Creating Images w/ Simulated Diffraction-Limited Objects.

Last updated 4 March 2020.

Contents

1. Downloading “PSF Generator” & “simulate\_psf\_image”
2. Installation and Startup
3. Defining PSF Properties
4. User Arguments
5. Creating Simulated Image(s)
6. Output
7. References

1. Downloading “PSF Generator” & “simulate\_psf\_image”

**PSF Generator** is an open-source software package that was created by Kirshner and colleagues (1) that allows one to generate 3D models of a microscope’s PSF. These models can in turn be used to simulate data collected in epifluorescence microscopy.

The package utilized in this set of scripts is the ‘PSFGenerator.jar’ standalone java application, and can be downloaded at the following location:

<http://bigwww.epfl.ch/algorithms/psfgenerator/>

Further instructions on installation are available at the indicated web page, but the exact installation process used is indicated within Section 2 of this document (Installation and Startup). The developers request that PSF Generator be downloaded from their source and not re-packaged with other software tools.

The ‘simulate\_psf\_image’ package can be downloaded from the Lee Lab GitHub, which can be reached using the following link:

<https://github.com/recleelab/simulate_psf_image>

The GitHub page has a “clone/download” button which can be used to download a .zip file containing the entirety of this package.

2. Installation and Startup

In order for PSF Generator to run, the <jar> file (PSFGenerator.jar) needs to be placed in the java folder of the MATALB application currently installed on your machine. Save the downloaded <jar> file (PSFGenerator.jar) in the java folder of MATLAB. Note that you may need admin access to add <jar> files to the MATLAB\<VERSION>\java folder.

Once the PSFGenerator.jar has been added to the ‘java’ folder within your current MATLAB version, open up MATLAB. Unzip the ‘simulate\_psf\_image’ package downloaded from GitHub and copy and paste the ‘simulate\_psf\_image’ folder into the MATLAB folder window. Right click on the folder, navigate to “Add to Path”, and click “Selected Folders and Subfolders”. Then navigate into the “simulate\_psf\_image” directory by double-clicking the “simulate\_psf\_image” directory within the ‘Current Folder’ window.

Before “simulate\_psf\_image” can be used, the PSFGenerator.jar package must be added to the current path. To do this via MATLAB’s command window, type the following (replace bolded version with your own MATLAB version):

[WINDOWS]

> javaaddpath(‘C:\Program Files\MATLAB\**R2017b**\java\PSFGenerator.jar’);

[MAC]

> javaaddpath(/Applications/MATLAB\_**R2017b**.app/java/PSFGenerator.jar);

3. Defining PSF Properties

The properties defining the Point Spread Function (PSF) model to be constituted by the PSF Generator are set using a <txt> file (‘config.txt’). The original configuration file that is packaged with PSF Generator can be found within the “simulate\_psf\_image” package. The configuration file used by the Lee Lab for creating images simulating EGFP-NEMO complexes in U2OS cells (2) is also provided (‘config\_NEMO.txt’). Create your own configuration file by updating feature properties in the original configuration file and saving your own copy which best reflects your microscope’s PSF.

4. User Arguments

The main function of the “simulate\_psf\_image” package is the following script:

> simulate\_psf\_image.m

To run the script, first ensure that PSFGenerator.jar has both (1) been added to the java folder of the MATLAB application (see Section 2) and (2) added to the workplace path using the system-appropriate command (see Section 2).

Before running the script, it is beneficial to examine the user arguments to define the simulated image’s dimensions, PSF, etc. Open the script ‘simulate\_psf\_image.m’ in the MATLAB editor. User arguments, along with their descriptions, can be found in lines 41—114 of the script. Their descriptions are restated here.

* **psf\_config\_filename** : string containing the filename of the configuration file which should be used to define the PSF (see Section 3, PSFGenerator documentation (1)) (default ‘config\_NEMO.txt’).
* **psf\_config\_filepath** : string containing the filepath where the configuration file can be found on the current system (default current directory).
* **image\_dimensions** : vector containing the [X, Y, Z] measurements for the simulated image in pixels (X, Y) and slices (Z) (default 512, 512, 64).
* **number\_of\_signals** : double value indicating the number of theoretical PSFs to insert into the image (default 1500).
* **lower\_int\_bound** : lower bound for intensity manipulation. A random value taken from the distribution [lower\_int\_bound, 1] will be applied to all values of the PSF. This has the effect of randomly dimming the PSF to some lower bound, with the default being half of the original PSF’s intensity (0.5).
* **include\_cell** : Boolean value indicating whether to include a polygon in the image which can approximate the cellular background (default True).
* **signals\_in\_cell\_only** : Boolean value indicating whether PSFs should only be inserted within the bounds of the polygon approximating cellular background (default True)
* **noise\_range** : vector of integer values indicating standard deviations of the distributions of Gaussian noise to be applied to copies of the simulated image. These images are saved alongside the original image in the same folder (default [50, 300]).
* **output\_filename** : string indicating what name to save created image(s) under. Default set to empty, wherein a prompt will arise after the user has confirmed to save the simulated image requesting an output filename / location.
* **output\_filepath** : string containing the filepath where the output images should be stored (default current directory).

5. Creating Simulated Image(s)

After user arguments have been assigned, ensure that “simulate\_psf\_image” is on the current MATLAB path by right-clicking the folder within the ‘Current Folder’ window and selecting ‘Add to Path’ > ‘Selected Folders and Subfolders’. If the prompt ‘Add to Path’ doesn’t appear, the folder is already on the current path.

The main script for this package is ‘simulate\_psf\_image.m’. To run this script, either open the script by double-clicking on ‘simulated\_psf\_image.m’ in the ‘Current Folder’ window and clicking ‘Run’ in the Editor Toolbar, or type ‘simulated\_psf\_image’ in MATLAB’s Command Window.

The script will accept the user arguments as defined by the user (see Section 4) and create a simulated image containing the theoretical PSFs defined by the configuration file (see Section 3). Once a simulated image made with the user arguments is created, a preview will pop up along with a prompt for the user to continue with the process. If the process is terminated at this point, the user arguments can be adjusted and the script can be run again.

Once accepting the simulated image, the image is written to file. If no output file name / location was indicated in the user arguments, a prompt asking the user for the file name / location will appear. After the simulated image is written to file, copies of the image with noise added to the image will be written to file, so long as the user has defined a set of values for the ‘noise\_range’ argument. In addition to the original simulated image and any copies with noise, a mat-file and excel-spreadsheet will be saved containing the PSF coordinates, random intensity scalar value, and user arguments used to create the simulated image(s) (see Section 6).

6. Output

Images / mat-files / excel spreadsheets created using this script package will be saved to their own folder which will contain the following:

* <FILENAME>.tif : the simulated image without any noise
* <FILENAME>\_NOISE\_STD\_XX.tif : simulated image with noise (XX = STD)
* <FILENAME>\_INFO.mat : mat-file containing user arguments, PSF insertion coordinates, and intensity values applied to each PSF (randomly assigned prior to insertion into simulated image). Will additionally save coordinates for polygon representing the cell background if that argument was set by the user.
* <FILENAME>\_INFO.xlsx : excel spreadsheet containing user arguments, PSF insertion coordinates, and intensity values applied to each PSF (randomly assigned prior to insertion into simulated image). Will additionally save coordinates for polygon representing the cell background if that argument was set by the user.

7. References

1. KIRSHNER, H., F. AGUET, D. SAGE, and M. UNSER. 2013. 3-D PSF fitting for fluorescence microscopy: implementation and localization application. Journal of Microscopy 249(1):13-25.

2. Kowalczyk, G. J., J. A. Cruz, Y. Guo, Q. Zhang, N. Sauerwald, and R. E. C. Lee. 2019. dNEMO: a tool for quantification of mRNA and punctate structures in time-lapse images of single cells. bioRxiv:855213.