SMIS Manual

Latest version: 2.1

=======================================================

Copyright (C) 2022 Dominique Bourgeois

E-mail: [dominique.bourgeois@ibs.fr](mailto:dominique.bourgeois@ibs.fr)

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

=======================================================

# Introduction

* SMIS (Single Molecule Imaging Simulator) is a Matlab-based simulation software that enables to simulate a large variety of single-molecule fluorescence imaging experiments in Widefield mode.
* The novelty of SMIS is the *advanced description of the fluorophores* used in the simulations, notably taking into account their *spectral and photophysical characteristics*.
* Examples of simulations that can be performed include PALM, dSTORM, sptPALM, PAINT, in 2D or 3D.
* Multicolor experiments can be simulated with unlimited number of colors.
* Complex laser excitation schemes can be simulated with unlimited number of lasers.
* Complex diffusion patterns of the fluorophores can be simulated with unlimited number of diffusion states.
* SMIS outputs .tif image stacks and ground truth .mat data.

# Reference

Bourgeois, D.; Single Molecule Imaging Simulations with Advanced Fluorophore Photophysics. <https://www.biorxiv.org/content/10.1101/2022.06.14.496133v2>

# Download and Installation

The software can be used either as a standalone application for Windows, MacOS or Linux, or as a MATLAB app.

SMIS was developed under Windows. *Proper running of SMIS under MacOS or Linux has not been thoroughly checked*.

## Downloading SMIS from Github

* Go to: <https://github.com/DominiqueBourgeois/SMIS>
* For **Windows** and **Linux**:
  + Use the default SMIS branch called *main\_LINUX\_WINDOWS*
  + Click on the green button “Code”, and select “download ZIP”. This will download the file: “SMIS-main\_LINUX\_WINDOWS.zip”
* For **MacOS**:
  + Switch to the Github SMIS branch called *MACOS*
  + Click on the green button “Code”, and select “download ZIP”. This will download the file: “SMIS-MACOS.zip”
* Finally, unzip the SMIS .zip file in your preferred directory.

## Running SMIS

* **Standalone** SMIS
  + To run the standalone SMIS, the Matlab runtime must be installed on your computer. The runtime with proper version should be installed. To download the Matlab runtime, go to <https://fr.mathworks.com/products/compiler/matlab-runtime.html>
  + **Windows**
    - Make sure Matlab runtime 2022a (9.12) is installed on your computer.
    - The SMIS executable is found in: SMIS/STANDALONE/DISTRIBUTE/WINDOWS
    - **To execute SMIS, double-click on “SMIS.exe”**
  + **Linux**
    - Make sure Matlab runtime 2021b (9.11) is installed on your computer.
    - The SMIS executable is found in: SMIS/STANDALONE/DISTRIBUTE/LINUX
    - **To execute SMIS**, open a terminal, move to the directory where SMIS is installed and type at the prompt: “*./run\_SMIS.sh <mcr\_directory>*”, where <mcr\_directory> is the location of the Matlab runtime.
  + **MacOS**
    - Make sure Matlab runtime 2022b (9.13) is installed on your computer.
    - The SMIS executable is found in: SMIS/STANDALONE/DISTRIBUTE/MACOS
    - **To execute SMIS**, open a terminal, move to the directory where SMIS is installed and type at the prompt: “*./run\_SMIS.sh <mcr\_directory>*”, where <mcr\_directory> is the location of the Matlab runtime.
* Using the **SMIS app** with Matlab
  + To properly run SMIS with Matlab, you need the **Image Processing Toolbox**, and preferably the **Parallel Computing Toolbox** (not compulsory).
  + In Matlab, go to the APPS tab, and click on “*Install App*”
    - For **Windows** select the “SMIS\_Windows.mlappinstall” located in SMIS/APP
    - For **Linux** select the “SMIS\_Linux.mlappinstall” located in SMIS/APP
    - For **MacOS** select the “SMIS.mlappinstall” located in SMIS/APP

# SMIS quick help

**In SMIS, interactive help is available by moving the mouse to the desired field.**

## Getting familiar with SMIS:

To start learning about SMIS, it is advisable to **load simulation examples** by clicking on: “*Load simulations*”.

Then navigate throughout the different SMIS submenus. Enter a proper output directory for the simulation and click on “Launch simulation”.

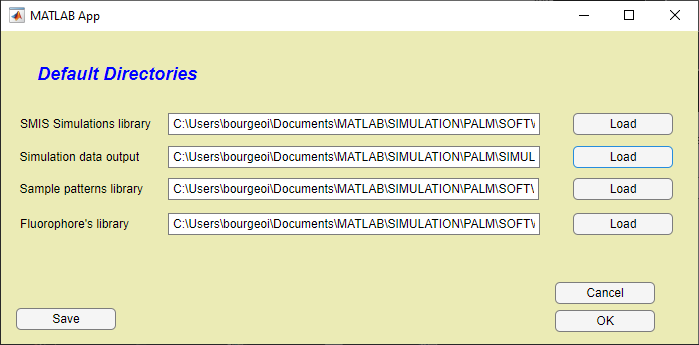
*Some of the simulations can take very long to run. To run faster, in the “Setup Fluorophore” menu, you may want to reduce the number of used molecules. *

## Main steps to design a simulation :

1. If there is a similar simulation available, load it with « *Load simulation parameters* ».
2. Enter the name of your simulation in the « *SMIS Simulation Title* ».
3. Choose whether this is a 2D or 3D simulation with « *3D Off-On* ».
4. Set the number of different fluorophores (e.g. for multicolor experiments) you want to use in « *Number of fluorophores* ».
5. Load the virtual samples for each defined fluorophore via the « *Load virtual sample* » menu. You can create virtual samples with the menu “*Create virtual samples*”.
6. Set up labeling and photophysics for each fluorophore with the « *Setup Fluorophore* » menu. In this menu, you can load existing fluorophores, or create your own fluorophore by entering “*Define new fluorophore*”.
7. Decide whether this is a « *sptPALM* », « *qPALM* » or « *FRET* » experiment. The fluorophores and virtual samples must have been chosen accordingly. You will get warnings if this is not the case.
8. Defined the number of lasers to be used and set up each laser with the « *Setup Laser* » menu.
9. Define the final frame size of the detector (and output stack) by setting the « *Binning factor* »
10. Define the « *Number of frames* », « *Raster size* », « *Frametime* » and « *Time between frame time* ».
11. Define eventual « *Sample drift* ».
12. Decide whether this is a single or two-channel experiment with the « *Number of acquisition channels* » toggle.
13. Set up the microscope « *Objective and PSF* » parameters.
14. Set up the « *Emission Filters* *and Acquisition Channels options*», and eventually the parameters of the « *EMCCD camera* » (such as the EMCCD gain).
15. Define the « *Fluorescence background* ».
16. Choose the « *Output Directory* » and « *Stack File Name* ».
17. Save your simulation with the « *Save simulation parameters* » button. Saving can be repeated at any time during the process.
18. Finally launch the simulation with the « Launch simulation » button (single molecule mode) or « Launch ensemble simulation » (ensemble mode).
19. The output .tif stacks can be analyzed as real experimental data. The ground truth SMIS data can be analyzed with the « *Analyze Results* » tool.

# Defining default directories

Defaults directories can be set up in « *Default SMIS Directories* ».

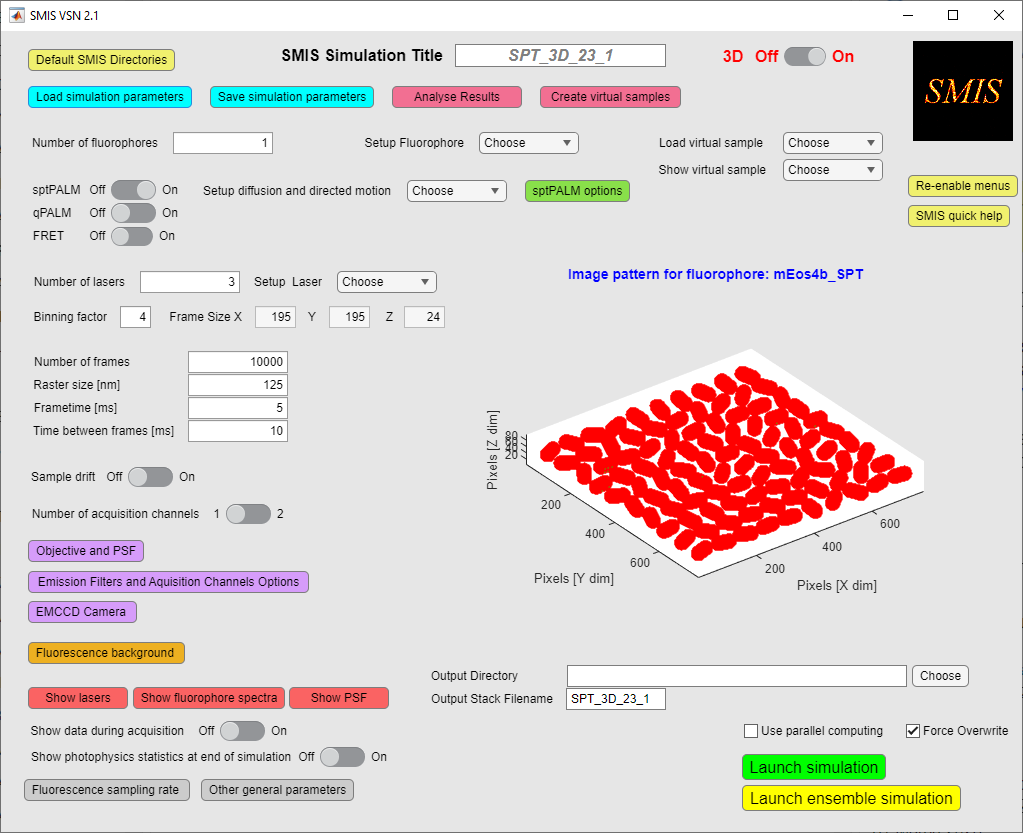


This is useful to directly access your *SMIS simulations library*, *SMIS simulation output directory*, *virtual sample library* and *fluorophore library*.

If you want to save the entered directories for future SMIS sessions, click on « *Save* ».

# Running simulations

## Choosing between 2D and 3D simulations



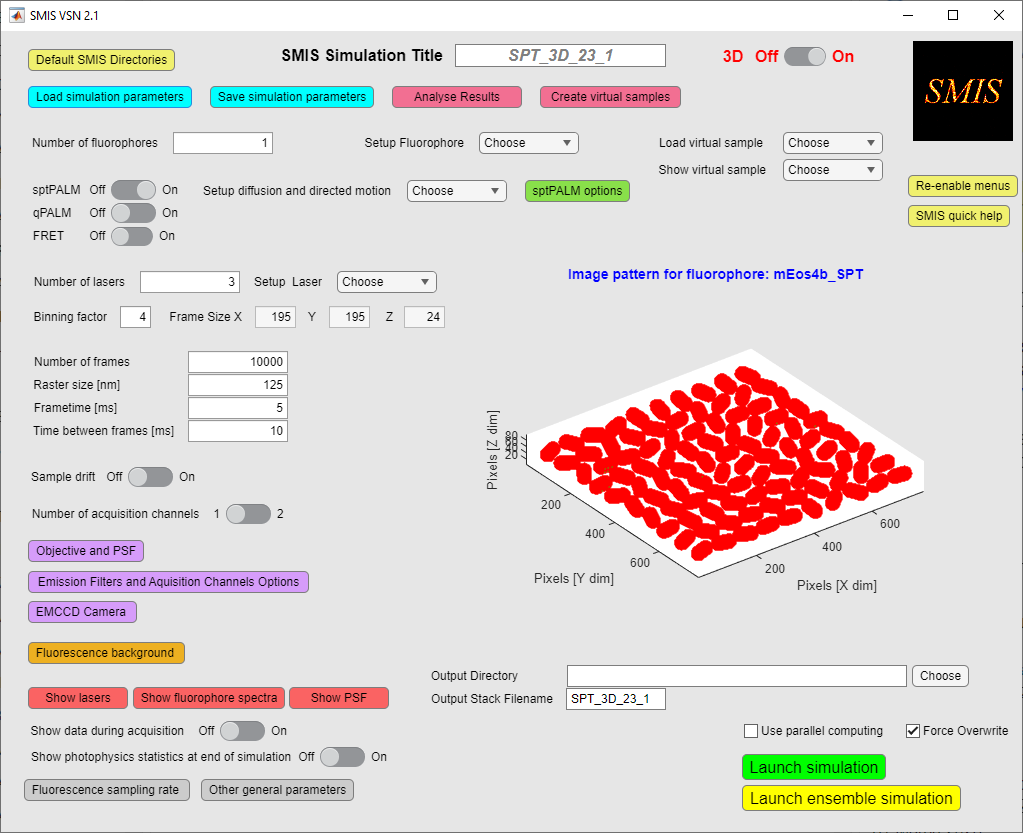
If you want to run a 2D simulation set 3D to « *Off* », and if you want to run a 3D simulation, set 3D to « *On* ».

For 2D simulations, virtual samples are 2D segmented images in a .tif format. Ground truth molecule positions will only be defined in X and Y.

For 3D simulations, virtual samples are 3D Matlab arrays. Ground truth molecule positions will be defined in X, Y and Z.

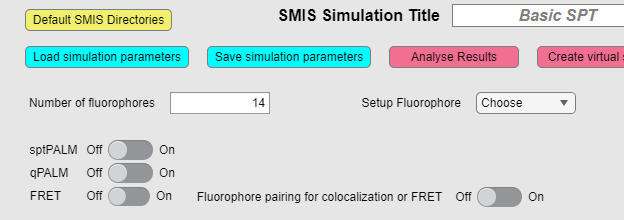
Toggling between 3D on and off will **reset the virtual samples** and some laser parameters.

## Choosing the number of fluorophores (or number of colors)



Choose the number of different fluorophores in the « *Number of fluorophores* » box. The number of fluorophores is not limited, so if you want to run a 10-color experiment, you can …

As soon as you use more than one fluorophore, you will get the option to pair fluorophores for colocalization studies.



# Choosing virtual samples

### General characteristics of virtual samples:

Virtual SMIS samples are *segmented* images or 3D arrays. Each Pixel (2D) or Voxel (3D) will thus have a value called a *pattern ID* in SMIS. All pixels or voxels with a particular value (or pattern ID) can be considered as a *labeling target* for a fluorophore.

Let’s take the example of the SMIS logo, a simple 2D virtual sample with only 2 IDs (0 and 255 in that particular case)

Pixel value = 0

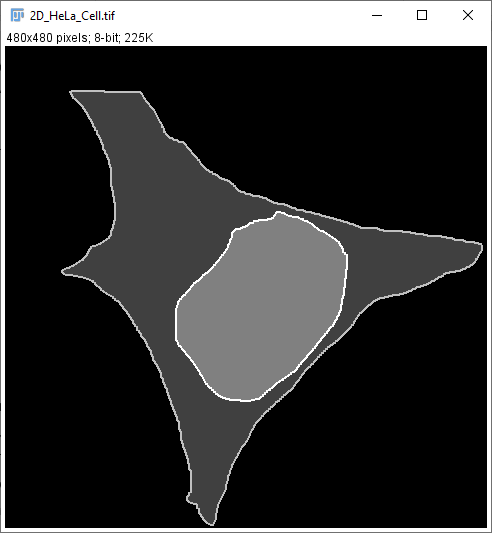


Pixel value = 255

We will see later in the « *Setup* *Fluorophore* » how to label the SMIS logo with the fluorophores.

In this particular case if we want to label the *SMIS* pattern, we will address the fluorophores to the pattern ID = 255. Instead, if we want to label the background, we will address the fluorophores to the pattern ID = 0. Maybe, if we want to simulate unspecific labeling, we will label 95% of the fluorophores to the pattern ID = 255, and the 5% left to the pattern ID = 0.

You may have virtual samples with multiple pattern IDs, for example the one below:



Pixel value = 3

Pixel value = 4

Pixel value = 0

Pixel value = 1

Pixel value = 2

In this particular case of a HeLa cell, you could address fluorophores to: the background (pattern ID = 0), the cytoplasm (pattern ID = 1), the nucleus (pattern ID = 2), the plasma membrane (pattern ID = 3) or the nuclear membrane (pattern ID = 4).

*As a consequence of this, loading a nonsegmented image with multiple pixel values into SMIS will result in a big mess. So don’t do that (except for qPALM samples, see below) !*

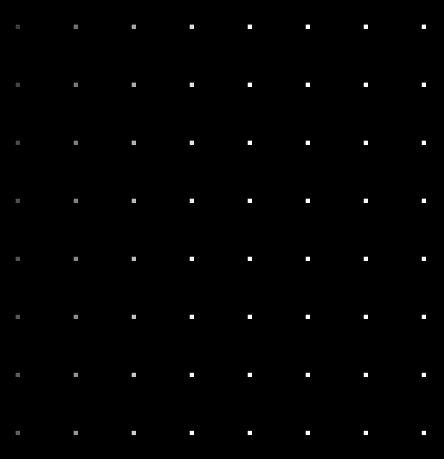
#### For 2D simulations

For 2D simulations, virtual samples are 2D segmented images in *.tif* format. You can define new SMIS virtual samples using for example *ImageJ, or Matlab.* A limited set of 2D virtual samples can also be created with the« *SMIS Create Virtual Sample tool* »*.*

#### For 3D simulations

For 3D simulations, virtual samples are 3D Matlab arrays. You need *Matlab* to define new 3D virtual samples (which is presently a limitation). A limited set of 3D virtual samples can be created with the« *SMIS Create Virtual Sample tool* »

#### For qPALM simulations

Virtual samples for quantitative PALM simulations are typically images displaying repeated features (e.g. cluster sites) and allow placing a well-defined number of fluorophores in each of these features (i.e. defining a well-defined stoichiometry). In that case, each individual feature should be given a different pixel value, such as in the example below:

Pixel value = 57

Pi value = 64

Pixel value = 8

Pixel value = 1

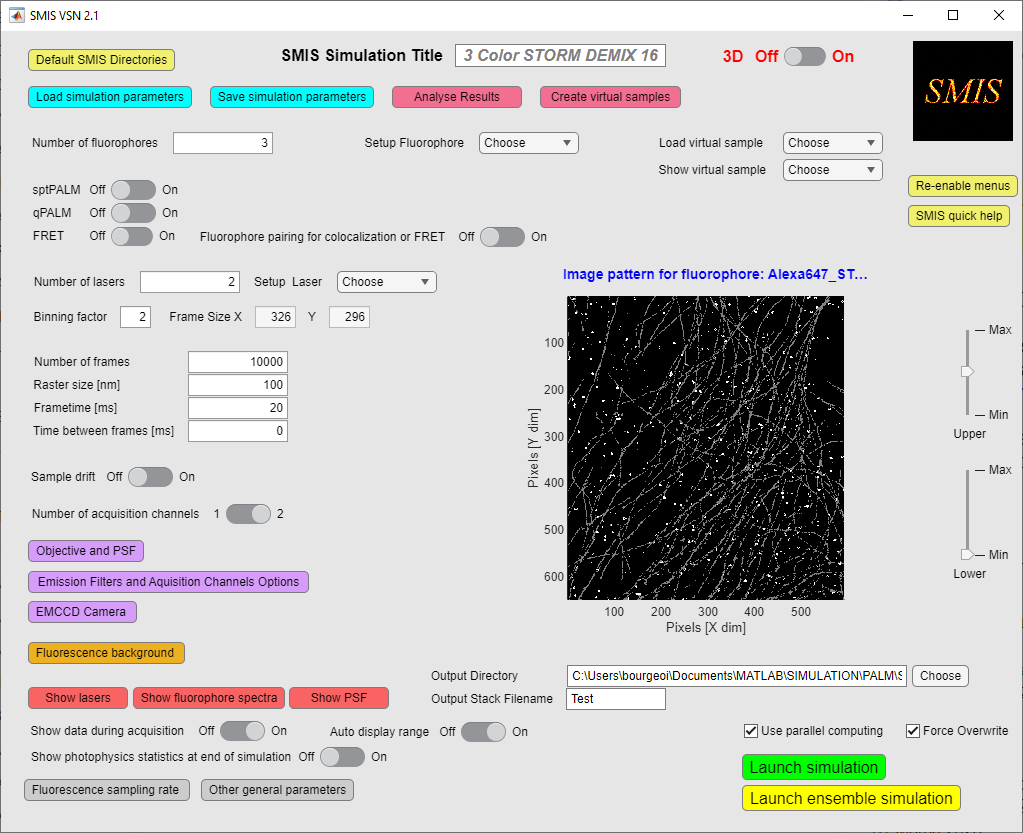
### Loading virtual samples:

Once you have defined the number of fluorophores, virtual samples can be loaded with « *Load virtual sample* »

# 

You can load a different virtual sample for each fluorophore in a multicolor experiment, or you can choose a single virtual sample that you will use for all the fluorophores. In case you choose different samples for different fluorophores, the virtual sample image sizes need to be the same. If a newly loaded virtual sample as a different size than previously loaded samples, SMIS will ask if you agree to reset those previous samples.

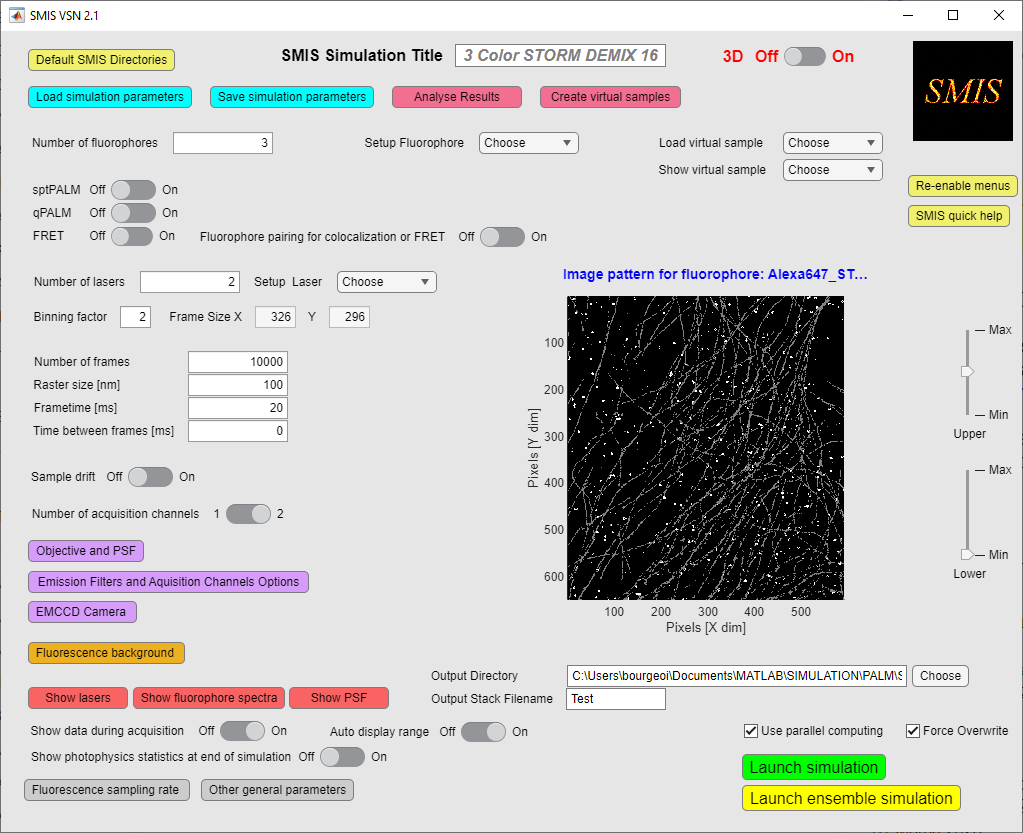
Once the samples are loaded, you can look at them with « *Show virtual sample* ».



You can adjust the contrast of the displayed virtual samples.

In principle, you can zoom in, zoom out, pan, or even rotate (3D patterns) the virtual samples, as for normal Matlab figures, by moving the mouse within the orange ellipse (above figure). However this is not very advisable, as this can be slow for 3D images, and it can mess up the SMIS interface.

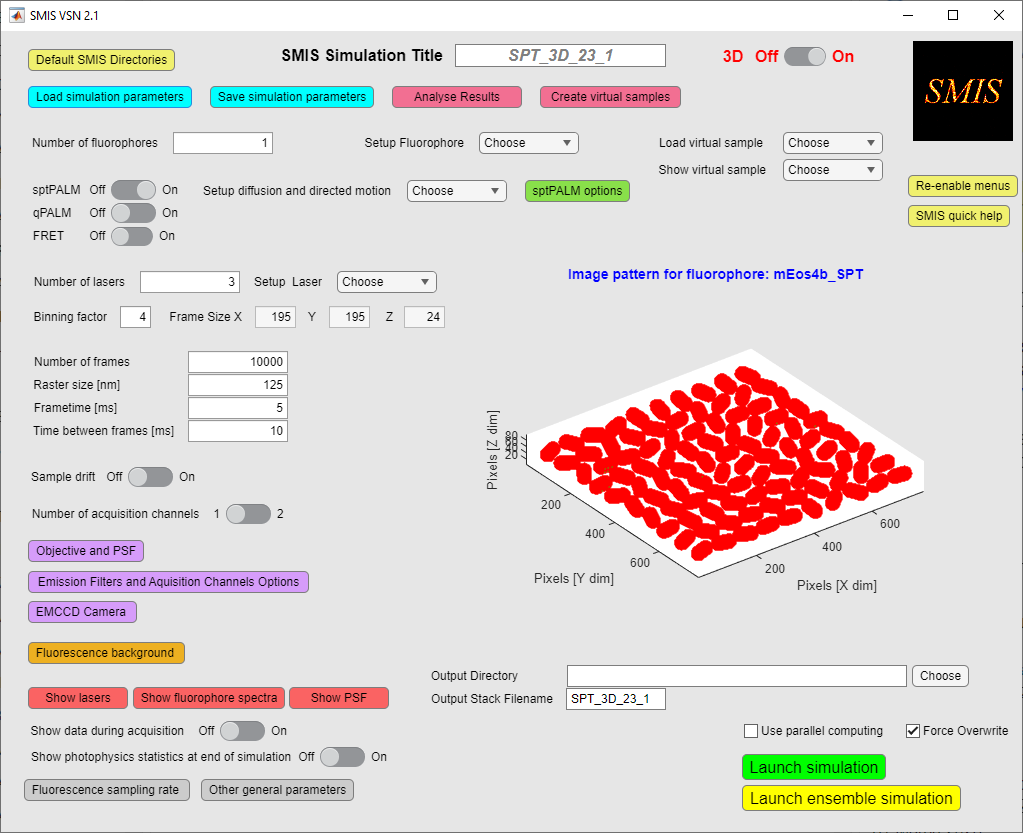
**The number of pixels in X and Y dimensions indicated on the displayed images correspond to the true size of the chosen virtual samples. The actual frame size (*Frame Size X and Y*) that will be “recorded” on the SMIS detector correspond to these dimensions divided by the « *Binning factor* », as shown on the image below.



It is thus important to realize that the “true size” of a pixel **(or voxel) in the virtual sample images is equal to *Raster size / Binning factor*. Thus, changing either values will affect the dimensions of objects that will be reconstructed upon processing the image stacks. You can get a feeling of this in the « *Create virtual samples* » tool.

# Setting up the fluorophores

Once the number of different fluorophores has been chosen, each fluorophore type should be set up:



The following new window will open:

**17**

**16**

**15**

**14**

**13**

**1231**

**111**

**10**

**9**

**6**

**5**

**3**

**2**

**8**

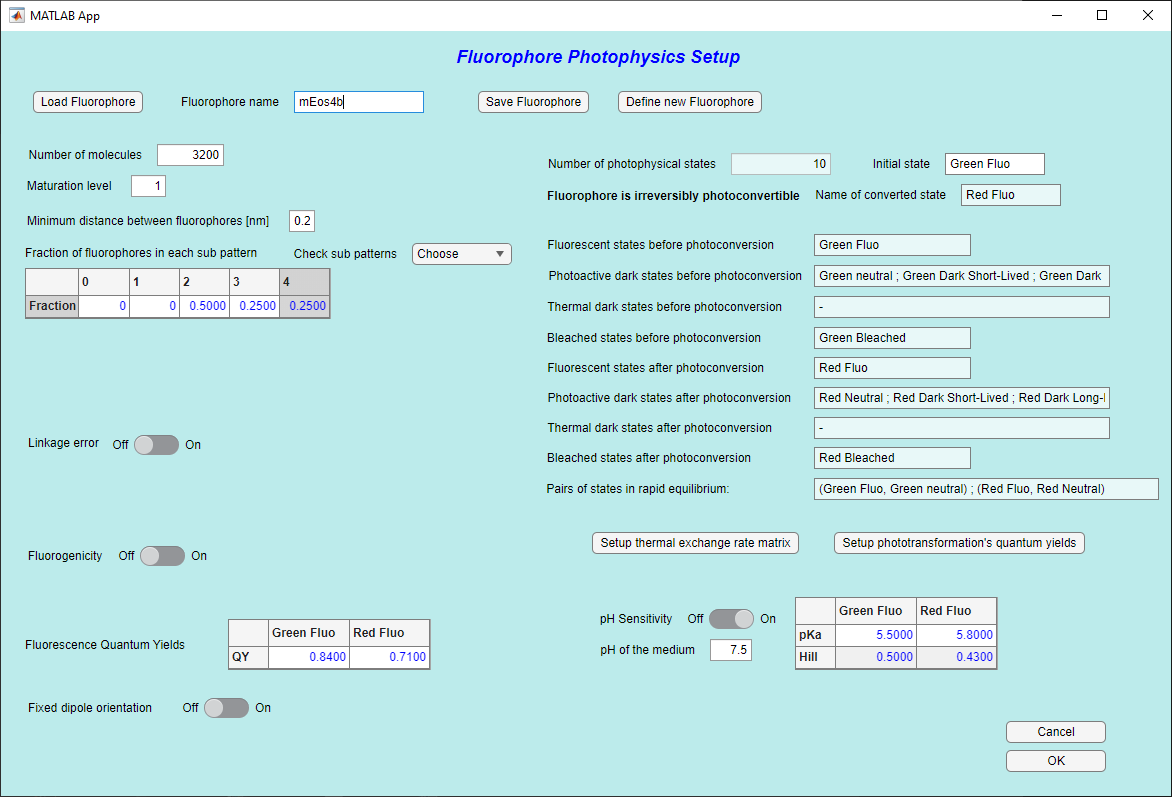
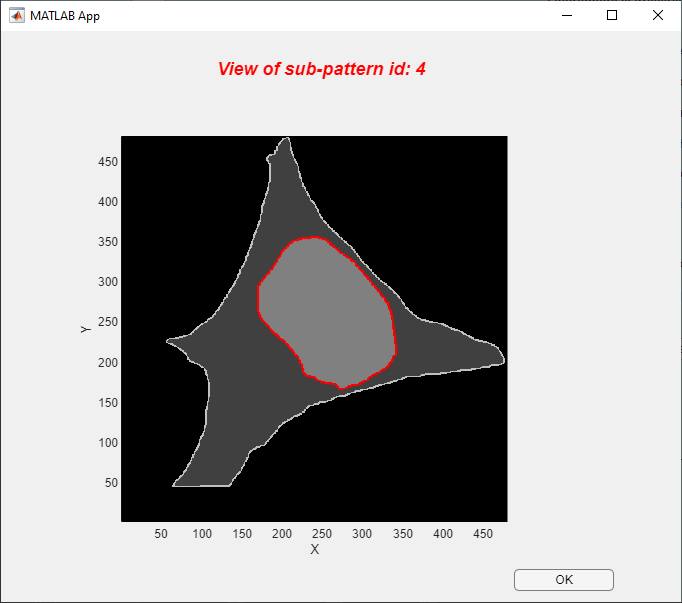
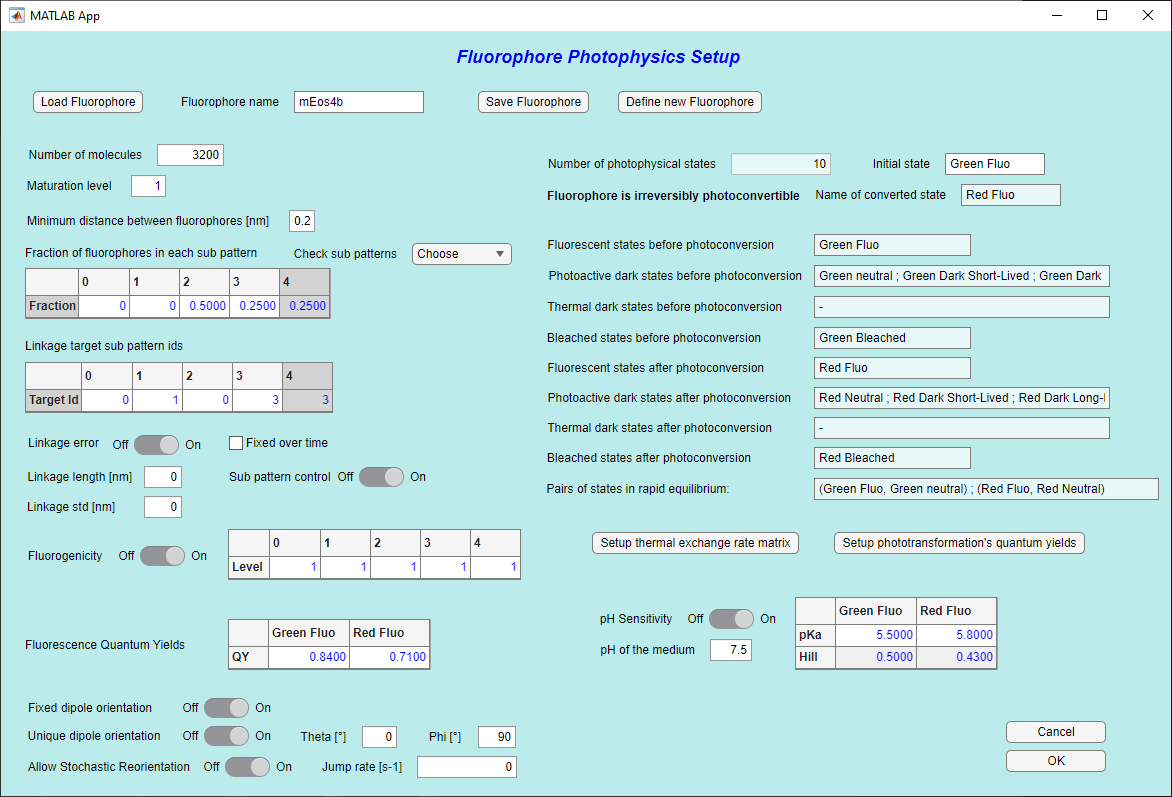
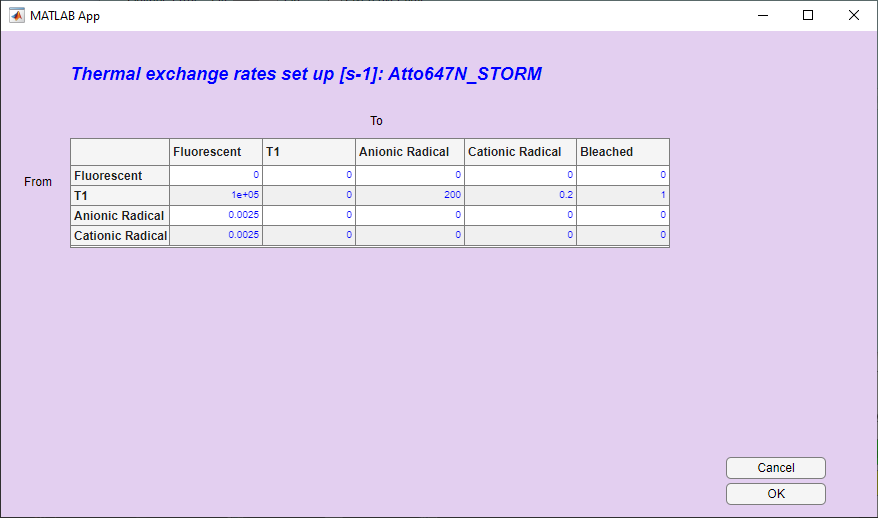
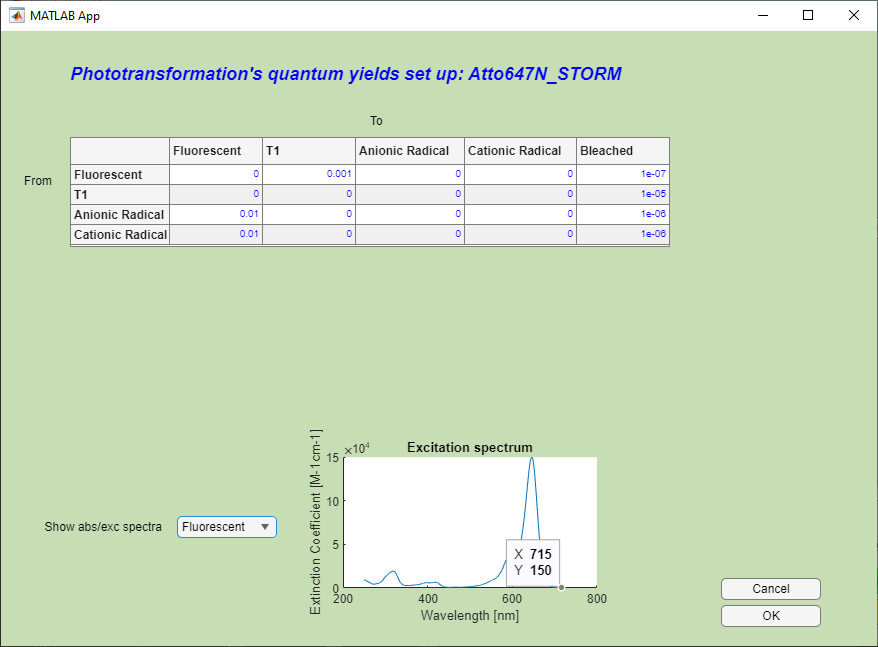
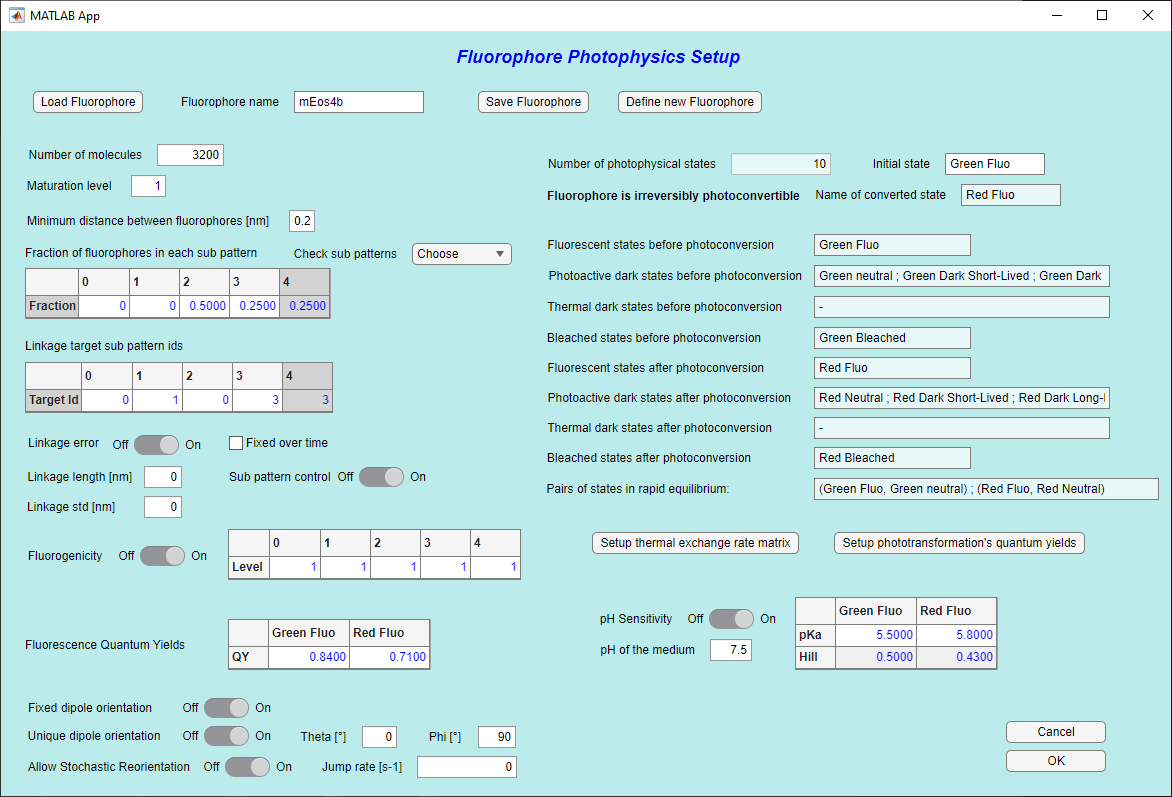
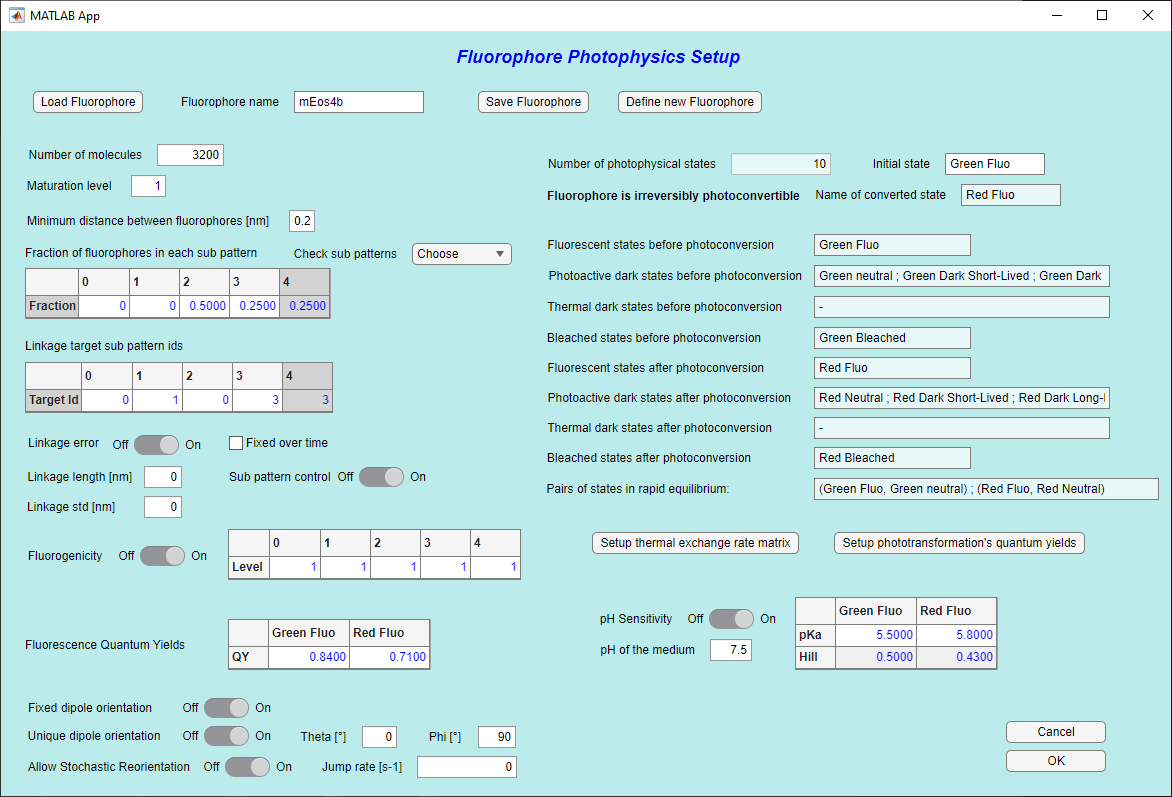
**7**

**4**

**1**

Description of photophysical states

Here is what you can do, preferably in order.

1. Load a fluorophore from the SMIS fluorophore database
2. Give it a name
3. If you cannot find the fluorophore you like in the database, you can define a new fluorophore (explained below)
4. Set the number of target molecules for that particular fluorophore
5. Distribute the fluorophore through the various features (“sub-patterns”) of the virtual sample, according to their *pattern IDs* (see section *Choosing virtual samples*).
6. You can check visually which pattern IDs correspond to which feature by using « *Check sub patterns* »: for example, on the image below you can see which feature corresponds to the pattern ID = 4: this is the nuclear membrane.
7. You can define a maturation level (for fluorescent proteins), which also corresponds to a labeling efficiency (for organic dyes). This corresponds to the fraction of target molecules that will be effectively labeled.
8. You can also set a minimum distance between target molecules. This will ensure that target molecules will not be too close to each other.
9. ****You can define a linkage error. This is the distance between the fluorophore and the actual target.   
   Be careful: a linkage error only makes sense if its value is greater than the digital resolution of the virtual sample. This is rarely the case, unless you use a highly resolved virtual sample. You can also artificially increase the binning value, or use a big linkage error, but this might not make a lot of sense. Linkage errors are not enabled in the case of FRET simulations.  
     
   As shown on the image above, if you opt for a linkage error, then you have to define the linkage length and standard deviation, and you have to choose whether this length remains constant (in length and orientation) over the whole data acquisition (typically the case for a fixed sample, but not for a live sample or in sptPALM experiments).  
   Also if « *Sub-pattern control* » is set to on, you can choose whether target molecules and attached fluorophores are in different sub-patterns of the virtual sample. In that case fill the "*linkage target sub pattern ids*" table. You might want to do this for example if the target is a membrane protein and the fluorophore should stand either inside the cytoplasm or in the external medium.   
   Do not use this option when the target is positioned inside a large sub pattern, because then the fluorophore would never reach the desired sub pattern. Thus, be careful if you use this option !
10. You can change interactively the rates of thermally activated transformations between photophysical states, although initial values are defined when the fluorophore is designed. For example, this is useful if you want to turn off a certain transformation.
11. Likewise, you can change interactively the quantum yields of light-activated transformations between photophysical states, although initial values are defined when the fluorophore is designed. Again, this is useful if you want to turn off a certain transformation, or increase it to look at the effect.  
      
      
    Of note, you can look at the spectra of all states in this menu.
12. Importantly, you can change the initial state of the fluorophore at the start of the simulation, by entering the *name* of the state in the text box. For example you could start a simulation with a photoconvertible fluorescent protein that is already fully convertible to the red state. If you want a fluorophore to start in a mixture of different states, you can duplicate the fluorophore and set each of the copies to a different initial starting state. Alternatively you can add an initial dummy state (using « *Define new fluorophore* ») that very rapidly equilibrate towards real photophysical states with different rates. This will allow you to populate these photophysical states differently, at the start of a simulation.
13. You can set your fluorophore to be fluorogenic. Here, fluorogenic means that the fluorophore will exhibit different quantum yield depending on the part of the virtual sample (sub-patterns) in which the fluorophore resides. This is typically useful for fluorophores that are only fluorescent when they e.g. are bound to a membrane.  
      
      
    In the table, enter the fractions of the default fluorescence quantum yield corresponding to the effective quantum yields in each sub pattern of the virtual sample.
14. You can modify the fluorescence quantum yields for each fluorescent states of the fluorophore.
15. You can enable pH sensitivity of the fluorophore. This is only possible for fluorophores that are initially defined with rapidly exchanging states between protonated (nonfluorescent) and ionic (fluorescent) states. See « *Define new fluorophore* ». If pH sensitivity is enabled, then you can modify the pKa and Hill coefficients for each fluorescent state of the fluorophore, and you can also modify the pH of the medium.
16. You can control the tumbling of fluorophores. Set to On if fluorophores have a fixed dipole orientation. Set to Off if fluorophores tumble more rapidly than the acquisition rate.  
      
    If a fixed dipole orientation is chosen, it can be random for all fluorophores, or a unique dipole orientation can be chosen for all fluorophores. Fluorophores can also reorient stochastically at a defined rate.
17. You can save your fluorophore at any moment. The properties that will be saved if you do so are the properties intrinsic to the fluorophore (photo-transformations, pKa’s, quantum yields …) but not the properties related to labeling (number of molecules, dipole orientation …).

&&&&

Copy setup

qPALM

define new fluorophore !

# Annex 1: Content of the *im\_par* Matlab structure

# Annex 2: Content of the *sm\_par* Matlab structure

# Annex 3: Content of the *sms* Matlab structure