1

SimuCell Documentation

SimCell Overview

Introduction

SimuCell is an open-source framework for specifying and rendering realistic microscopy images containing diverse cell phenotypes, heterogeneous populations, and microenvironmental effects.

SimuCell can generate heterogeneous cellular populations composed of diverse cell types. Each cell type (i.e. subpopulation) can be defined independently by specifying models for cell and organelle shape, and distributions of markers over these shapes. Models are typically algorithmic, but there is support for rendering produced by other tools, such as the highly realistic models learned from image data by CellOrganizer (via the new SLML markup language).

SimuCell allows users to specify interdependencies among biomarker-, cell-, and population-level phenotypes. For example, a marker's cellular distribution can be affected by the cell's microenvironment or the localization pattern of another marker. These definable image properties are accessible to users either via a novel scripting syntax built on top of MATLAB, or through the graphical user interface, while intermediate results can define further "ground truths" (e.g. cell boundaries can be used to validate segmentation algorithms).

SimuCell was designed to be easily extensible, providing a standard framework for defining new plugins. Users interested in adding novel phenotypes to SimuCell's palette can typically do so by writing just a few lines of code, in part due to MATLAB's extensive library of functions.

Taken together, SimuCell allows the definition of a broad range of phenotypes, encompassing highly non-trivial population-level effects such as cell-type heterogeneity or local cell-density effects.

Overall

To produce an image, SimuCell requires you to define several elements:

- Subpopulation(s)
- Object/Shape(s) (via shape plugins)
- Marker(s) (via marker plugins)
- Other parameters like placement, artifact, overlapping and compositing (other plugins)

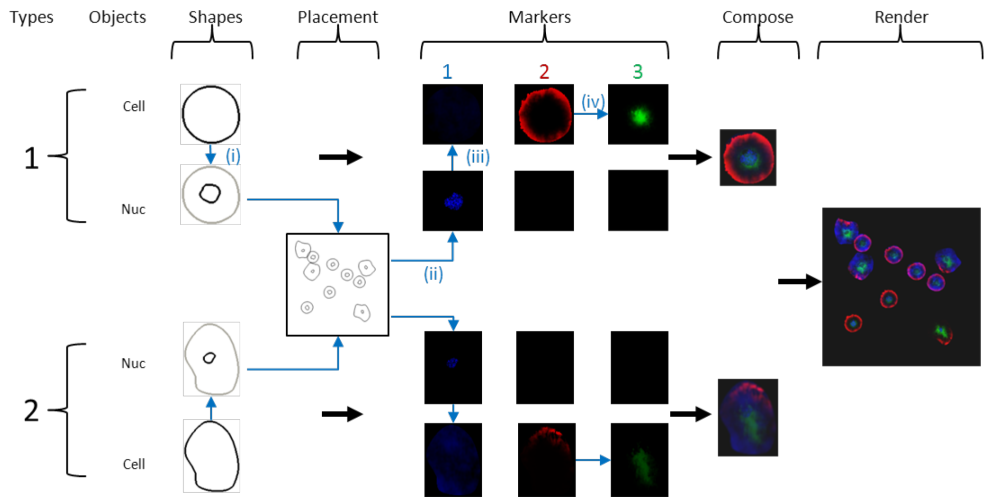


Figure 1 - SimuCell layout process (Blue arrows and images indicate interdependencies for specific example: (i) nuclear shape depends on cell shape; (ii-iii) cell microenvironment (number of nearby cells) affects marker 1 distribution; and (iv) marker 3 distribution depends on marker 2's distribution. Subpopulations

SimuCell allows you to define multiple cell types (each termed a ***subpopulation)*** and specify the fractions of cells belonging to the different subpopulation per image. As shown in Figure 1, for each subpopulation, you will have to define objects/shape, markers and so on.

Objects/Shapes

Objects/shapes are basically the cell components you want to draw in your images. Examples of ***objects*** include cytoplasm, nucleus, lipid droplet, nuclear body. Each object in a cell is described, and subsequently rendered, using its own model (i.e. the appropriate plugin). You are required to choose the appropriate plugin for your object, and set the correct model parameters (default values are a good starting point).

Each object is rendered independently, so there needs to be some way to connect the different objects in a cell to place them close to each other. This connection is done by choosing appropriate models (models allow one object to depend on another). For example, to ensure that the nucleus is inside the cytoplasm you could pick a nucleus model that places it inside an already defined cytoplasm. Alternately, you could pick a nucleus model that draws nuclei independently, but then you need to define a cytoplasmic model that draws the cytoplasm around the nucleus.

In this basic tutorial, you will see that the ***model Elliptical cytoplasm*** is used to define the cytoplasm and by pointing the model parameter (***Centered Around***) to the nucleus you can connect the two objects nucleus and cytoplasm.

Thus, there are two kinds of shape models: ones that don't depend on other objects/shapes (these anchor the cell) and models that depends on other shapes (these will effectively draw the other shapes around the anchor). For any subpopulation, you will have one (and only one) model of the first kind, while all the other models must be depend on each other in some way.

Markers

Next, you need to define the markers. First you add/declare the markers that you want rendered in the image. Then SimuCell requires independent definition of markers (in terms of how they will be rendered) for each defined object of each subpopulation giving a relationship matrix between objects/shapes and markers like the following one:

	Marker 1	**Marker 2**	**Marker 3**
Nucleus	☑	-	-
Cytoplasm	-	☑	☑

As you can see, all the markers have to be defined by object, although you don't need to define all the pair object/marker (in our example, Marker 1 is present only in the nucleus and not in the cytoplasm rather than Marker 2 and 3 are only present in the cytoplasm).

In contrast to shape each marker-object pair is defined using a sequence of elementary operations. There are basically two kinds of operation:

- Operations that set the basic intensity/level of the marker on the object: An operation of this kind should be the first in your sequence of operations. Examples include ***Constant marker level operation*** and ***Cell Density Dependent Marker Level*** which both set the intensity of all pixels in the shape to a constant level. However, in the case of the former this level is sampled from a normal distribution while the latter chooses a level based on the local cell density.
- Operations that redistribute the initial marker setting: These form the subsequent operations. Examples are ***Linear marker gradient,*** which scales the output of the previous operation by an intensity gradient or ***Perlin Texture*** which adds noise to the result of the previous operation,

Here is a quick summary of the marker plugin operations currently available by default in SimuCell.

Angular marker	Produce an angular dependence (gradient) of the marker level

gradient	
Cell density dependent marker	Operation that sets the marker level in an object, based on the local cell density.
Constant dependent marker level	Marker is set to a constant level on an object, with this level being determined by distribution of another Marker/Object pair (e.g. cytoplasm intensity of Marker 1 is decided by the nuclear density of Marker 2)
Constant marker level	Constant marker level with no dependencies. Also, it will change from cell to cell.
Distance to edge marker level	Marker level at a pixel will depend on its distance to the edge of the object
Distance to shape marker gradient	Marker level at a pixel will depend on its distance to an object in the same cell
Linear marker gradient	Marker level will decrease/increase following a linear gradient
Locally dependent marker level	Level of the marker at a pixel depends of the level of another marker at the same pixel
Micro environmental marker level	Will make the cells close to each other having a similar level of marker intensity
Perlin texture	Adds noise like texture
Turbulent texture	Redistribution of the marker level among neighboring pixels to produce a turbulence effect. Works only on non-uniform marker level intensity
Rescale Marker level	Stretch out the marker level (for example to increase contrast)

The operations marked blue are the ones that set the initial distribution while the others redistribute and refine this initial marker level distribution.

Subpopulation, object and marker definitions are needed for SimuCell to produce images. However, there are a number of other optional effects that can be added to make more realistic images:

Placement

Placement is used to determine how your cells will be distributed in an image (randomly, clustered and so on). Currently, SimuCell provides:

- Clustered (clusters of cells)
- Nearby (cells placed near existing cells)
- Random (cells placed randomly)

Overlap

Overlap lets you define to what extent different cells can overlap, and how this should be measured (on the nucleus, cytoplasm etc).

Composite

Compositing allows you to change the way a marker is rendered when there is an overlap between two objects, both containing the marker. Currently, only one plugin exists (Default compositing)

Image Artifacts

These allow you simulate imaging effects from microscopy image acquisition at a whole image level. SimuCell currently provides:

- Add Basal Brightness
- Linear Image Gradient
- Radial Image Gradient

Cell Artifacts

Cell artifacts allow you to add effects (typically imaging/staining) on a cell by cell basis. SimuCell currently provides:

- Cell Staining
- Out of Focus

Basic SimuCell Tutorial

Requirement:

SimuCell requires MATLAB version ≥ 2011a to work.

This tutorial is a step by step guide to build a synthetic image containing a single subpopulation of cells. The cells contain only a nucleus and cytoplasm. They are "stained" with a nuclear marker, a membrane marker and a cytoplasmic marker whose intensity depends on the local cell density.

Step 0: Start SimuCell Interface

Open a MATLAB terminal and go to the SimuCell project folder and enter the following commands:

```
>cd simucell/src
```

```
>simucell
```

```
>simucellGUI
```

Step 1: Define the Subpopulation(s)

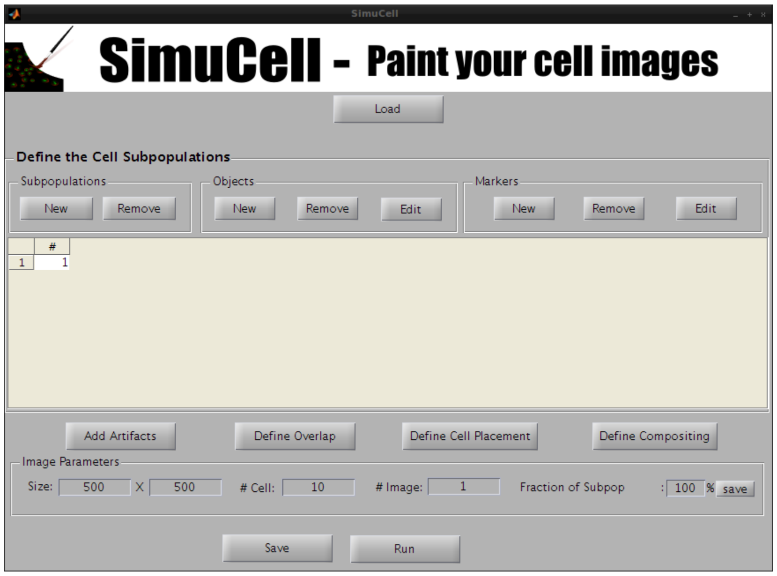


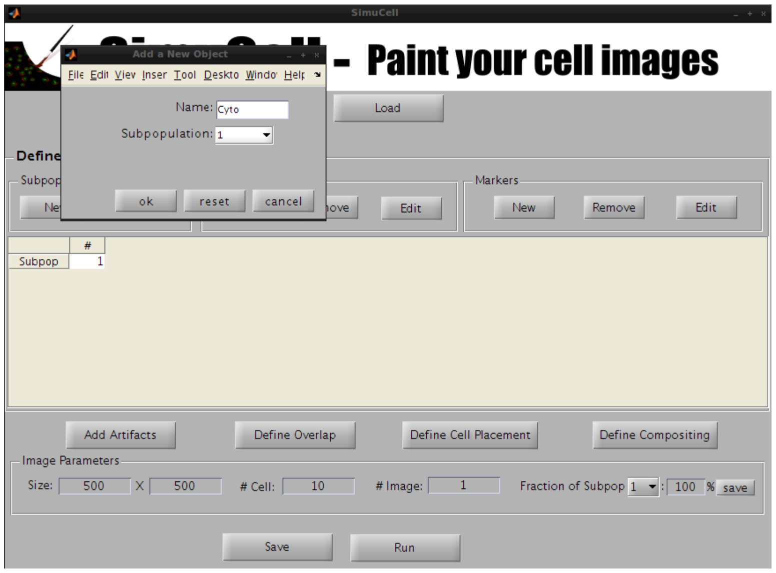
Figure 2- SimuCell Startup Screen: 1 empty subpopulation is present

The SimuCell interface starts with a subpopulation pre-declared (Subpopulation 1). If you want to work with more than one subpopulation, you add one by clicking on ***New*** in the ***Subpopulation*** subpanel.

In this example, we will use only one subpopulation.

Step 2: Declare the Objects (nucleus and cytoplasm)

We add objects to our subpopulation by clicking on the ***New*** button in the ***Objects*** subpanel.



Name: Cyto
Subpopulation: 1

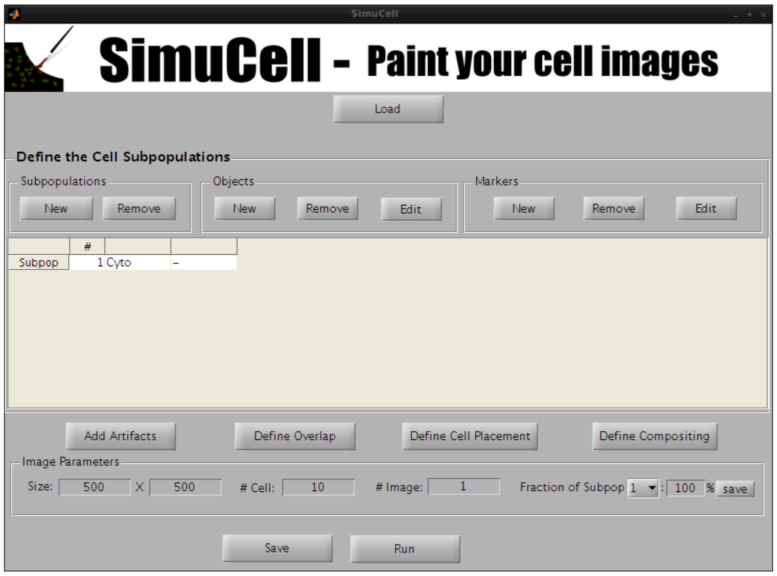
	#
Subpop	1

Image Parameters
Size: 500 X 500   # Cell: 10   # Image: 1   Fraction of Subpop 1 : 100 %

Figure 3 - Create a new Object/Shape named Cyto

A new popup window will appear. Enter the name of the object (in our example, we will use *Cyto*) and press OK.

The *Cyto* object is now visible in the main window in the Subpopulation table.



	#		
Subpop	1	Cyto	–

Image Parameters
Size: 500 X 500   # Cell: 10   # Image: 1   Fraction of Subpop 1 : 100 %

Figure 4 - Cyto Object/Shape is now visible in the Main table interface

Let's create another object. Click on the button ***New*** of the subpanel ***Objects*** to add a new object/shape.

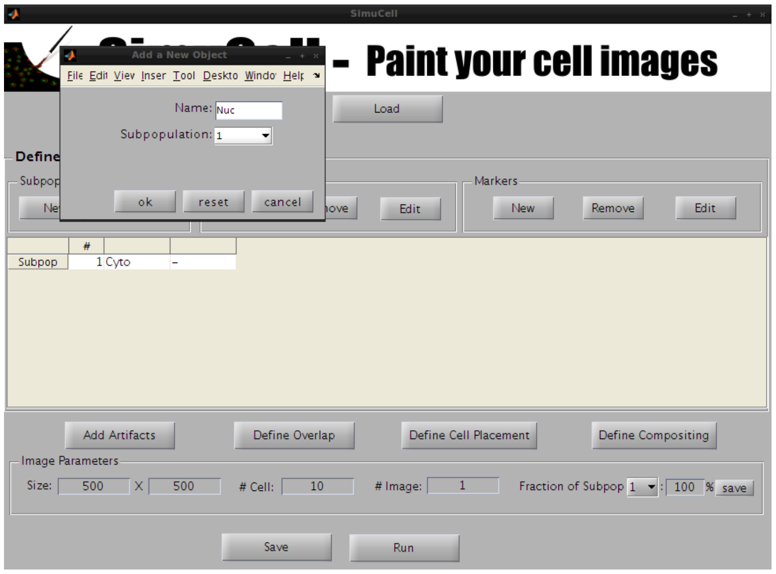


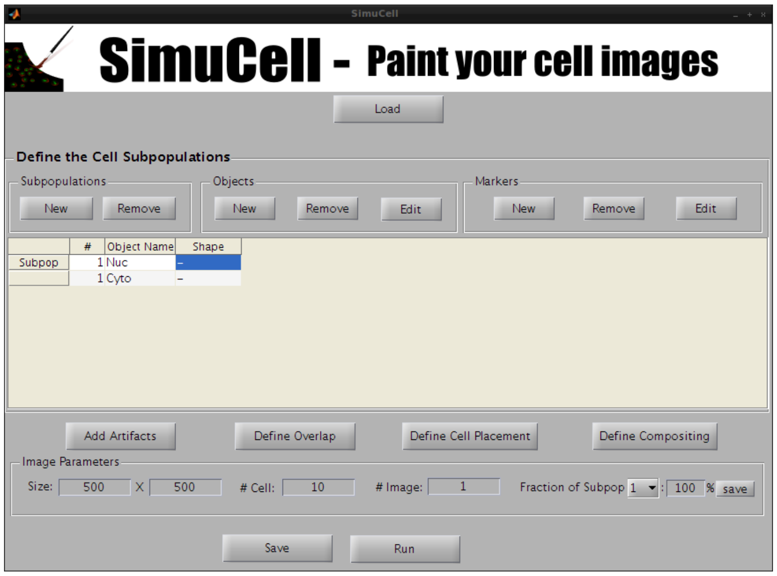
Figure 5 - Create a new Object/Shape named Nuc

A new popup window appears. Enter the name of the object (in our example, we will use ***Nuc***) and press OK.

The ***Nuc*** object is now visible in the main window in the Subpopulation table.

Step 3: Define the Shapes Models for the Objects

Once all the objects types have been created, they can be defined in detail by choosing appropriate models/parameters to render their shapes.



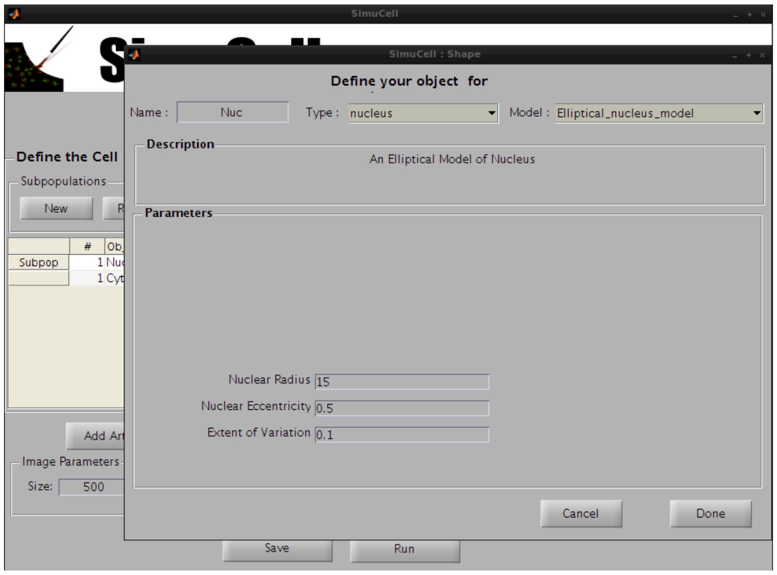
	#	Object Name	Shape
Subpop	1	Nuc	–
	1	Cyto	–

Image Parameters
Size: 500 X 500 # Cell: 10 # Image: 1 Fraction of Subpop 1 : 100 % save

Figure 6 - Select the Object/Shape Nuc to define it in more detail

First, select the object ***Nuc*** and click ***Edit*** in the ***Objects*** subpanel.

A new window will pop up allowing you to define the Object/Shape ***Nuc***.



Name : Nuc Type : nucleus Model : Elliptical_nucleus_model

An Elliptical Model of Nucleus

Nuclear Radius 15
Nuclear Eccentricity 0.5
Extent of Variation 0.1

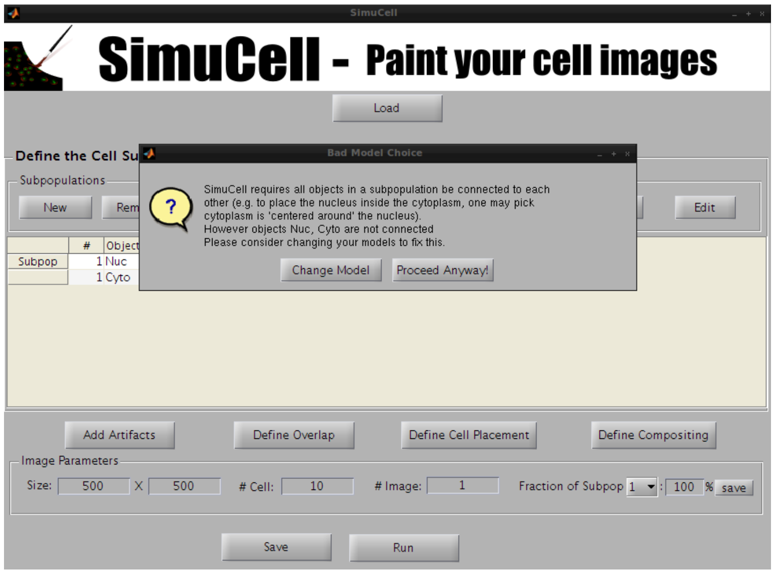
Figure 7 - Select a Type, Model and set the parameters for the Object/Shape Nuc

Choose the type ***nucleus*** and the model ***Elliptical_nucleus_model*** and use the following parameters:

- Nuclear Radius: 15
- Nuclear Eccentricity: 0.5 (a circle has eccentricity 0 , and a straight line 1)
- Extent of Variation: 0.1 (0 is no randomness, and 1 means completely random)

Then press ***Done***.

A warning dialog will popup stating that the two objects/shapes are not connected to each other. We are going to fix this by next by defining the object/shape ***Cyto*** to be dependent of the object/shape ***Nuc***. So for now, press ***Proceed Anyway!*** And let's define the other object/shape.



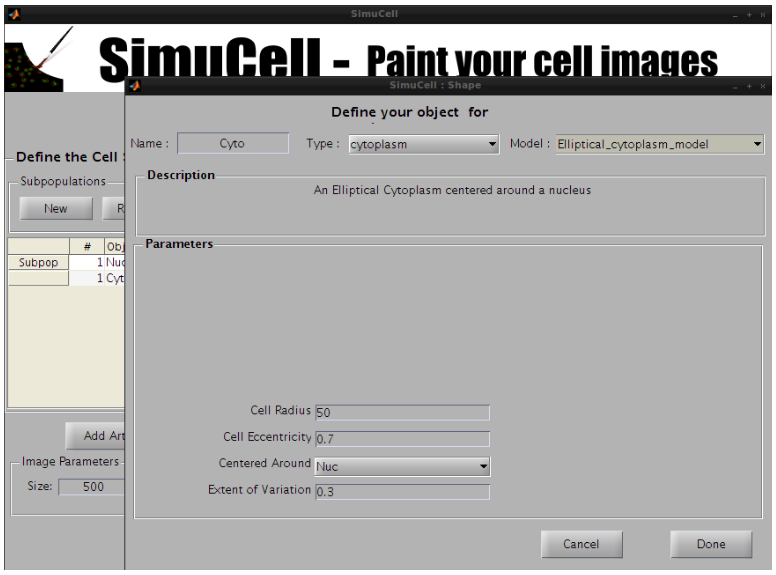
SimuCell requires all objects in a subpopulation be connected to each other (e.g. to place the nucleus inside the cytoplasm, one may pick cytoplasm is 'centered around' the nucleus).
However objects Nuc, Cyto are not connected
Please consider changing your models to fix this.

	#	Object
Subpop	1	Nuc
	1	Cyto

Size: 500 X 500 # Cell: 10 # Image: 1 Fraction of Subpop 1 : 100 %

Figure 8 - When the Objects/Shape are not connected to each other, a warning dialog will pop up to inform you that Object/Shape should be inter-connected to each other (e.g., Cytoplasm should be centered around the Nucleus).

Select the object ***Cyto*** and click ***Edit*** in the ***Objects*** subpanel.



Name : Cyto Type : cytoplasm Model : Elliptical_cytoplasm_model

Description: An Elliptical Cytoplasm centered around a nucleus

Parameters:

Cell Radius 50
Cell Eccentricity 0.7
Centered Around Nuc
Extent of Variation 0.3

Figure 9 - Select a Type, Model and set the parameters for the Object/Shape Cyto

Choose the type ***cytoplasm*** and the model ***Elliptical_cytoplasm_model*** and use the following parameters:

- Cell Radius: 50
- Cell Eccentricity: 0.5
- Centered Around: Nuc (tells SimuCell to draw the cytoplasm around the nucleus)
- Extent of Variation: 0.3

Then press ***Done***.

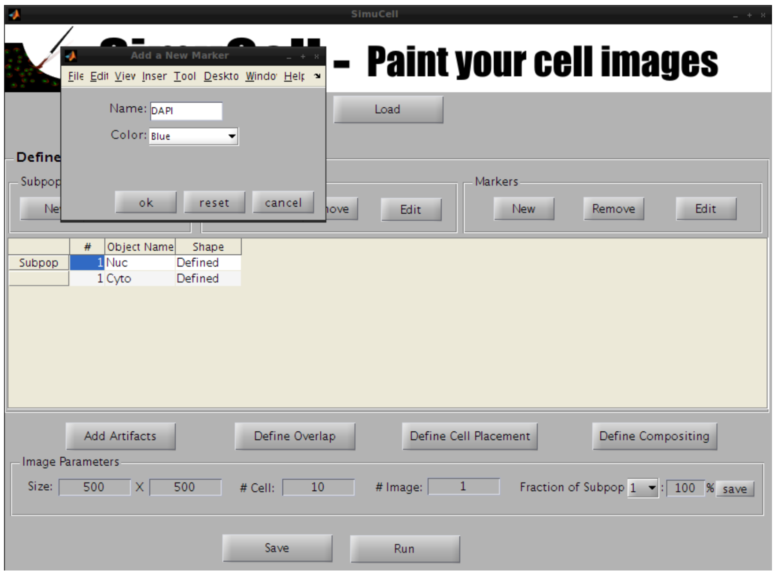
Congratulations, you have just finished defining the objects/shapes used in this tutorial.

Step 4: Create the Markers

Next, you will create the markers used to "stain" the cells in our synthetic images.

Select an entry in the ***Subpopulation table*** and click on the ***New*** button in the ***Markers*** subpanel.

A new pop-up window will appear. Enter the name of the marker (in our example, we will use ***DAPI***), select the marker color (in our example, we will use ***Blue***) and press *OK*.



Name: DAPI
Color: Blue

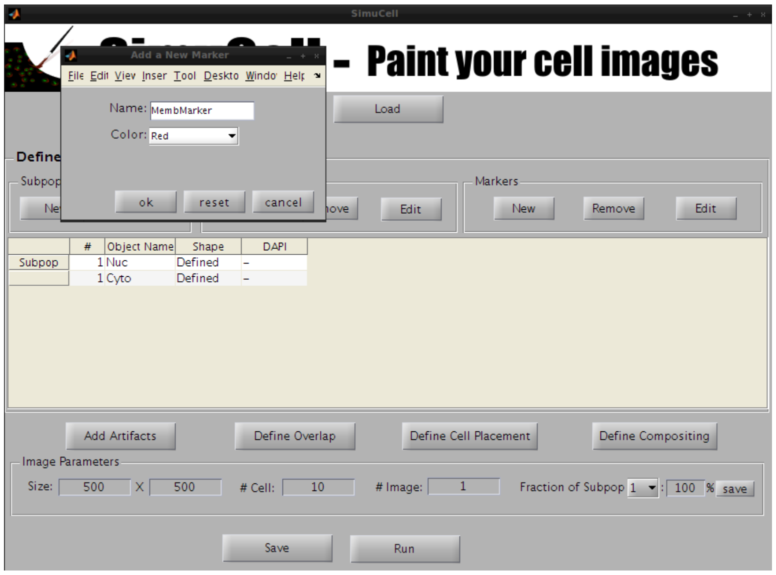
	#	Object Name	Shape
Subpop	1	Nuc	Defined
	1	Cyto	Defined

Image Parameters
Size: 500 X 500 # Cell: 10 # Image: 1 Fraction of Subpop 1 : 100 %

Figure 10 - Create a new marker named DAPI and set it to be Blue in color

The subpopulation table should now display a new column named ***DAPI***.

Repeat the previous step to create the markers ***MembMarker*** in ***Red*** and ***CytoMarker*** in ***Green***.

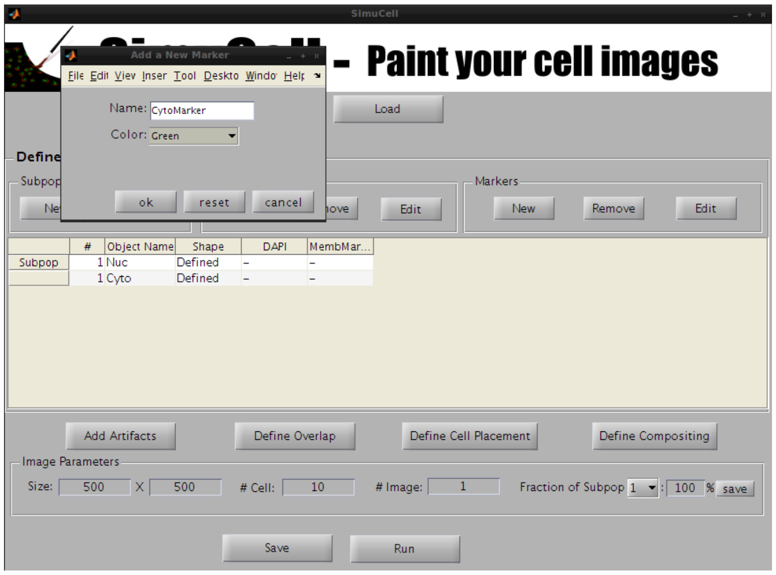


Name: MembMarker
Color: Red

	#	Object Name	Shape	DAPI
Subpop	1	Nuc	Defined	–
	1	Cyto	Defined	–

Image Parameters
Size: 500 X 500 # Cell: 10 # Image: 1 Fraction of Subpop 1 : 100 %

Figure 11 - Create a new marker named MembMarker (Red)



Name: CytoMarker
Color: Green

	#	Object Name	Shape	DAPI	MembMar...
Subpop	1	Nuc	Defined	–	–
	1	Cyto	Defined	–	–

Image Parameters
Size: 500 X 500 # Cell: 10 # Image: 1 Fraction of Subpop 1 : 100 %

Figure 12 - Create a new marker named CytoMarker (Green)

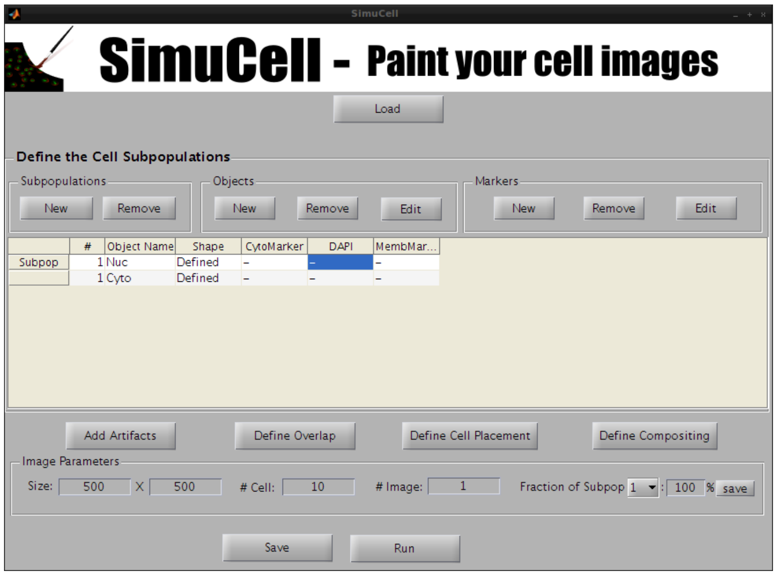
Step 5: Define the Markers

Once the markers have been created, you will have to define them for each object/shape by defining a sequence of operation(s). These operations will determine how SimuCell renders the markers within the selected object/shape.

Start by selecting which object/marker pair you would like to define.

Define the marker DAPI for for the Object/Shape Nuc

In the subpopulation table, select the marker ***DAPI*** for the object/shape ***Nuc***.

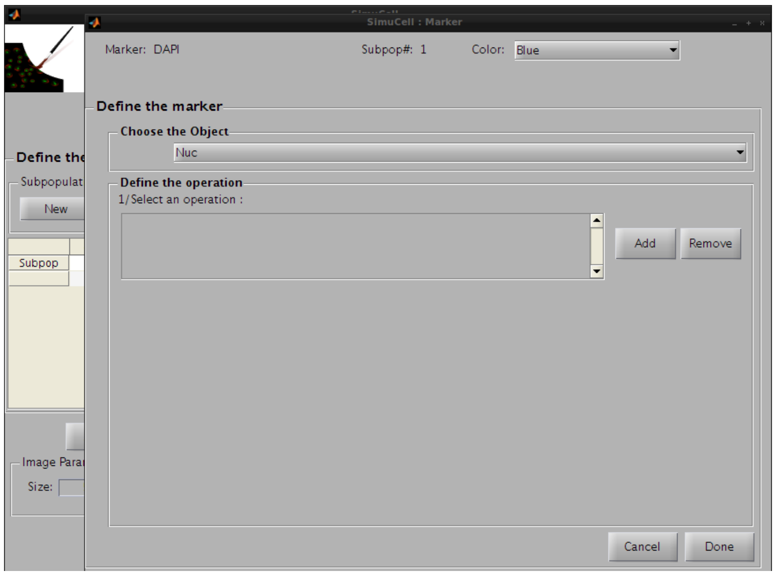


	#	Object Name	Shape	CytoMarker	DAPI	MembMar...
Subpop	1	Nuc	Defined	–	–	–
	1	Cyto	Defined	–	–	–

Size: 500 X 500 # Cell: 10 # Image: 1 Fraction of Subpop 1 : 100 %

Figure 12 - Select the Marker DAPI for the Shape/Object Nuc

Then click ***Edit*** in the ***Markers*** subpanel. A new window to define the marker will appear.



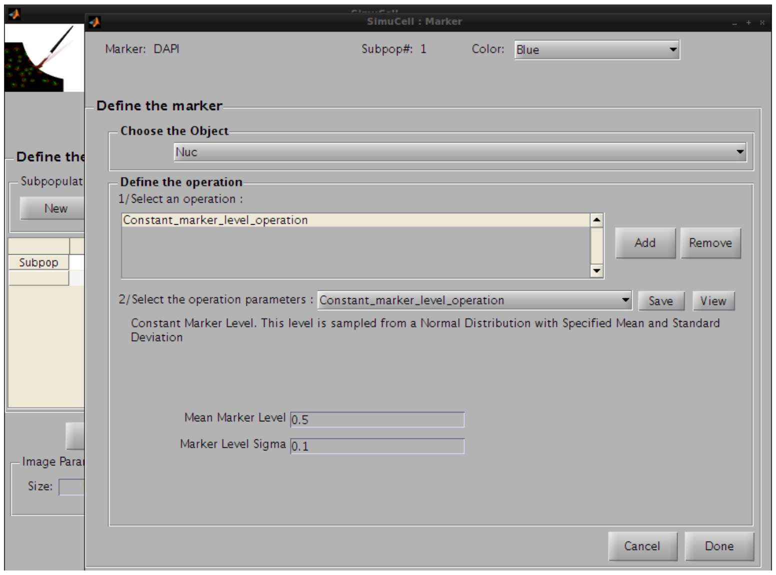
Marker: DAPI Subpop#: 1 Color: Blue

Choose the Object: Nuc

Figure 13 - Define the pair DAPI/Nuc by adding some operations

Click on ***Add*** in the ***Define Operation*** subpanel to add an operation. The default operation, ***Constant_marker_level_operation,*** will appear in the operation list. We will use this default operation with the following parameters:

- Mean Marker Level: 0.5
- Marker Level Sigma: 0.1


SimuCell : Marker

Marker: DAPI Subpop#: 1 Color: Blue

Define the marker

Choose the Object: Nuc

Define the operation

1/Select an operation :

Constant_marker_level_operation

2/Select the operation parameters : Constant_marker_level_operation

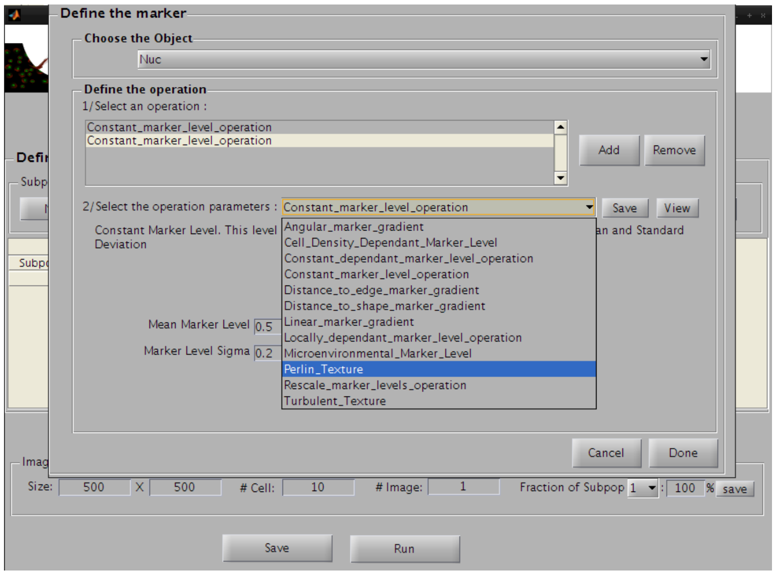
Constant Marker Level. This level is sampled from a Normal Distribution with Specified Mean and Standard Deviation

Mean Marker Level 0.5

Marker Level Sigma 0.1

Figure 14 - Define the first operation for the pair DAPI/Nuc

Then click on the ***Save*** button to save this operation and press the ***Add*** button to add a new one. Click on the operation list and select ***Perlin_Texture***. This operation adds a noise-like texture, and will make ***DAPI*** look more realistic on ***Nuc***.


Define the marker

Choose the Object: Nuc

Define the operation

1/Select an operation :

Constant_marker_level_operation
Constant_marker_level_operation

2/Select the operation parameters : Constant_marker_level_operation

Angular_marker_gradient
Cell_Density_Dependant_Marker_Level
Constant_dependant_marker_level_operation
Constant_marker_level_operation
Distance_to_edge_marker_gradient
Distance_to_shape_marker_gradient
Linear_marker_gradient
Locally_dependant_marker_level_operation
Microenvironmental_Marker_Level
Perlin_Texture
Rescale_marker_levels_operation
Turbulent_Texture

Mean Marker Level 0.5

Marker Level Sigma 0.2

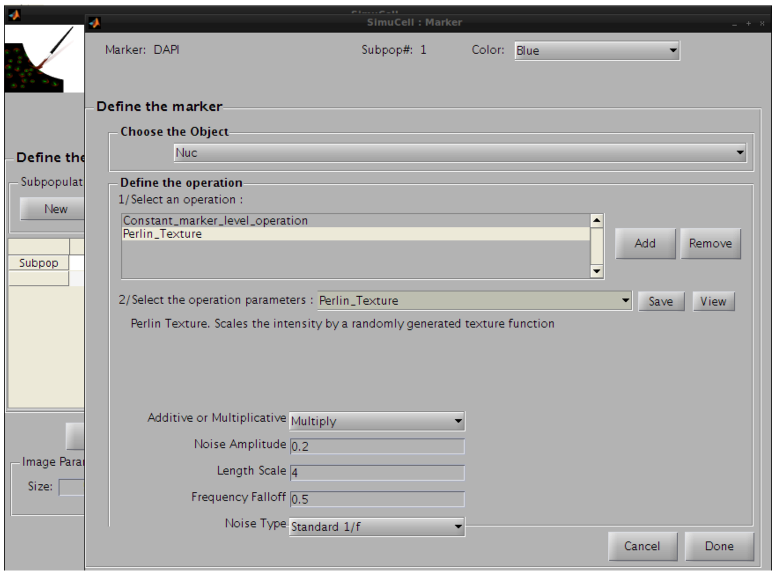
Size: 500 X 500 # Cell: 10 # Image: 1 Fraction of Subpop 1 : 100 %

Figure 15 – Select the cell corresponding to the Marker/Object pair DAPI/Nuc

We will use the following parameters:

- Additive and Multiplicative: Multiply (scale the intensity rather than add to it)

- Noise Amplitude: 0.2
- Length Scale: 4 (2 – gives coarse variation , while 6 is very fine)
- Frequency FallOff: 0.5 (the weight given to higher frequency noise)
- Noise Type: Standard 1/f (the other choice is turbulent which is more abrupt)


SimuCell : Marker

Marker: DAPI   Subpop#: 1   Color: Blue

Define the marker

Choose the Object: Nuc

Define the operation

1/ Select an operation :

Constant_marker_level_operation
Perlin_Texture

2/ Select the operation parameters : Perlin_Texture

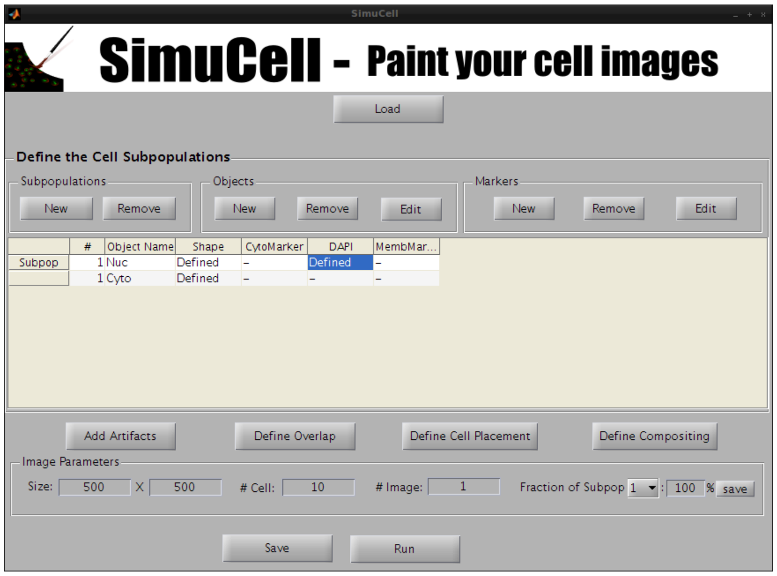
Perlin Texture. Scales the intensity by a randomly generated texture function

Additive or Multiplicative: Multiply
Noise Amplitude: 0.2
Length Scale: 4
Frequency Falloff: 0.5
Noise Type: Standard 1/f

Figure 16 - Set the Perlin Texture operation parameters for the pair DAPI/Nuc

Press the ***Save*** button to save this operation. The operation's name on the operation list will be changed from ***Constant_marker_level_operation*** to ***Perlin_Texture*** and the parameters for this operation will be saved. Then press ***Done*** to go back to the main interface.

You can now see that the marker ***DAPI*** for the object/shape ***Nuc*** has been defined.



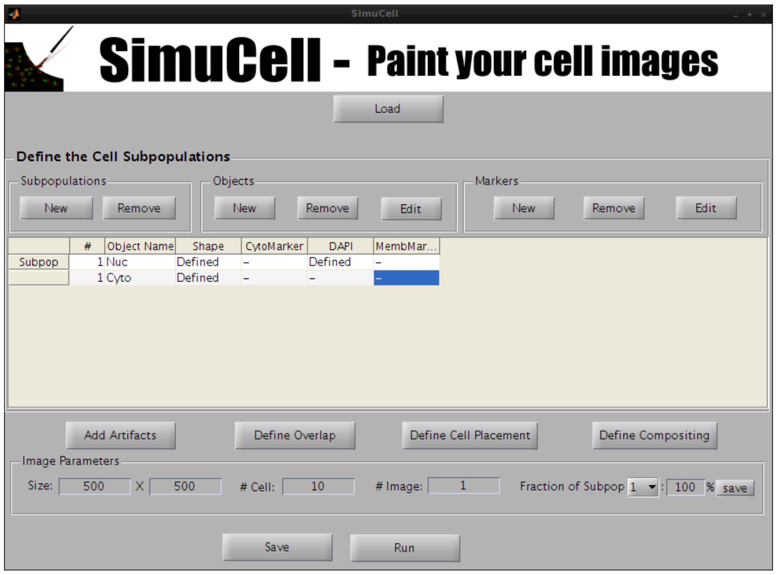
	#	Object Name	Shape	CytoMarker	DAPI	MembMar...
Subpop	1	Nuc	Defined	–	Defined	–
	1	Cyto	Defined	–	–	–

Size: 500 X 500 # Cell: 10 # Image: 1 Fraction of Subpop 1 : 100 %

Figure 17 - The Marker DAPI for the Object/Shape Nuc has been defined

Define the marker MembMarker for for the Object/Shape Cyto

In the subpopulation table, select the marker ***MembMarker*** for the object/shape ***Cyto***.



	#	Object Name	Shape	CytoMarker	DAPI	MembMar...
Subpop	1	Nuc	Defined	–	Defined	–
	1	Cyto	Defined	–	–	–

Size: 500 X 500 # Cell: 10 # Image: 1 Fraction of Subpop 1 : 100 %

Figure 18 - Select the Marker MembMarker for the Shape/Object Cyto

Then click ***Edit*** of the subpanel ***Markers***. The window to define the marker will show up.

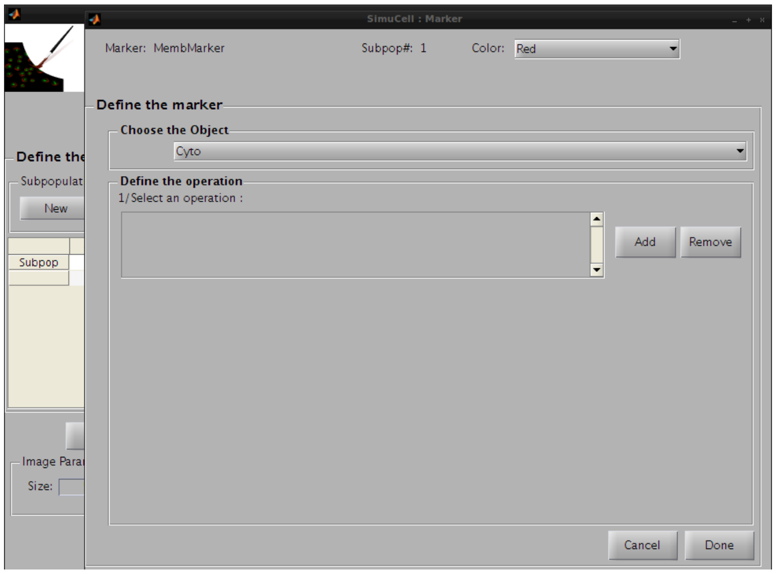
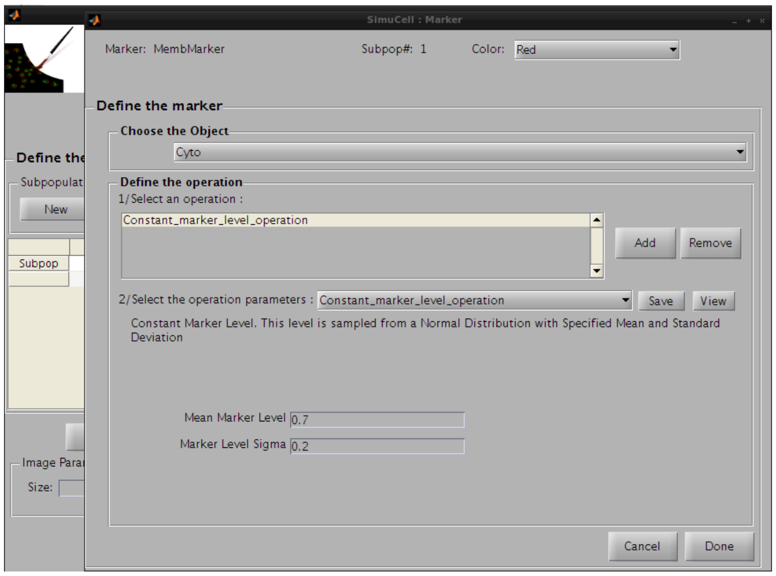


Figure 19 - Define the pair MembMarker/Cyto by adding operations

Click on ***Add*** in the subpanel ***Define Operation*** to add an operation. The default operation ***Constant_marker_level_operation*** will appear in the operation list. We will use first this operation and set the following parameters:

- Mean Marker Level: 0.7
- Marker Level Sigma: 0.2

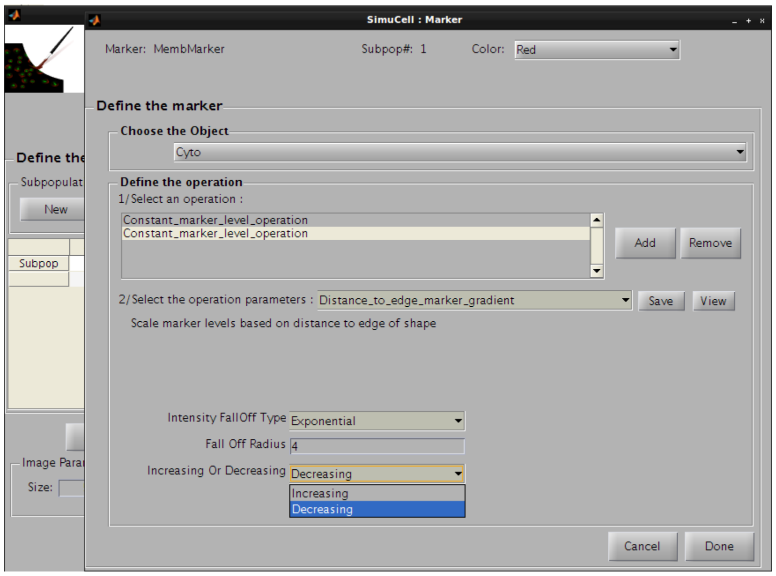


Constant Marker Level. This level is sampled from a Normal Distribution with Specified Mean and Standard Deviation

Mean Marker Level 0.7
Marker Level Sigma 0.2

Figure 20- Define the first operation for the pair MembMarker/Cyto

Then click on the ***save*** button to save this operation and press the ***add*** button to add a new one. Click on the operation list and select ***Distance_to_edge_marker_gradient***. This will be used to concentrate the high intensity regions of the marker close to the edge so that ***MembMarker*** will appear localized to the membrane



Marker: MembMarker Subpop#: 1 Color: Red

Cyto

Constant_marker_level_operation
Constant_marker_level_operation

2/ Select the operation parameters : Distance_to_edge_marker_gradient

Scale marker levels based on distance to edge of shape

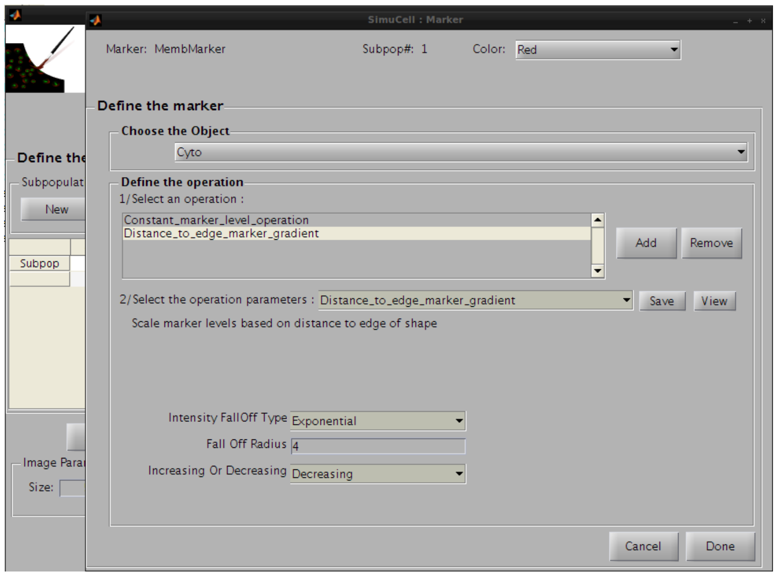
Intensity FallOff Type: Exponential
Fall Off Radius: 4
Increasing Or Decreasing: Decreasing
Increasing
Decreasing

Figure 21 - Add the second operation Distance to edge marker gradient and set the parameters

We will use the following parameters:

- Intensity FallOff Type: Exponential (functional form of intensity falloff with distance to the edge)
- Fall Off Radius: 4 (the number of pixels over which the intensity falls by 1/e)
- Increasing Or Decreasing: Decreasing (higher close to the edge)

Press the ***Save*** button to save this last operation



Marker: MembMarker   Subpop#: 1   Color: Red

Choose the Object: Cyto

1/ Select an operation :
Constant_marker_level_operation
Distance_to_edge_marker_gradient

2/ Select the operation parameters : Distance_to_edge_marker_gradient

Scale marker levels based on distance to edge of shape

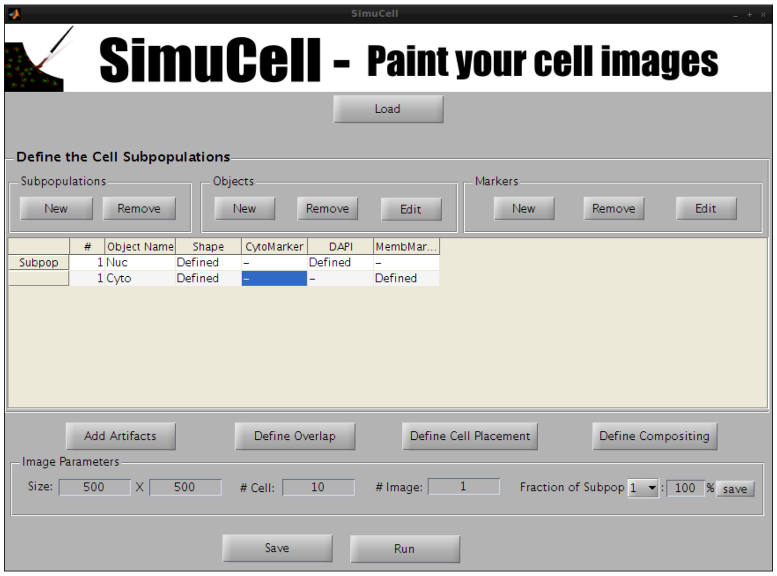
Intensity FallOff Type: Exponential
Fall Off Radius: 4
Increasing Or Decreasing: Decreasing

Figure 22-Save the second operation

and then the button ***Done*** to go back to the main interface. You can see that the marker ***MembMarker*** for the object/shape ***Cyto*** has been defined.

Define the marker CytoMarker for for the object/shape Cyto

In the subpopulation table, select the marker ***CytoMarker*** for the object/shape ***Cyto***.

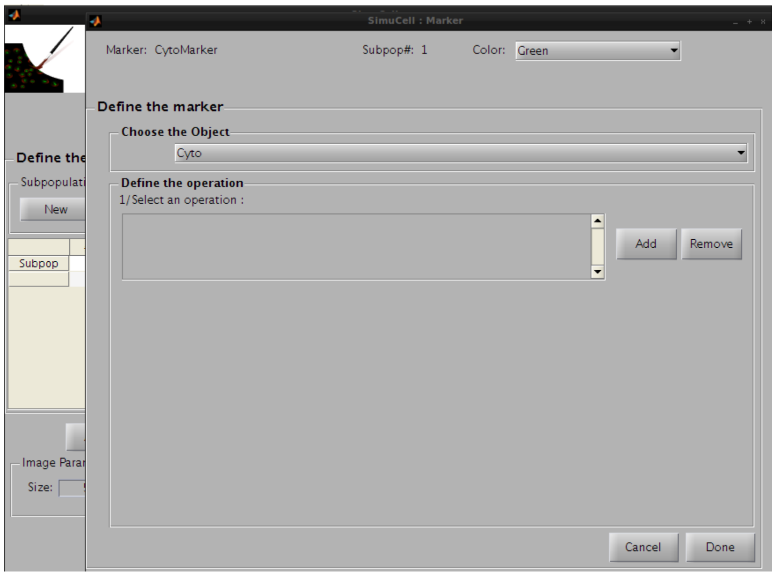


	#	Object Name	Shape	CytoMarker	DAPI	MembMar...
Subpop	1	Nuc	Defined	–	Defined	–
	1	Cyto	Defined	–	–	Defined

Size: 500 X 500   # Cell: 10   # Image: 1   Fraction of Subpop 1 : 100 %

Figure 23 - Select the marker CytoMarker for the Object/Shape Cyto

Then click ***Edit*** of the subpanel ***Markers***. The window to define the marker will appear.

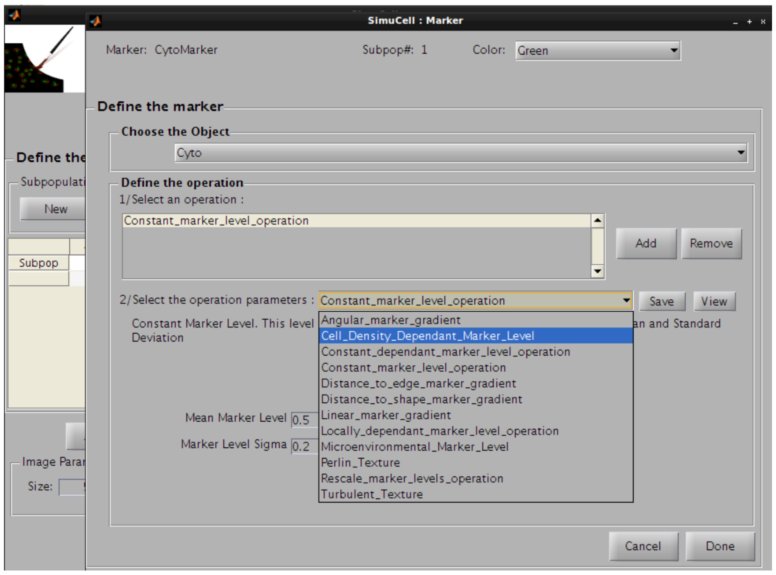


Marker: CytoMarker Subpop#: 1 Color: Green

Choose the Object: Cyto

Figure 24 - Define the Marker CytoMarker for the Object/Shape Cyto

Click on *Add* in the subpanel *Define Operation* to add an operation. The default operation *Constant_marker_level_operation* will appear in the operation list. We won't use this one but instead the operation *Cell_Density_Dependant_Marker_Level* to produce a cell density dependent marker intensity.


SimuCell : Marker
Define the marker
Choose the Object
Define the operation
1/ Select an operation :
Add
Remove
Save
View
Cancel
Done
Define the
Subpopulati
New
Subpop
Image Parar
Size:
Angular_marker_gradient
Cell_Density_Dependant_Marker_Level
Constant_dependant_marker_level_operation
Constant_marker_level_operation
Distance_to_edge_marker_gradient
Distance_to_shape_marker_gradient
Linear_marker_gradient
Locally_dependant_marker_level_operation
Microenvironmental_Marker_Level
Perlin_Texture
Rescale_marker_levels_operation
Turbulent_Texture

Marker: CytoMarker Subpop#: 1 Color: Green

Choose the Object: Cyto

1/ Select an operation : Constant_marker_level_operation

2/ Select the operation parameters : Constant_marker_level_operation

Constant Marker Level. This level … an and Standard Deviation

Mean Marker Level 0.5

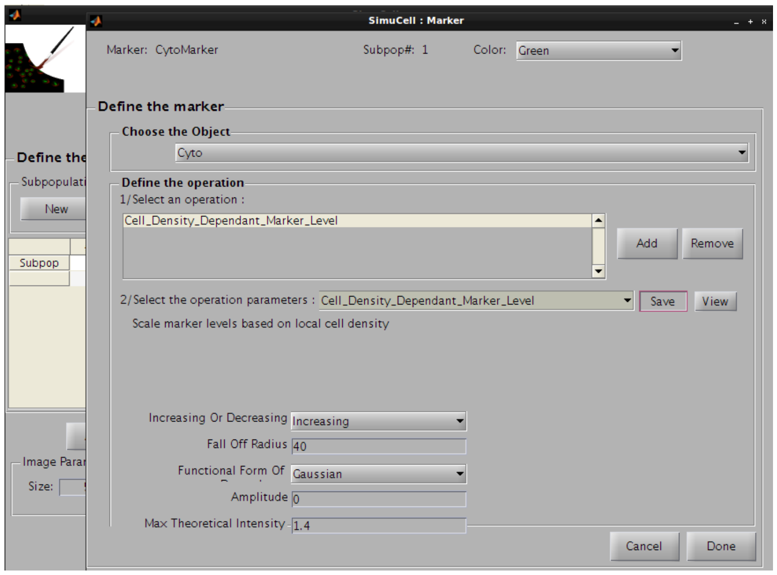
Marker Level Sigma 0.2

Figure 25 - Use Cell density dependant marker level as first operation of the pair CytoMarker/Cyto

The following parameters will be used:

- Increasing or Decreasing: Increasing (Intensity increases with cell density)

- Fall Off Radius: 40 (pixels over which intensity falls by 1/e)
- Functional Form Of Gradient: Gaussian (functional form of intensity falloff)
- Amplitude: 0 (minimum intensity level)
- Max Theoretical Intensity: 14



Marker: CytoMarker   Subpop#: 1   Color: Green

Cyto

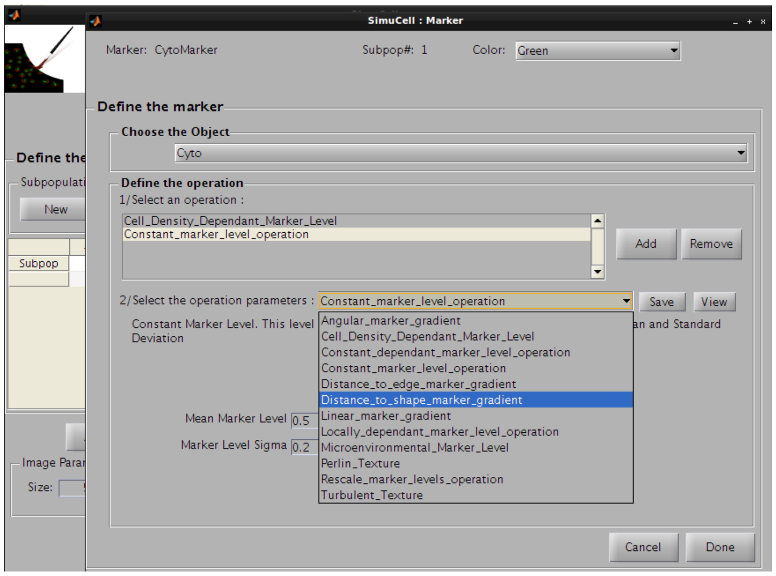
1/ Select an operation :
Cell_Density_Dependant_Marker_Level

2/ Select the operation parameters : Cell_Density_Dependant_Marker_Level
Scale marker levels based on local cell density

Increasing Or Decreasing	Increasing
Fall Off Radius	40
Functional Form Of	Gaussian
Amplitude	0
Max Theoretical Intensity	1.4

Figure 26- Define parameters for the first operation of the pair CytoMarker/Cyto

Then click on the ***save*** button to save this operation and press the ***add*** button to add a new one. Click on the operation list and select ***Distance_to_shape_marker_gradient***, so the ***CytoMarker*** marker for ***Cyto*** object/shape will appear with a light gradient from the nucleus.


SimuCell : Marker
Define the marker
Choose the Object
Define the operation
1/ Select an operation :
Add
Remove
Save
View
Angular_marker_gradient
Cell_Density_Dependant_Marker_Level
Constant_dependant_marker_level_operation
Constant_marker_level_operation
Distance_to_edge_marker_gradient
Distance_to_shape_marker_gradient
Linear_marker_gradient
Locally_dependant_marker_level_operation
Microenvironmental_Marker_Level
Perlin_Texture
Rescale_marker_levels_operation
Turbulent_Texture
Cancel
Done
Define the
Subpopulati
New
Subpop
Image Parar
Size:

Marker: CytoMarker   Subpop#: 1   Color: Green

Cyto

Cell_Density_Dependant_Marker_Level
Constant_marker_level_operation

2/ Select the operation parameters : Constant_marker_level_operation

Constant Marker Level. This level ... an and Standard Deviation

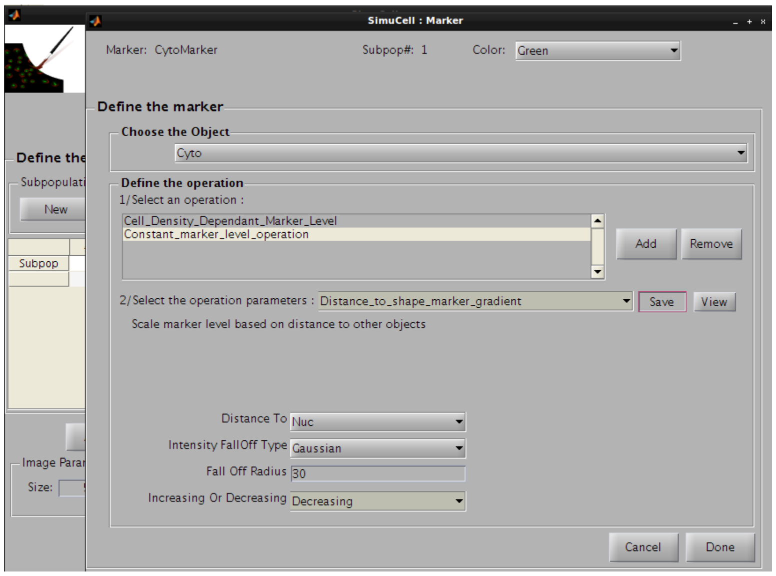
Mean Marker Level 0.5

Marker Level Sigma 0.2

Figure 27 - Use Distance to shape marker gradient for the second operation for the pair CytoMarker/Cyto

The following parameters will be used:

- Distance To: Nuc
- Intensity FallOff Type: Gaussian
- Fall Off Radius: 30
- Increasing or Decreasing: Decreasing



Marker: CytoMarker   Subpop#: 1   Color: Green

Cyto

Cell_Density_Dependant_Marker_Level
Constant_marker_level_operation

2/ Select the operation parameters : Distance_to_shape_marker_gradient

Scale marker level based on distance to other objects

Distance To Nuc

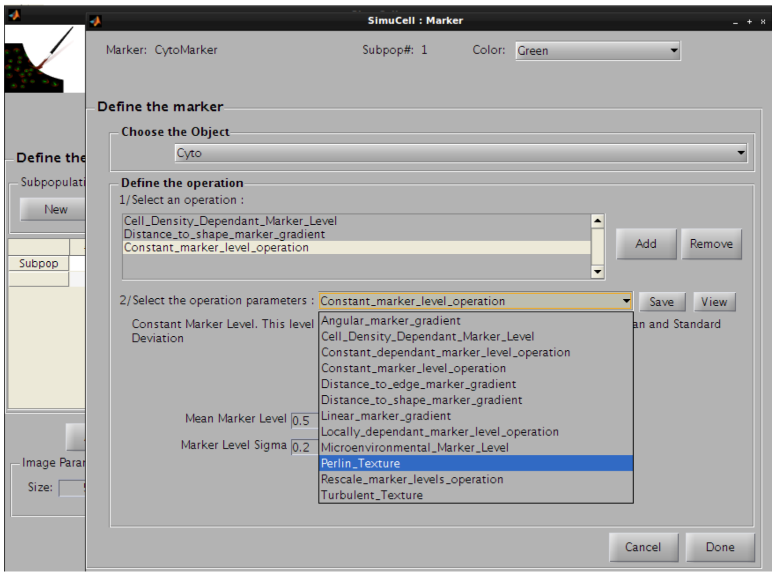
Intensity FallOff Type Gaussian

Fall Off Radius 30

Increasing Or Decreasing Decreasing

Figure 28 - Define parameters for the second operation of the pair CytoMarker/Cyto

Then click on the ***Save*** button to save this operation and press the ***Add*** button to add the last operation. Click on the operation list and select ***Perlin_Texture***, so the ***CytoMarker*** marker for ***Cyto*** object/shape will look more realistic.


SimuCell : Marker

Marker: CytoMarker Subpop#: 1 Color: Green

Define the marker

Choose the Object: Cyto

Define the operation

1/Select an operation :

Cell_Density_Dependant_Marker_Level
Distance_to_shape_marker_gradient
Constant_marker_level_operation

2/Select the operation parameters : Constant_marker_level_operation

Constant Marker Level. This level ... an and Standard Deviation

Mean Marker Level 0.5

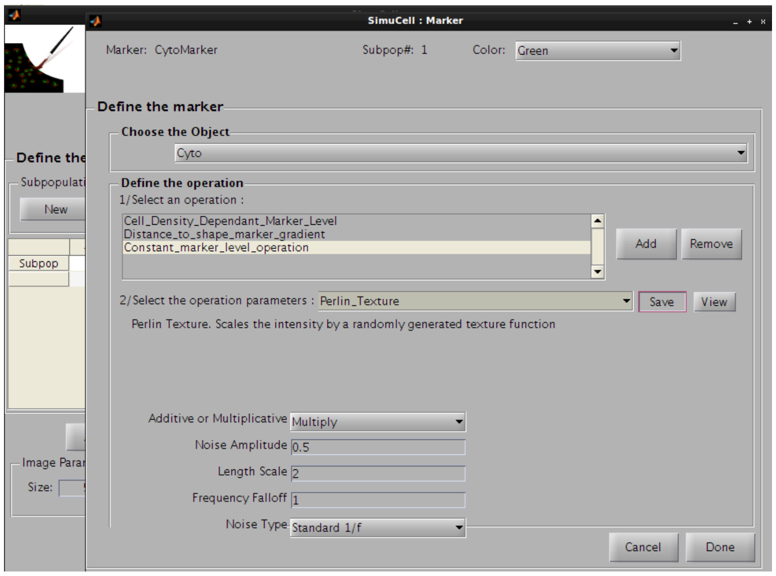
Marker Level Sigma 0.2

Add
Remove
Save
View
Angular_marker_gradient
Cell_Density_Dependant_Marker_Level
Constant_dependant_marker_level_operation
Constant_marker_level_operation
Distance_to_edge_marker_gradient
Distance_to_shape_marker_gradient
Linear_marker_gradient
Locally_dependant_marker_level_operation
Microenvironmental_Marker_Level
Perlin_Texture
Rescale_marker_levels_operation
Turbulent_Texture
Cancel
Done
Define the
Subpopulati
New
Subpop
Image Para
Size:

Figure 29 - Use Perlin Texture for the last operation for the pair CytoMarker/Cyto

The following parameters will be used:

- Additive and Multiplicative: Multiply
- Noise Amplitude: 0.5
- Length Scale: 2
- Frequency FallOff: 1
- Noise Type: Standard 1/f



SimuCell : Marker

Marker: CytoMarker   Subpop#: 1   Color: Green

Define the marker

Choose the Object: Cyto

Define the operation

1/ Select an operation :

```
Cell_Density_Dependant_Marker_Level
Distance_to_shape_marker_gradient
Constant_marker_level_operation
```

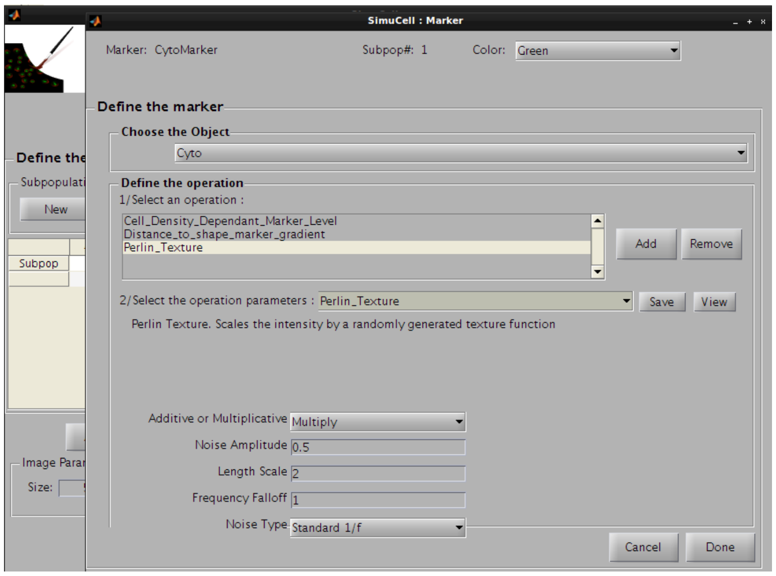
2/ Select the operation parameters : Perlin_Texture

Perlin Texture. Scales the intensity by a randomly generated texture function

```
Additive or Multiplicative  Multiply
           Noise Amplitude  0.5
              Length Scale  2
         Frequency Falloff  1
                Noise Type  Standard 1/f
```

Figure 30 - Define parameters for the last operation of the pair CytoMarker/Cyto

Press the ***Save*** button to save this operation. The operation's name on the operation list will be changed from ***Constant_marker_level_operation*** to ***Perlin_Texture*** and the parameters for this operation will be saved.



SimuCell : Marker

Marker: CytoMarker   Subpop#: 1   Color: Green

Define the marker

Choose the Object: Cyto

Define the operation

1/ Select an operation :

```
Cell_Density_Dependant_Marker_Level
Distance_to_shape_marker_gradient
Perlin_Texture
```

2/ Select the operation parameters : Perlin_Texture

Perlin Texture. Scales the intensity by a randomly generated texture function

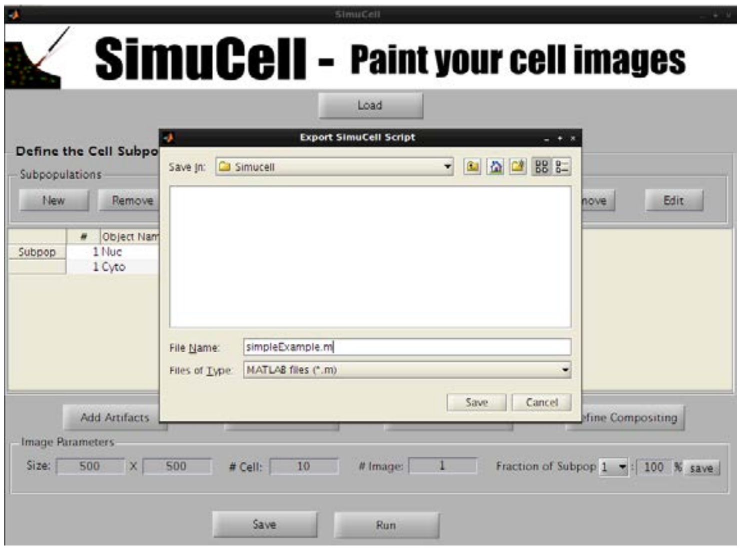
```
Additive or Multiplicative  Multiply
           Noise Amplitude  0.5
              Length Scale  2
         Frequency Falloff  1
                Noise Type  Standard 1/f
```

Figure 31 - Save the last operation

Then press ***Done*** to go back to the main interface. You can see that the marker ***MembMarker*** for the object/shape ***Cyto*** has been defined.

Step 6: Save the SimuCell synthetic model

We are now almost done with this tutorial. You should first save your work by clicking on the ***Save*** on the main interface.



Export SimuCell Script

Save In: Simucell

File Name: simpleExample.m

Files of Type: MATLAB files (*.m)

Save Cancel

#	Object Nam
Subpop 1	Nuc
1	Cyto

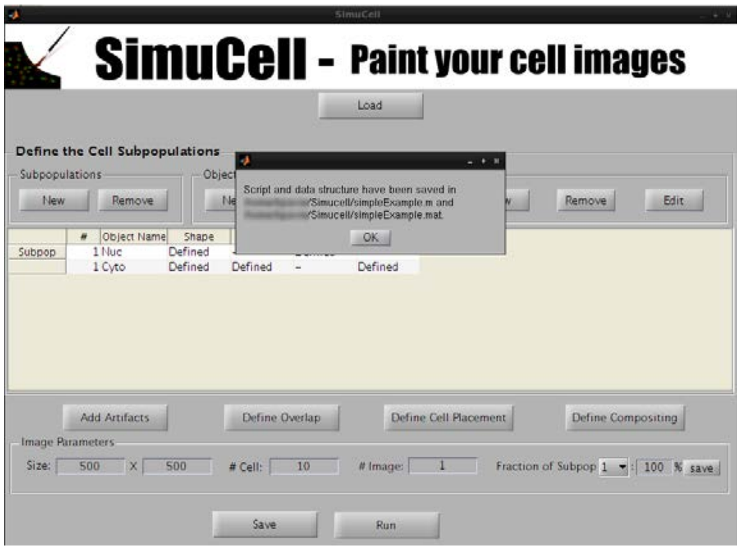
Image Parameters

Size: 500 X 500 # Cell: 10 # Image: 1 Fraction of Subpop 1 : 100 % save

Figure 32- Save the SimuCell model

Select a directory to save the files and enter a file name (don't forget to add the extension ***.m*** at this end), like `simpleExample.m`.

Two files will be saved, one ***.m*** which contains a MATLAB script that you could use to generate your synthetic images, one ***.mat*** that contain the SimuCell MATLAB structure that you can load from the SimuCell interface.



Script and data structure have been saved in
[redacted]/Simucell/simpleExample.m and
[redacted]/Simucell/simpleExample.mat.

OK

#	Object Name	Shape			
Subpop 1	Nuc	Defined			
1	Cyto	Defined	Defined	–	Defined

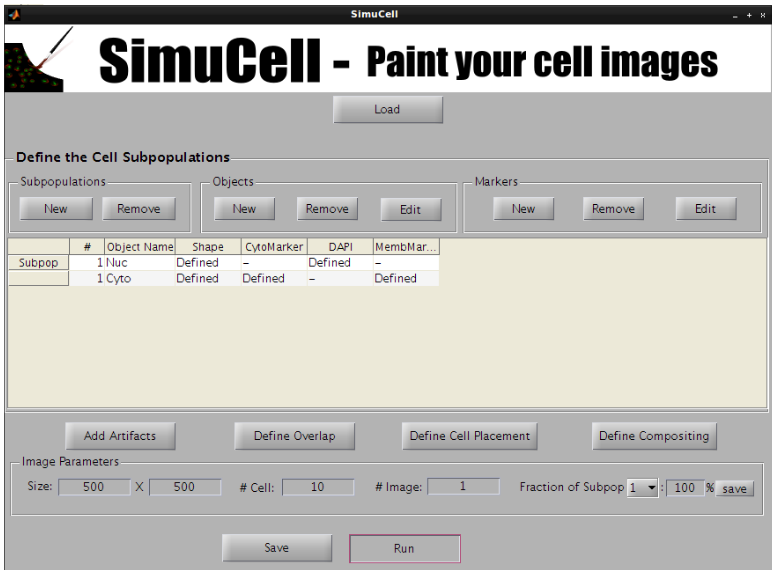
Image Parameters

Size: 500 X 500 # Cell: 10 # Image: 1 Fraction of Subpop 1 : 100 % save

Figure 33 - SimuCell data file and script file save successfully!

Step 7: Run your SimuCell model to visualize your image

Press the ***Run*** button to run SimuCell and generate the image.



	#	Object Name	Shape	CytoMarker	DAPI	MembMar...
Subpop	1	Nuc	Defined	–	Defined	–
	1	Cyto	Defined	Defined	–	Defined

Size: 500 X 500 # Cell: 10 # Image: 1 Fraction of Subpop 1 : 100 %

Figure 34 – Press the Run button to run your SimuCell model

A waiting bar will popup and after a pair of second/minute you should see your images.

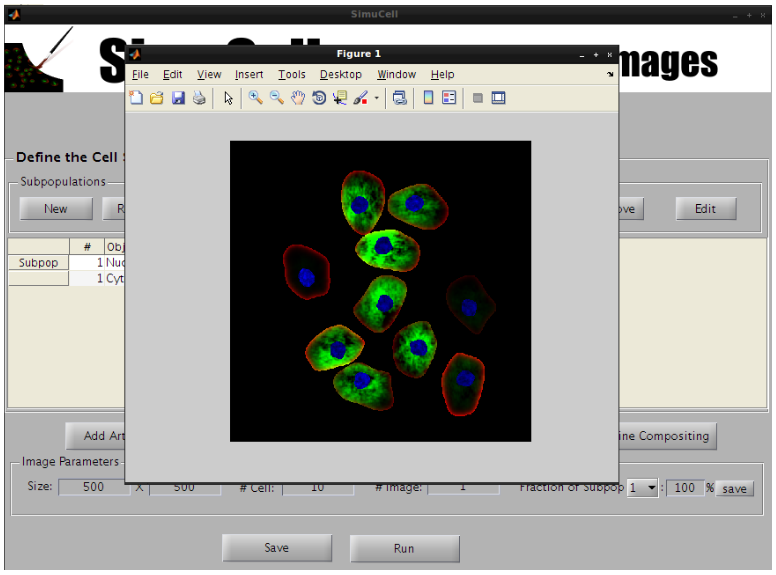


Figure 35 - Your first cell images through SimuCell

To really see what’s going on, you can open the ***.m*** file and review the MATLAB script to see how SimuCell operate.

For this tutorial, we didn’t discuss other parameters like placement, cell and images artifact, overlapping and compositing. Since we didn’t define them, SimuCell will use automatically default settings for those parameters.

SimuCell Scripts

Another way to generate synthetic images using SimuCell is to create and run scripts.

Some documented script examples are located in `simucell/saved_data`

`simucellTestSimple.m`	The simple SimuCell example described in the Basic SimuCell tutorial
`polarized.m`	A more complex script producing two subpopulation of polarized cells
`microenv.m`	A script example to show how to add micro-environment effect.

To run these scripts, use `simucell.m` (located in `simucell/scr)` that we used previously in the tutorial.

```
>cd simucell/src
```

```
>result=simucell('simucellTestSimple')
```

`Where 'simucellTestSimple'` is the script name.

This will execute the `simucellTestSimple.m` script, plot one image and return the result variable:

```
> result =

    subpopulation_numbers_of_cells: [10x1 double]

                         RGB_image: [500x500x3 double]

            mask_of_object_by_cell: [1x10 struct]

          marker_on_object_by_cell: [1x10 struct]

                    channel_images: [1x1 struct]
```

Here is a quick description of the output variable:

- `Result.subpopulation_numbers_of_cells` contains the subpopulation number for each cell.
- `result.RGB_image` contains the RGB synthetic image generated using the script.
- `result.mask_of_object_by_cell` contains the mask of each object organized per cell.
- `result.marker_of_object_by_cell` contains the pixel intensity distribution of every marker on each object organized per cell.
- `result.channel_image` contains the synthetic images organized by color (channels).

Additional parameters may be used (see example below):

```
>simucell('simucellTestSimple', 'save_results', 2)
```

image	Will additionally plot the first RGB image result. If the number of images is bigger than one, it will also offer to save these images.
save_results	Will offer to save the result into a *.mat file.
save_params	This option won't run the SimuCell engine and will just save the script parameters into a .mat file that can be loaded into the GUI interface using the load button.