform, superimposed on a cell-stained image channel, that begin to grow as the algorithm progresses. Rather than a

Geometries for Spatial Cell Simulations

threshold determining the centers of the growing region, the segmentation is guided by the nuclear channel. In ImageJ, load both a nuclear image and cell image. From the Plugins menu, select MorphoLibJ, then Segmentation, and Marker-controlled Watershed. From the dialogue box, select the cell image as Input and the nuclear image as Marker. Click OK to perform segmentation.

2. If image uploading is successful, the upload job in your history will turn green, signifying upload completion.
3. Both a name and an annotation can be added to each workflow. If workflows are specialized for a single cell line or type, add an annotation to better distinguish similar workflows.
4. Options can also be set for each workflow from within the workflow canvas. This allows the user to set default options for each workflow that may be modified when choosing to run.
5. Much like other online tools, Galaxy allows the user to be notified by email when each job is finished running. Select 'Yes' under Email notification on the right-hand side of the screen on the workflow page to receive these emails.
6. If multiple vesicle models have been trained, comparison between trained model parameters can be visualized using the Compare models tool. Select the tool from the Useful tools for models category and input the two trained models. Once completed, this tool returns figures comparing each trained model parameter.
7. Multiple protein models can be used, however, cell and nuclear shape models can only be input in model1. All other cell and nuclear shape models will be ignored.
8. All tools include an example images (if applicable). To view, navigate to the CellOrganizer for Galaxy homepage and click on a tool. Scroll to the resulting page.

Acknowledgments

The original research upon which these protocols are based was supported in part by National Institutes of Health grant R01 GM090033 and P41 GM103712.

References

1. Resasco, D.C., et al., *Virtual Cell: computational tools for modeling in cell biology*. Wiley Interdisciplinary Reviews. Systems Biology and Medicine, 2012. **4**(2): p. 129-140.
2. Robinson, M., S.S. Andrews, and R. Erban, *Multiscale reaction-diffusion simulations with Smoldyn*. Bioinformatics, 2015. **31**(14): p. 2406-8.
3. Kerr, R.A., et al., *Fast monte carlo simulation methods for biological reaction-diffusion systems in solution and on surfaces*. SIAM J Sci Comput, 2008. **30**(6): p. 3126.
4. Mochly-Rosen, D., *Localization of protein kinases by anchoring proteins: a theme in signal transduction*. Science, 1995. **268**(5208): p. 247-51.

Geometries for Spatial Cell Simulations

5. Huh, W.-K., et al., *Global analysis of protein localization in budding yeast*. Nature, 2003. **425**(6959): p. 686-691.
6. Hung, M.C. and W. Link, *Protein localization in disease and therapy*. J Cell Sci, 2011. **124**(Pt 20): p. 3381-92.
7. Zhao, T. and R.F. Murphy, *Automated learning of generative models for subcellular location: building blocks for systems biology*. Cytometry A, 2007. **71**(12): p. 978-90.
8. Johnson, G.R., et al., *Joint modeling of cell and nuclear shape variation*. Mol Biol Cell, 2015. **26**(22): p. 4046-56.
9. Peng, T. and R.F. Murphy, *Image-derived, three-dimensional generative models of cellular organization*. Cytometry A, 2011. **79**(5): p. 383-91.
10. Li, J., et al., *Estimating Microtubule Distributions from 2D Immunofluorescence Microscopy Images Reveals Differences among Human Cultured Cell Lines*. PLoS ONE, 2012. **7**(11): p. e50292.
11. Shariff, A. and R.F. Murphy, *Automated estimation of microtubule model parameters from 3-D live cell microscopy images*. IEEE, 2011. **11**: p. 1330-1333.
12. Shariff, A., R.F. Murphy, and G.K. Rohde, *A generative model of microtubule distributions, and indirect estimation of its parameters from fluorescence microscopy images*. Cytometry. Part A : the journal of the International Society for Analytical Cytology, 2010. **77**(5): p. 457-466.
13. Afgan, E., et al., *The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update*. Nucleic Acids Research, 2016.
14. Schneider, C.A., W.S. Rasband, and K.W. Eliceiri, *NIH Image to ImageJ: 25 years of image analysis*. Nature methods, 2012. **9**(7): p. 671-675.
15. Legland, D., I. Arganda-Carreras, and P. Andrey, *MorphoLibJ: integrated library and plugins for mathematical morphology with ImageJ*. Bioinformatics, 2016. **413**.
16. Faeder, J.R., M.L. Blinov, and W.S. Hlavacek, *Rule-Based Modeling of Biochemical Systems with BioNetGen*, in *Systems Biology*, V.I. Maly, Editor. 2009, Humana Press: Totowa, NJ. p. 113-167.
17. Smith, A.M., et al., *RuleBlender: integrated modeling, simulation and visualization for rule-based intracellular biochemistry*. BMC Bioinformatics, 2012. **13**(8).
18. Waltemath, D., et al., *Toward community standards and software for whole-cell modeling*. IEEE Transactions on Biomedical Engineering, 2016. **PP**(99): p. 1-1.

Geometries for Spatial Cell Simulations

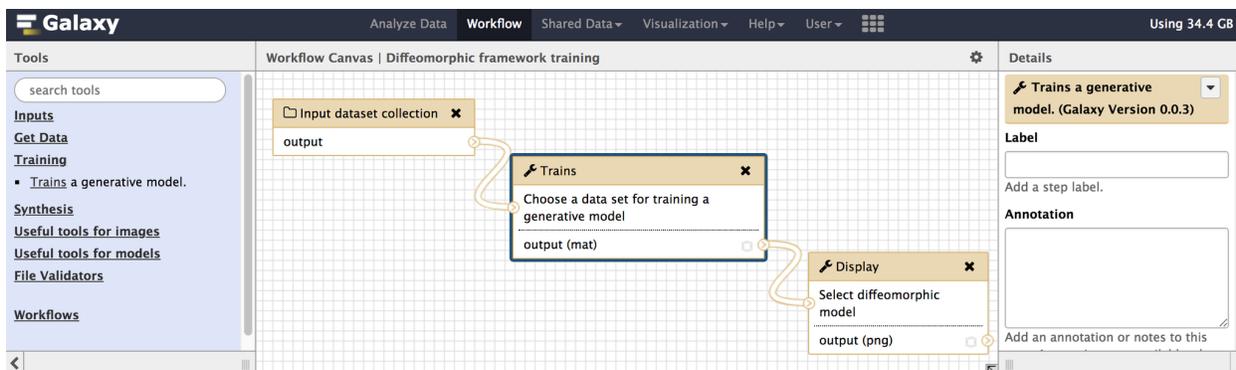


Figure 1. The workflow canvas in CellOrganizer for Galaxy showing the diffeomorphic model training pipeline. The menu on the left contains all tools available divided into various categories. Currently shown are visualization tools, allowing the user to view images in various forms within the browser. The *Display shape space plot from a diffeomorphic model* tool is included at the end of the workflow; this will generate a projection of the learned shape space in two dimensions. See Figure 8 for a learned shape space visualization.

Input Parameter	Value	Note for rerun
Choose a data set for training a generative model	51: 2D HeLa LAMP2	
Provide a name for the model	Empty.	
Selected the cellular components desired for modeling	framework	
Dimensionality	2D	
nuc		
Provide the DNA channel number for the image dataset	1	
Provide an (optional) name for the nuclear model	default	
Choose a class for this model	framework	
Choose a model type	Diffeomorphic	
mem		
Provide the cell channel number for the image dataset	?	

Figure 2. The result of running a workflow on the job history. Each tool in the workflow generates a single job in the history. Jobs in grey are waiting to be run, yellow are currently running, and green have been completed. More detailed information can be viewed by selecting the job in the history and clicking the 'i' in the lower left corner (highlighted in red). Outputs can be viewed or saved by clicking the eye icon in the upper right corner (highlighted in blue).

Geometries for Spatial Cell Simulations

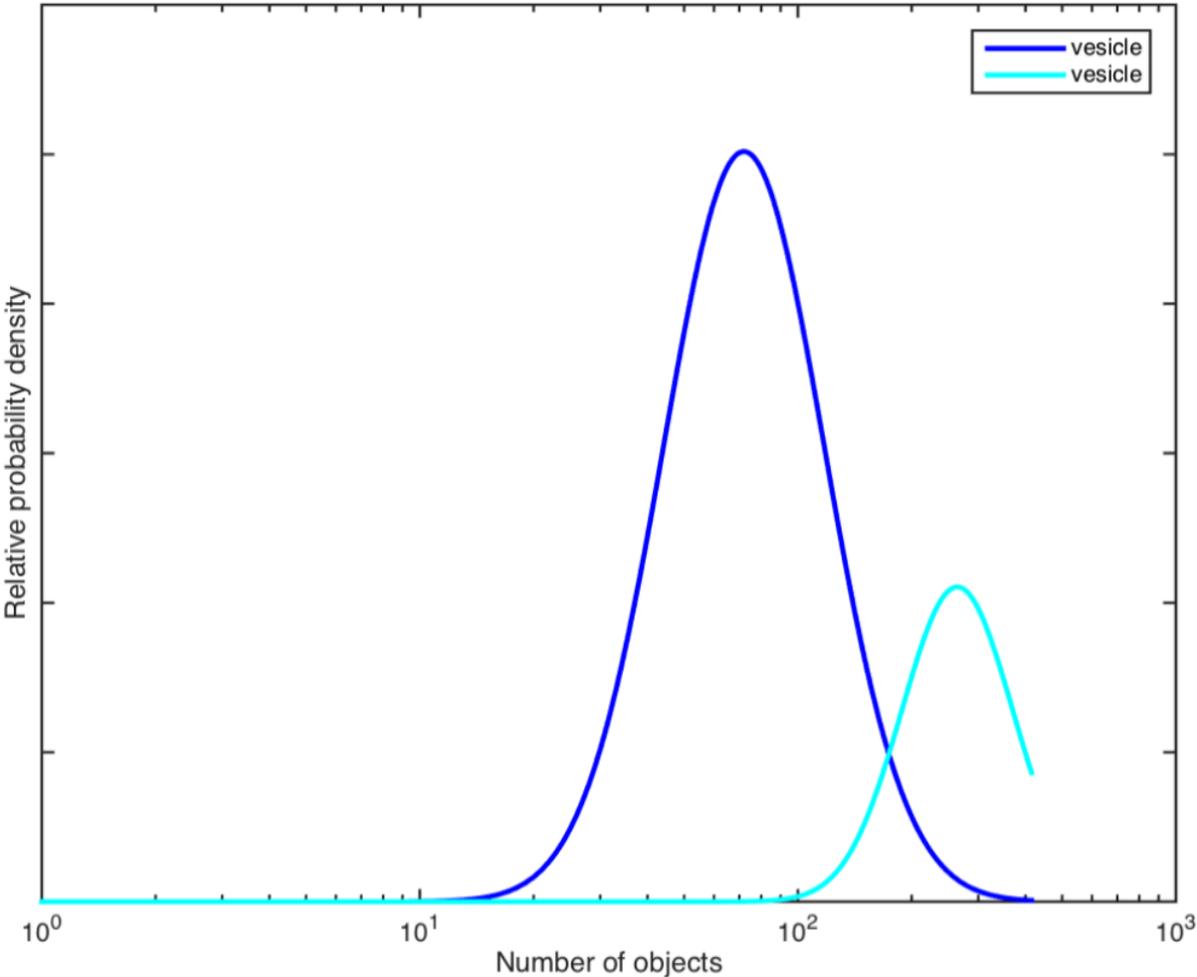


Figure 3. Number of objects. Comparison of the distributions of the number of objects for the two trained models. Values are in logarithmic scale. Two vesicles models were trained from the same image collection: mitochondria and lysosomal tags.

Geometries for Spatial Cell Simulations

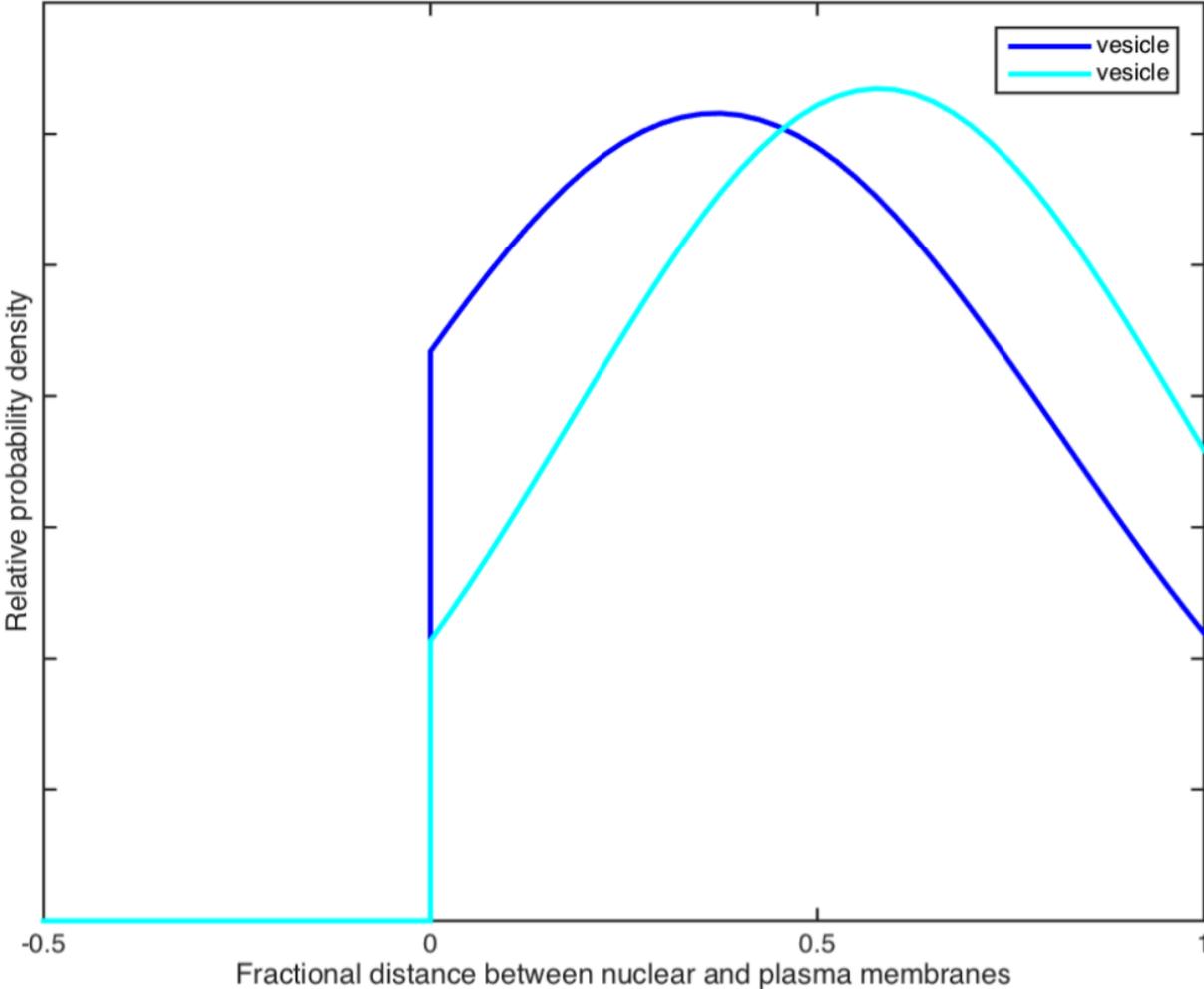


Figure 4. Object spatial distributions. Comparison of the spatial distributions of vesicular objects by the fractional distances between nuclear and plasma membranes.

Geometries for Spatial Cell Simulations

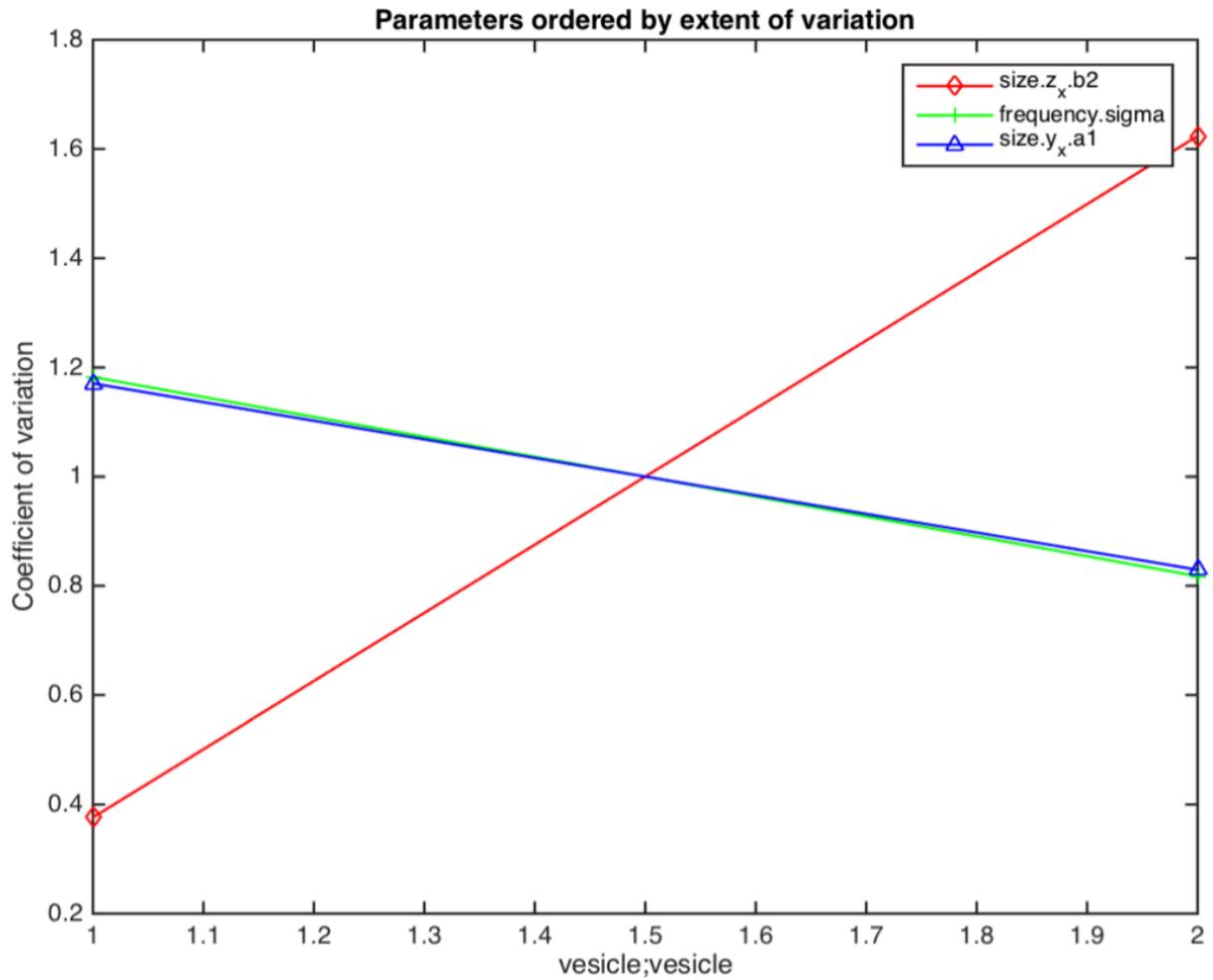


Figure 5. Parameters ordered by the extent of variation. Plot of parameters ordered by the extent of variation. The left axis points are values for the first model and the right for the second model.

Geometries for Spatial Cell Simulations

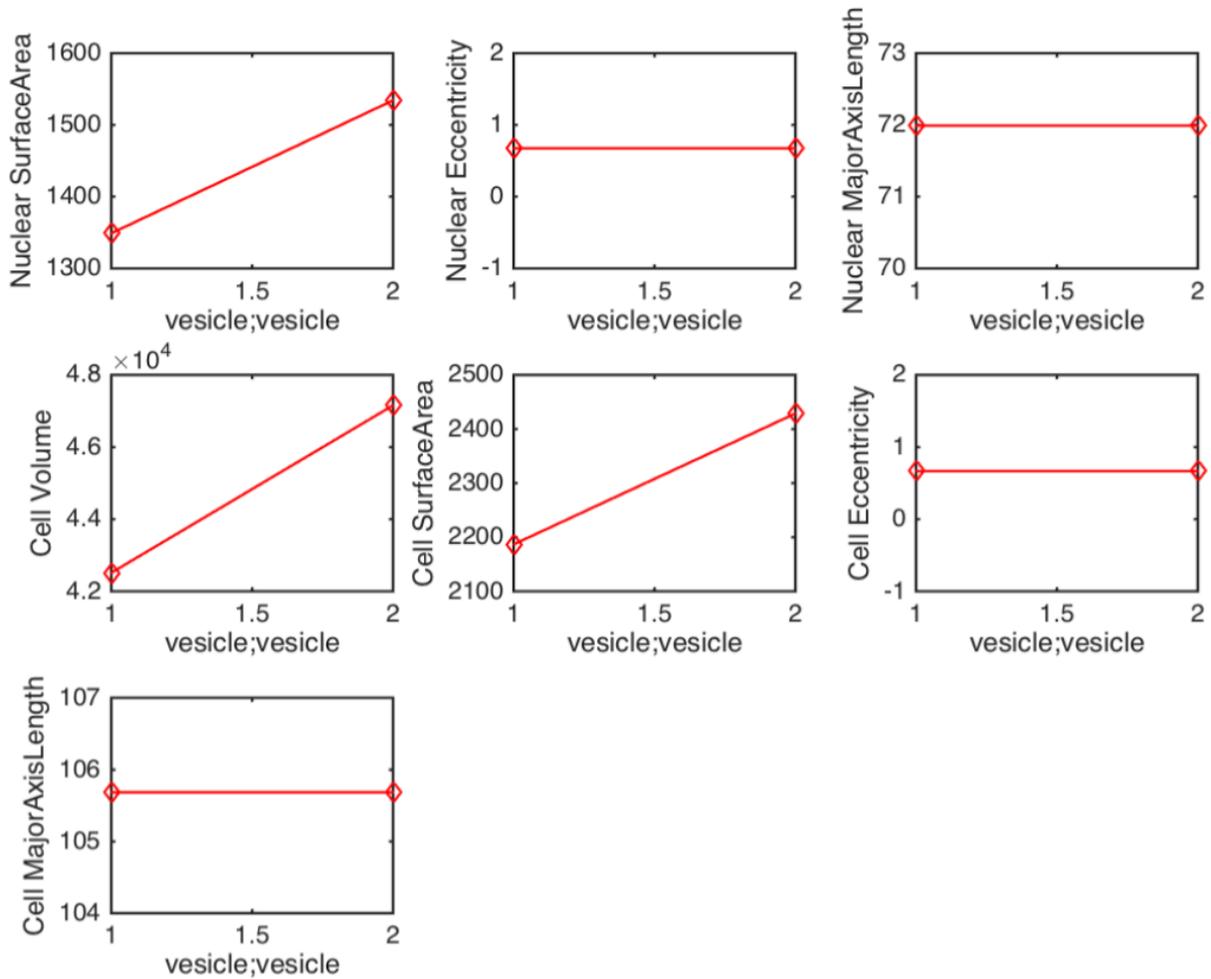


Figure 6. Comparison of different factors. Plots of various properties of the trained models. In each plot, the left axis points are values for the first model and the right for the second model. Here we show surface area, eccentricity, major axis length and volume of cells.

Geometries for Spatial Cell Simulations

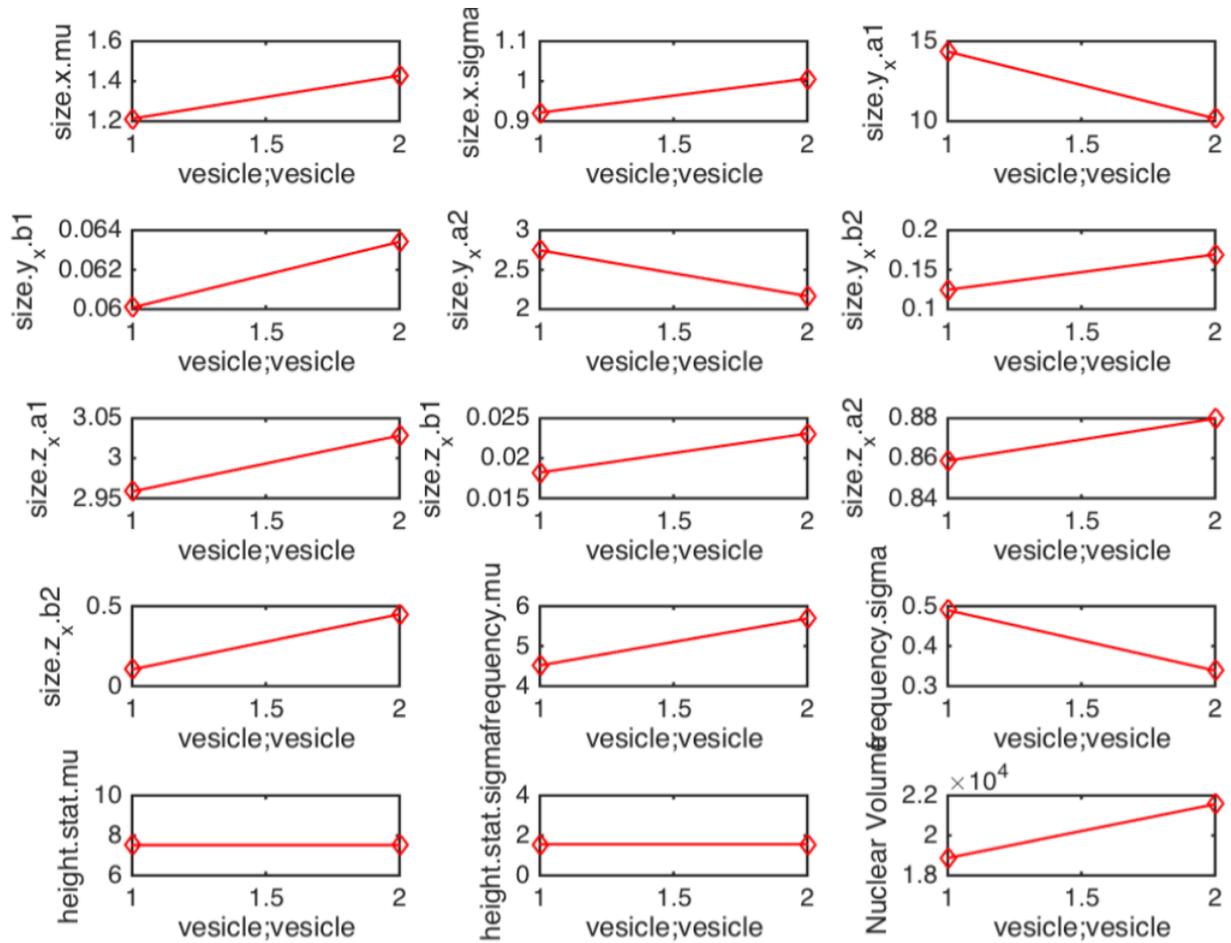


Figure 7. Detailed comparison of parameters. The figure shows the comparison of all main parameters of the models. In each plot, the left axis points are values for the first model and the right for the second model.

Geometries for Spatial Cell Simulations

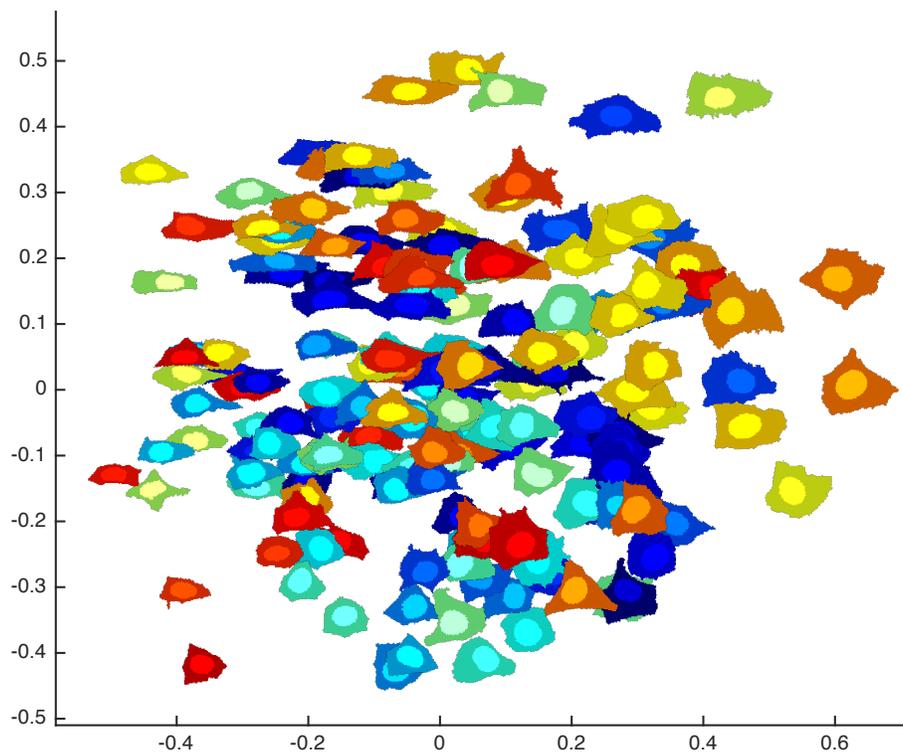


Figure 8. A shape space visualization generated from a diffeomorphic framework model trained on the Murphy lab 3D HeLa image collection (available at <http://murphy-lab.cbd.cmu.edu/data/>). The 7-dimensional shape space is projected into two dimensions. Euclidean distance within the projection paired with color difference can be interpreted as similarity or dissimilarity in shape. Images that are closer together and of similar hues are morphologically similar while further distances and highly varying colors show dissimilarity. Morphological trends can be seen from left (small, short) to right (large, tall) and from top (rectangular) to bottom (triangular).