



# SQUIRREL

super-resolution quantitative image rating  
and reporting of error locations

## NanoJ-SQUIRREL User Manual

Written by Siân Culley at Henriques Lab  
Quantitative Imaging and Nanobiophysics Group  
MRC Laboratory for Molecular Cell Biology  
University College London

Version 1.1 - July 2017

# Contents

1	What is NanoJ-SQUIRREL? . . . . .	2
2	Installing NanoJ-SQUIRREL . . . . .	2
3	Calculate Error Map, RSE and RSP . . . . .	3
3.1	Quick-start guide . . . . .	3
3.2	A brief note on image sizes . . . . .	4
3.3	Requirements for error map analysis . . . . .	4
3.4	Running the Error Map plugin . . . . .	5
3.5	Advanced Settings . . . . .	5
3.6	Output of Calculate Error Map, RSE and RSP . . . . .	6
3.7	Log messages . . . . .	9
4	FRC Mapping . . . . .	10
4.1	Requirements for FRC mapping . . . . .	10
4.2	Generating input super-resolution stacks for FRC mapping . . . . .	10
4.3	Running the FRC Map plugin . . . . .	12
4.4	Output of FRC Map . . . . .	12
5	Extract PSF . . . . .	13
5.1	Requirements for extracting PSF . . . . .	13
5.2	Running the Extract PSF plugin . . . . .	14
5.3	Output of Extract PSF . . . . .	14
6	Elastic image registration . . . . .	15
6.1	Requirements for elastic registration . . . . .	15
6.2	Running the Register SR Images to Reference - Elastic plugin . . . . .	15
6.3	Output of Register SR Images to Reference - Elastic . . . . .	17
7	Test data . . . . .	17

# 1 What is NanoJ-SQUIRREL?

NanoJ-SQUIRREL (Super-resolution **QU**antitative **I**mage **R**ating and **R**eporting of **E**rror **L**ocations) is a software package designed for assessing and mapping errors and artefacts within super-resolution images. This is achieved through quantitative comparison with a reference image of the same structure (typically a widefield, TIRF or confocal image). SQUIRREL produces quantitative maps of image quality and resolution as well as global image quality metrics.

## 2 Installing NanoJ-SQUIRREL

NanoJ-SQUIRREL is freely available as a plugin for the open-source image analysis software Fiji and ImageJ.

1. Open Fiji, go to the ‘Help’ menu and click ‘Update...’ (or ‘Update Fiji’ depending on your version of Fiji). Once the checksummer has run, the window shown below should appear (this window may contain items depending on the current status of your version of Fiji - this will not affect NanoJ installation).

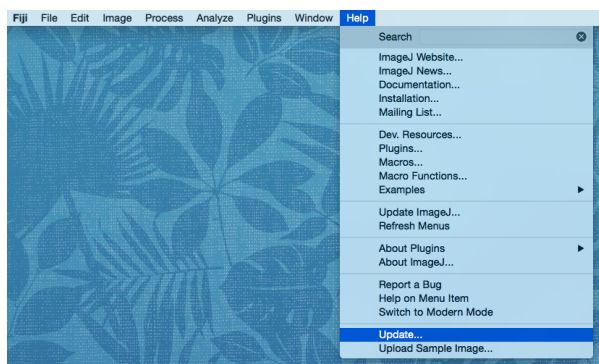


Figure 1: The Fiji ‘Help’ menu

2. Press ‘Manage update sites’ and tick the boxes next to:

Name	URL
NanoJ	<a href="http://sites.imagej.net/NanoJ/">http://sites.imagej.net/NanoJ/</a>
NanoJ-SQUIRREL	<a href="http://sites.imagej.net/NanoJ-SQUIRREL/">http://sites.imagej.net/NanoJ-SQUIRREL</a>

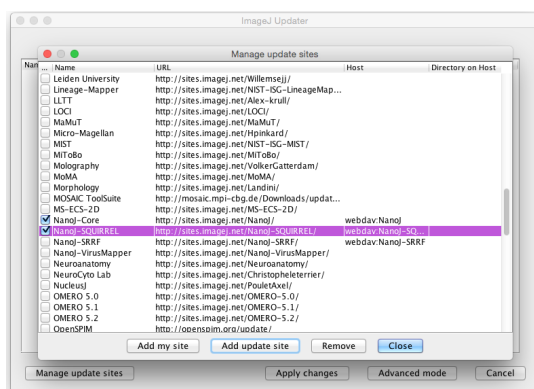


Figure 2: The ‘Manage update sites’ dialog box

3. Close the ‘Manage update sites’ window and apply changes when prompted.

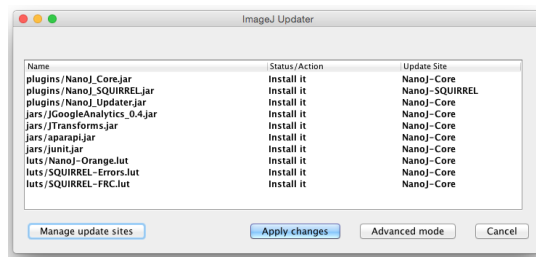


Figure 3: List of files which will be added to Fiji to run SQUIRREL analysis

4. Restart Fiji, and the new NanoJ and SQUIRREL menus should now appear under the Plugins menu.

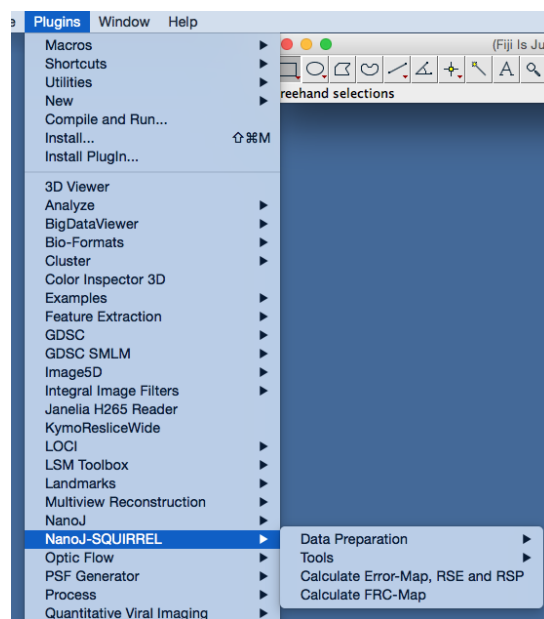


Figure 4: Plugins menu following NanoJ-SQUIRREL installation

### 3 Calculate Error Map, RSE and RSP

The first main analytical plugin in NanoJ-SQUIRREL is ‘Calculate Error Map, RSE and RSP’, and is found in the NanoJ-SQUIRREL sub-menu within the Fiji/ImageJ Plugins menu. Error mapping works by calculating a resolution scaling function (**RSF**) for each super-resolution image such that convolution with the **RSF** yields an equivalent widefield/confocal etc. image which can then be mathematically compared with the true **reference image**. This generates maps where local discrepancies between the super-resolution and reference images can be identified, and quality metrics for the super-resolution images (**RSE** and **RSP**).

#### 3.1 Quick-start guide

I recommend that you read the instructions and guidelines below before running SQUIRREL analysis; however if you are too eager to start seeing what’s wrong with your images then all you need to do is:

- Open your [reference image](#)
- Open your super-resolution image/[image stack](#)
- Navigate to Plugins>NanoJ-SQUIRREL>Calculate Error Map, RSE and RSP
- Select your images from the drop-down menus (select ‘- RSF unknown, estimate via optimisation -’ from the [RSF](#) estimate image drop-down menu)
- Click OK

And voilà, you are on your way!

### 3.2 A brief note on image sizes

NanoJ-SQUIRREL was originally programmed assuming that the super-resolution image is some form of magnified version of the reference image. For example, in many PALM and STORM algorithms the reconstructed image is rendered on a finer pixel scale (e.g. 20nm) than the original input image (e.g. 100nm). If this is the case then please do not enlarge your reference image as it will make computation very slow. Error Map calculation will still run if the images are the same magnification (e.g. both reference and super-resolution images have 20nm pixels) but the following message will appear in the Log:









NanoJ: Reference and SR images appear to be on same scale, second optimiser will be skipped.

Both the Log and the Fiji status bar will provide estimates of how long the analysis will take. Error map analysis is implemented on the [GPU](#) to speed up processing. However, this also means that your computer may struggle with very large images. The two approaches for handling large images are:

- Crop your reference image and super-resolution image to a smaller region
- Change the maximum magnification used by SQUIRREL (described further below)

### 3.3 Requirements for error map analysis

The plugin requires **two** images to run: a [reference image](#) and a super-resolution image. If an approximate resolution scaling function ([RSF](#)) for the imaging system is known, this can also be input (see Section 5). This works best if the reference image is on its **original raw scale** (e.g. pixel size unchanged from acquisition) - the software will deal with any differences in pixel size/magnification between the reference and super-resolution image(s). Currently error mapping only supports **single-frame** reference images; the table below shows the permissible combinations of frame numbers for input reference (‘Ref.’), super-resolution (‘SR’) and RSF image stacks.

# Ref. frames	# SR frames	# RSF frames	Result
1	1	0	
1	1	1	
1	1	$n_{\text{RSF}} > 1$	
1	$n_{\text{SR}} > 1$	0	
1	$n_{\text{SR}} > 1$	1	
1	$n_{\text{SR}} > 1$	$n_{\text{SR}}$	
1	$n_{\text{SR}} > 1$	$n_{\text{RSF}} \neq n_{\text{SR}} \wedge n_{\text{RSF}} > 1$	
$n_{\text{ref}} > 1$	$n_{\text{SR}} \in \mathbb{N}$	$n_{\text{RSF}} \in \mathbb{N}$	

### 3.4 Running the Error Map plugin

From the drop-down menus in the ‘Calculate Error Map, RSE and RSP’ dialog box, select your [reference image](#), super-resolution image (stack) and, if applicable, [RSF](#) estimate image (stack).

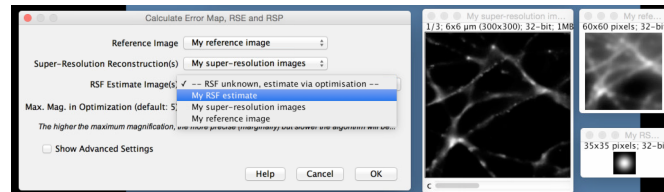


Figure 5: The ‘Calculate Error Map, RSE and RSP’ dialog box where an RSF estimate image is provided

If you do not have an RSF estimate then select the option ‘RSF unknown, estimate via optimisation’ and the plugin will generate an RSF for you.

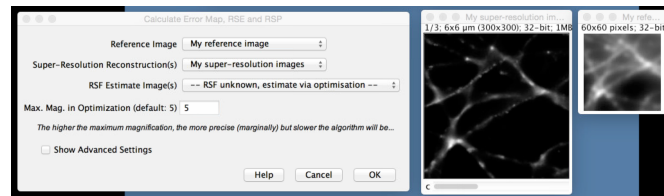


Figure 6: The ‘Calculate Error Map, RSE and RSP’ dialog box where there is no RSF estimate image

If you are intending to run analysis on large images then the maximum magnification value should be changed to reduce computation time and improve overall stability. For example, if the input super-resolution image is 10x magnified compared to the reference image, then a maximum magnification of 5 will result in a downscaling of the super-resolution image to 5x the magnification of the reference. This is also an appropriate strategy if the reference image is large, even if the super-resolution image magnification is  $\leq 5$ ; we have seen that even taking the maximum magnification down to 1 only minimally impacts upon the error maps and RSE/RSP metrics.

Press OK, and analysis will commence!

### 3.5 Advanced Settings

Checking the ‘Show Advanced Settings’ box will open the following dialog:

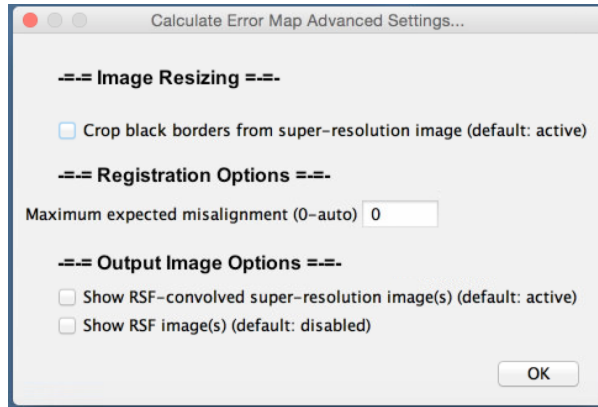


Figure 7: The dialog box for Error Map Advanced Settings with default values

<b>Crop black borders from super-resolution image</b>	This box should be checked if one or more of the frames in the super-resolution image stack has borders consisting of zeros or NaNs. If this box is checked, both the reference image and super-resolution image stack will be cropped to remove edges corresponding to the largest super-resolution borders identified. If cropping occurs, the cropped reference and super-resolution images will appear. If no borders are identified in any of the super-resolution frames no cropping will be performed.
<b>Maximum expected misalignment</b>	The value in this field will limit the maximum distance (in units of super-resolution pixels) over which to perform registration of the reference and super-resolution images. Setting this value to 0 will put no limit of the maximum distance for registration.
<b>Show RSF-convolved super-resolution image(s)</b>	Displays the result of registration, intensity-scaling and RSF convolution on the super-resolution image stack. This is the stack which is compared to the reference image for producing error maps.
<b>Show intensity-scaled super-resolution images</b>	Displays the result of applying $\alpha$ and $\beta$ intensity normalisation on the super-resolution image stack.
<b>Show RSF image(s)</b>	Displays the derived RSF(s) in a stack with the same number of frames as the super-resolution image stack.

### 3.6 Output of Calculate Error Map, RSE and RSP

The outputs of this plugin will be demonstrated using a [reference image](#) (left) and a super-resolution [image stack](#) containing three frames (centre). The montage shows the three super-resolution frames comprising the stack and is not used as an input for the plugin; it is just for illustrative purposes.

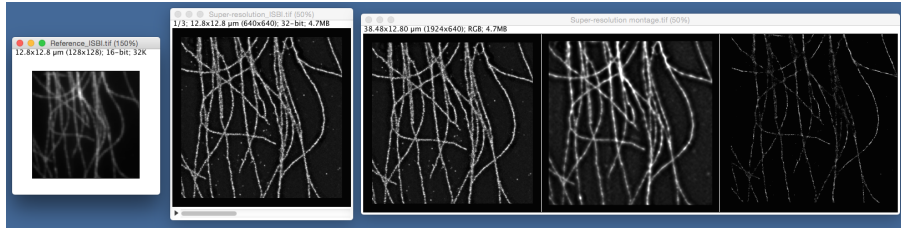


Figure 8: Test data from the ISBI localisation microscopy challenge for producing example outputs. Left: Reference image. Centre: Super-resolution image stack. Right: Montage showing the images in the super-resolution stack

The default outputs of this plugin are:

- An error map image stack containing the same number of frames as the input super-resolution image.

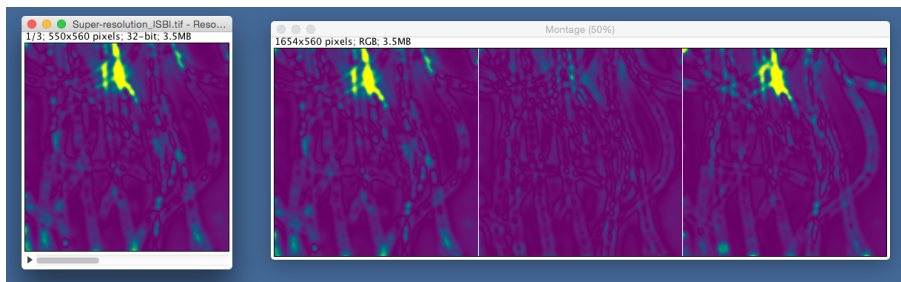


Figure 9: Left: Error map stack. Right: Montage showing the three error maps side-by-side

The SQUIRREL-Errors look-up table is automatically applied to the error map images; purple regions have low error (i.e. good agreement between super-resolution and reference) and yellow regions have high error (i.e. poor agreement between super-resolution and reference).

- A results table listing the [RSE](#) and [RSP](#) values for each frame in the super-resolution image stack.

RSP and RSE values.csv			
	Frame	RSP (Resolution Scaled Pearson-Correlation)	RSE (Resolution Scaled Error)
1	1	0.898	58.345
2	2	0.970	32.049
3	3	0.904	57.025

Figure 10: Results table with RSP and RSE values for each super-resolution frame

This table can be saved as a .csv file for later reference.

- Cropped reference and super-resolution images if border removal and/or registration was performed.



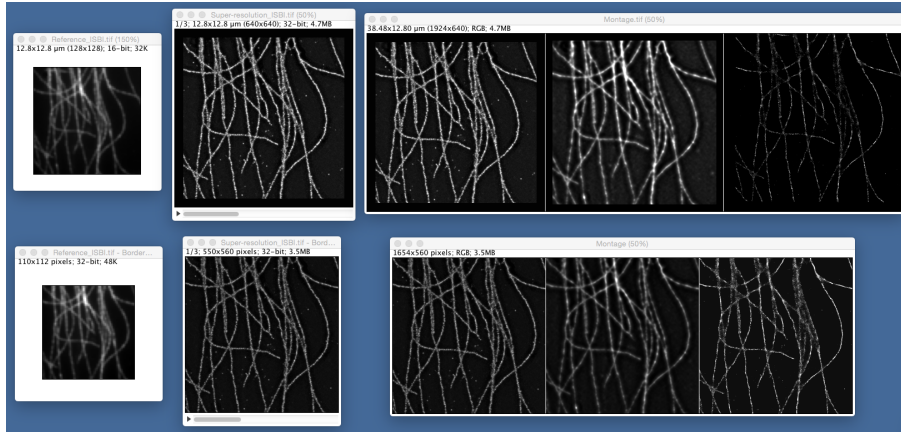


Figure 11: Top row: Original input reference image, original input super-resolution image stack and montage of original super-resolution image frames. Bottom row: Reference image and super-resolution image stack after border cropping and registration (plus intensity scaling in the case of the super-resolution image stack); montage of super-resolution images.

The top row of images shows the original input data and representative super-resolution montage, and the bottom row shows the reference and super-resolution images (plus montage) following cropping to remove borders and registration. The super-resolution images will also be automatically intensity scaled.

- An image stack where the intensity-scaled super-resolution frames have been convolved with their respective [RSFs](#).

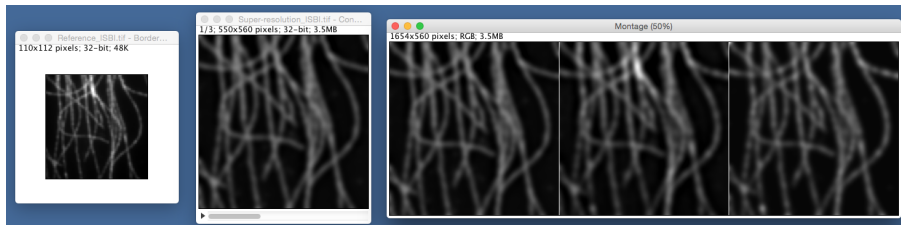


Figure 12: Left: Cropped reference image. Centre: Cropped, intensity-scaled and RSF-convolved super-resolution image stack. Right: Montage of convolved super-resolution image stack. All images are displayed on the same intensity scale.

The above figure shows the cropped reference image, the convolved super-resolution image stack and a representative montage.

Additional optional outputs (from Advanced Settings) are:

- Intensity-rescaled super-resolution image stack. This will display the super-resolution stack where the intensity of each individual frame has been linearly rescaled such that after convolution with the RSF the intensities match that of the reference image.
- Optimised RSF stack. These are the RSFs used for error mapping each super-resolution frame and will be produced irrespective of whether an RSF estimate was initially provided. The RSF stack will contain an RSF for each super-resolution frame and so will contain the same number of frames as the super-resolution image (shown in the montage below, next to the output). The scale of the RSF stack is the same as the super-resolution scale.

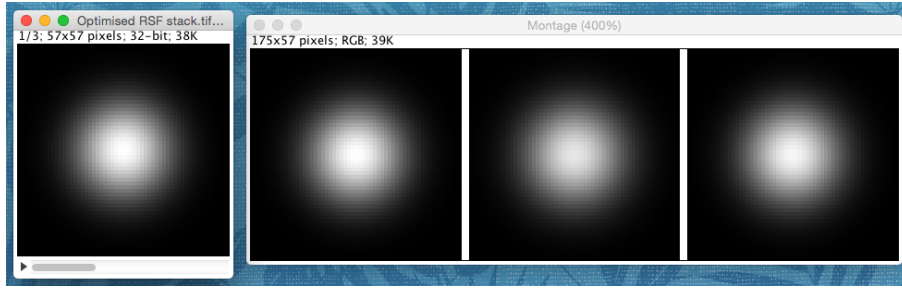


Figure 13: Left: Stack of optimised Gaussian RSF images. Right: Montage showing the three RSF images.

### 3.7 Log messages

The ImageJ Log provides information on any cropping which has been performed on the images, parameter guesses from [meta-optimisation](#), progress of optimisers and final parameter values. An example output of a portion of the Log is shown below:

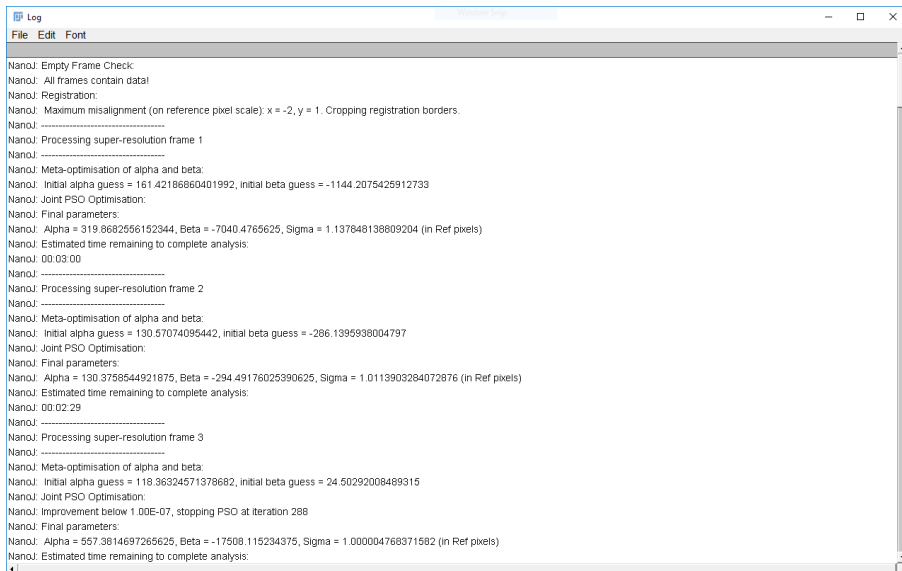


Figure 14: Example of the error map Log window output.

Possible messages displayed for each line of the Log are:

Field	Meaning	Possible messages
Empty Check	Frame	Prior to any further analysis, SQUIR-REL checks the super-resolution image stack to determine whether there are any frames where the intensity is homogeneous throughout (e.g. a frame full of zeros). Any homogeneous frames are removed from the super-resolution image stack before continuing.
		Frames purged: $[f_1, f_2, \dots]$

Border Control	If the ‘crop borders’ option is enabled in the Error Map Advanced Settings, this field will return information on the sizes of the largest borders found.	No borders to crop!  Rounded border widths (reference scale): Top = $n_1$ , Right = $n_2$ , Bottom = $n_3$ , Left = $n_4$
Registration	The largest x and y shifts to register the super-resolution image stack to the reference image are listed here, if applicable.	Images are already registered  Maximum misalignment (on reference pixel scale): x = $n_1$ , y = $n_2$ . Cropping registration borders.
Meta-optimisation of alpha and beta	Returns the initial guessed values of alpha and beta generated by a linear fit to the plot of reference image pixel values against super-resolution frame pixel values.	Initial alpha guess = $n_\alpha$ , initial beta guess = $n_\beta$
Joint <a href="#">PSO</a> Optimisation	Main PSO loop for <a href="#">RSF</a> optimisation is currently being performed. Values of $\alpha$ , $\beta$ and $\sigma$ , along with the estimated time remaining for optimisation, are displayed in the Fiji status bar. If optimisation does not improve beyond a set tolerance ( $1e-7$ ), this will be displayed in the log.	Improvement below $1e-7$ , stopping PSO at iteration $i$
Final parameters	Optimal alpha, beta and sigma parameters found for this super-resolution frame.	Alpha = $n_\alpha$ , Beta = $n_\beta$ , Sigma = $n_\sigma$ (in Ref pixels)

## 4 FRC Mapping

The other main analytical plugin in the NanoJ-SQUIRREL package is ‘Calculate FRC Map’, also found in the NanoJ-SQUIRREL menu in the Fiji/ImageJ Plugins menu.

### 4.1 Requirements for FRC mapping

The [FRC](#) Map plugin requires an [image stack](#) containing **two statistically independent super-resolution images of the same structure** for each super-resolution image. Therefore for FRC mapping of a single super-resolution image, an image stack containing two frames is required. FRC mapping supports multiple super-resolution image inputs; to perform FRC mapping of a stack of super-resolution images, the image stack should be organised as two frames for the first super-resolution image, then two frames for the second super-resolution image etc.

### 4.2 Generating input super-resolution stacks for FRC mapping

For datasets where the final super-resolution image is generated through computational analysis of multiple sequentially acquired raw images (e.g. PALM, STORM, SRRF) there are two

methods for generating an input [FRC](#) stack.

1. Partition the raw data prior to analysis. NanoJ-SQUIRREL includes a plugin which splits a raw image sequence into two separate image sequences; one containing the odd-numbered frames and the other containing the even-numbered frames.

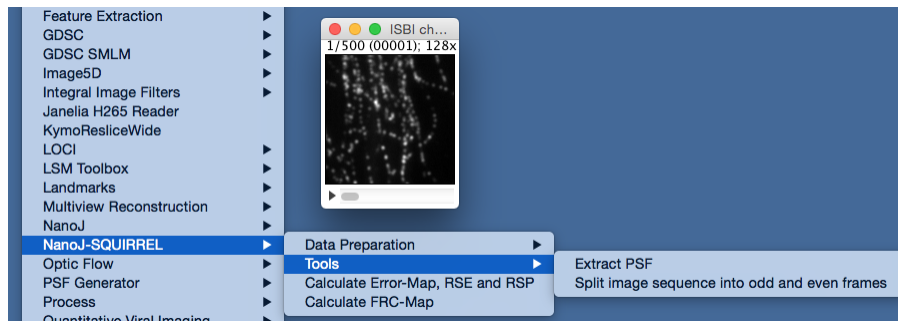


Figure 15: Location of image sequence splitter within NanoJ-SQUIRREL

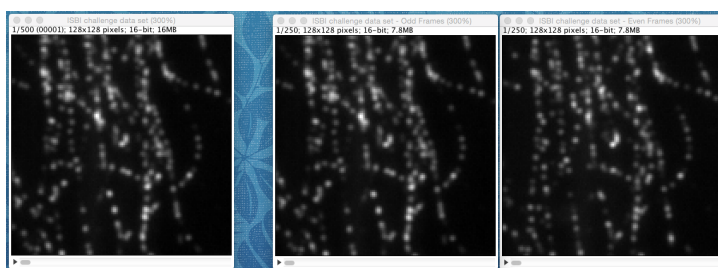


Figure 16: Left: Original 500-frame raw image sequence. Right: 250-frame image sequence composed of odd frames only and 250-frame image sequence composed of even frames only.

These two new image sequences can then be analysed using the same algorithm, parameters etc. as would be used to generate a super-resolution image from the original dataset to generate two super-resolution images, each containing half of the information within the raw image sequence. These two super-resolution images can then be concatenated to make a two-frame [image stack](#) for input into FRC.

2. Partition particle localisations after analysis. If a list of particle localisations was saved during generation of the original super-resolution image, these localisations can be filtered and used to make two super-resolution images from the original set of localisations. For example, if a super-resolution image is generated in ThunderSTORM, a super-resolution image consisting of localisations from odd frames only can be made by setting the filter to `frame%2=1` and likewise even frames only by setting the filter to `frame%2=0`. Once the two super-resolution images are split out from the original data, they can then be concatenated to make a two-frame image stack for input into FRC.

For other super-resolution methods, it is more difficult to satisfy the statistical independence condition of FRC analysis. For SIM and STED, you can try acquiring two images immediately after one another; however the FRC results will be affected by bleaching between the two acquisitions. If STED data has been acquired using TCSPC, one option (although not tested by our group) would be to split the image into two separate images according to photon arrival times.

### 4.3 Running the FRC Map plugin

The FRC Map plugin requires a split super-resolution [image stack](#) to run (as described above); no [reference image](#) or [RSF](#) is required for this plugin. With this image selected, run ‘Calculate FRC Map’ from the NanoJ-SQUIRREL menu.

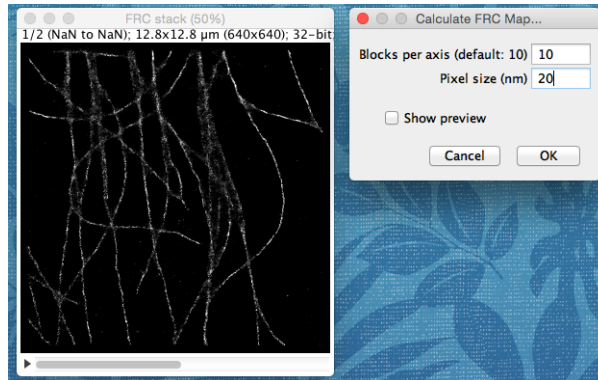


Figure 17: Input stack consisting of two super-resolution images from the same structure and Calculate FRC Map dialog

FRC mapping does not occur on a pixel-wise scale, rather the image is split into equally-sized spatial [blocks](#) over which FRC analysis is run. The ‘Blocks per axis’ field in the dialog allows you to specify the number of spatial blocks that the image is split into for analysis. For example, 10 blocks per axis will split a square image into 100 blocks and perform FRC analysis on each of these. The pixel size of the super-resolution image should be input into the ‘Pixel size (nm)’ field for correct calibration of the FRC results.

### 4.4 Output of FRC Map

The two outputs of the FRC Map plugin are a spatial map of the FRC across the image and a results table detailing the statistics of the FRC analysis.

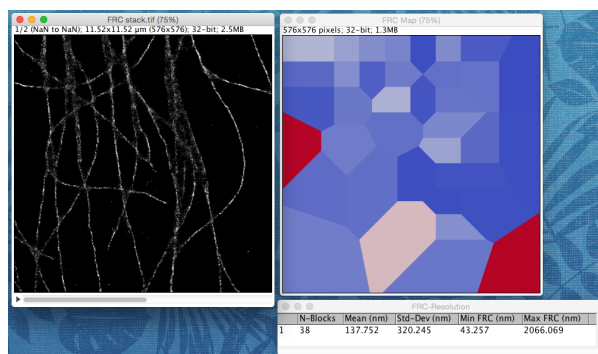


Figure 18: Left: Input super-resolution stack for FRC analysis. Right: Output FRC map of super-resolution image and table of results.

- **FRC Map** The FRC map is displayed using a red-white-blue colour map (SQUIRREL-FRC.lut) where red indicates higher FRC values (lower resolution) and blue indicates lower FRC values (higher resolution). There may be [blocks](#) within the map that are not square; this occurs when there are insufficient correlations between the two images at this

location to calculate an FRC value. To account for this, FRC values within these blocks are tessellated from neighbouring blocks.

- **FRC results table** The FRC results table lists:
  - N-blocks: the number of blocks in the FRC map following any required tessellation to compensate for blocks where an FRC value could not be calculated directly.
  - Mean (nm): the mean FRC resolution of all blocks
  - Std-Dev (nm): the standard deviation of the FRC resolutions of the blocks
  - Min FRC (nm): the resolution of the block which has the lowest FRC value (highest resolution)
  - Max FRC (nm): the resolution of the block which has the highest FRC value (lowest resolution)

## 5 Extract PSF

NanoJ-SQUIRREL includes a plugin which extracts an average **PSF** from a sparse blinking dataset. The ‘Extract PSF’ plugin is found under the Tools sub-menu of NanoJ-SQUIRREL.

### 5.1 Requirements for extracting PSF

Extract PSF runs on a single frame containing spatially distinct fluorophore appearances, as is typical for a PALM or STORM dataset. The plugin can either be run on a raw **image stack**, where **PSFs** for the active frame will be extracted, or in the case of very sparse data it is often useful to create a montage of several raw frames and run the plugin on this montage.

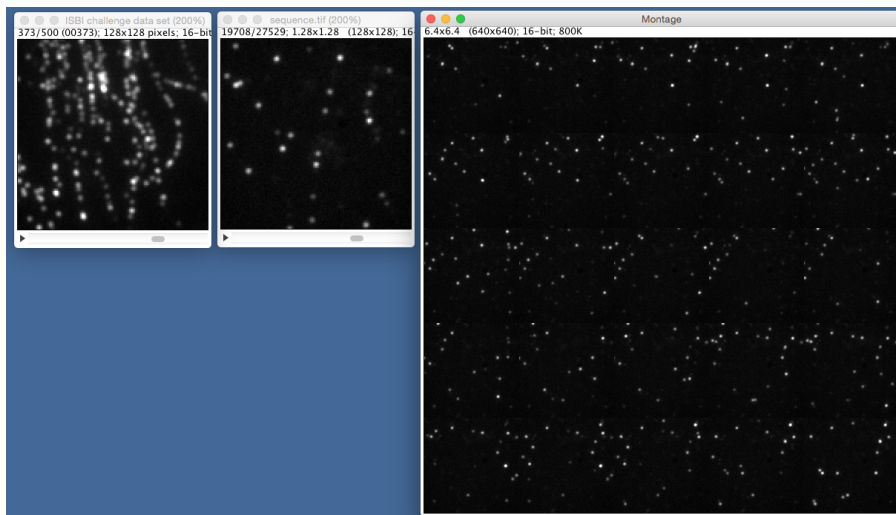


Figure 19: Left: Raw image sequence from which many PSFs can be extracted from a single frame. Centre: Raw image sequence where fluorophores are very sparse and each frame only contains a low number of molecules. Right: A montage of 25 frames of the sparse data providing many more PSFs for extraction than a single frame.

## 5.2 Running the Extract PSF plugin

With the image stack or montage from which a [PSF](#) is to be extracted selected, run the plugin from the NanoJ-SQUIRREL menu.

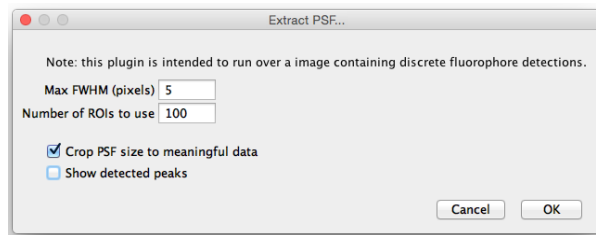


Figure 20: The ‘Extract PSF’ dialog box

The options available in the dialog box are:

- **Max FWHM (pixels)** This determines the size of the ROIs taken around each detected peak and should be slightly larger than the expected width of the PSF, in units of reference dataset pixels.
- **Number of ROIs to use** The number of ROIs extracted from the image for creating an average PSF image. More ROIs will give better averaging, but will also increase the processing time.
- **Crop PSF size to meaningful data** When this option is selected the output PSF will be cropped to such that border regions with intensities less than 1% of the maximum PSF intensity are discarded. If this option is not selected the size of the output PSF image will be  $(2 \times \text{MaxFWHM} + 1) \times (2 \times \text{MaxFWHM} + 1)$  pixels.
- **Show detected peaks** If this option is checked an additional image stack will appear after analysis is complete showing all of the extracted PSF ROIs used to create the average PSF aligned on top of each other.

If you are running Extract PSF on an [image stack](#) rather than a single frame/montage of several frames, changing the active frame in the image stack while the dialog box is open will update Extract PSF automatically to detect peaks in the new active frame.

## 5.3 Output of Extract PSF

The figure below shows an example input image and its associated outputs after running Extract PSF.

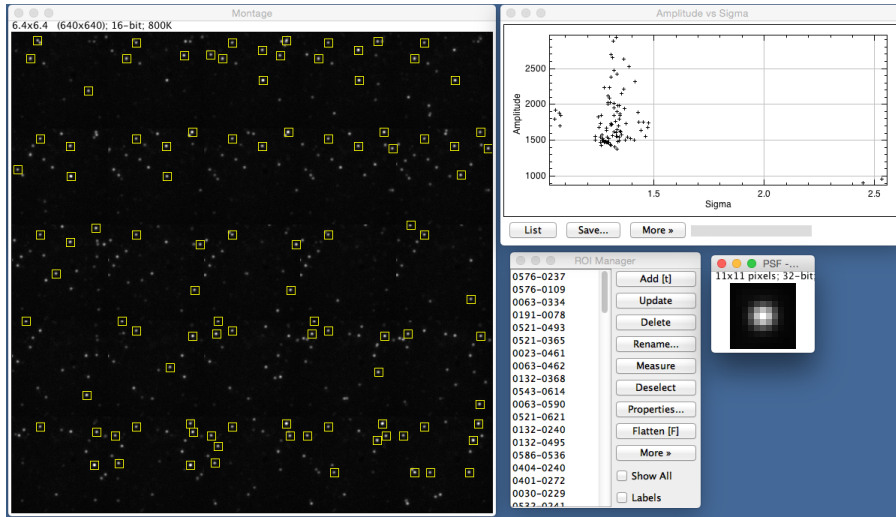


Figure 21: Input montage image and output plot, ROI list and PSF image

- **Average extracted PSF image** This image is normalised such that the total image intensity summates to 1 and is displayed on the same scale as the input image.
- **Amplitude vs sigma** During the Extract PSF computation, Gaussian fitting is performed on each ROI and the centres of these Gaussian fits used for aligning the ROIs onto each other. This plot displays the amplitude and sigma of the Gaussian fits to every ROI to give an idea of the range of apparent PSF sizes visible in the image.
- **ROI Manager** Contains all the ROIs used during analysis, displayed as yellow boxes on the input image.

## 6 Elastic image registration

In some instances a ‘rigid’ linear translation is insufficient for aligning images onto each other. This can occur when, for example, images of the same structure are acquired on two different microscopes with different aberration fields, causing local differences in misalignment. The Register SR Images to Reference - Elastic plugin is capable of correcting for these misalignments on a local scale.

### 6.1 Requirements for elastic registration

The Elastic Image Registration plugin requires a [reference image](#) and a super-resolution image or [image stack](#). Each frame in the super-resolution image stack will be locally realigned in order to best match the provided reference image.

### 6.2 Running the Register SR Images to Reference - Elastic plugin

With the reference and super-resolution images open, go to the NanoJ-SQUIRREL menu and select ‘Register SR Images to Reference - Elastic’ from the ‘Data Preparation’ sub-menu. The options in the dialog box are:

- **Reference image** Image to register on to.
- **Super-resolution images** Super-resolution images to be registered to reference.



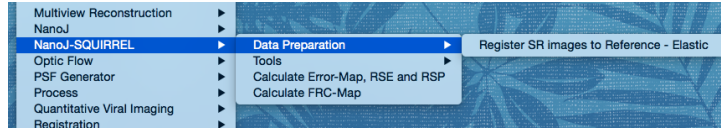


Figure 22: Location of elastic registration plugin within NanoJ-SQUIRREL

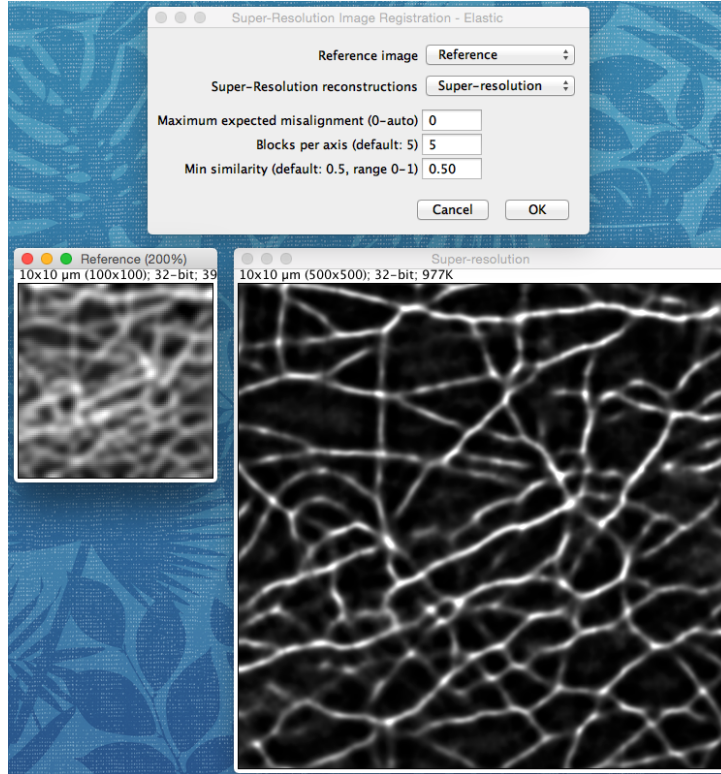


Figure 23: Elastic Registration dialog box and examples of two input images

- **Maximum expected realignment** The value in this field will limit the maximum distance (in units of super-resolution pixels) over which to perform local registration of the super-resolution images on to the reference. Setting this value to 0 will put no limit of the maximum distance for registration.
- **Blocks per axis** To perform local calculations of misalignment between reference and super-resolution images, the images are split into spatial **blocks** and the normalized cross-correlation between each reference and super-resolution block is calculated. This is then interpolated across the whole image so that each pixel has a registration vector attached to it. The blocks per axis determines the area over which this local registration calculation is performed. A higher number of blocks will give more detailed information on local misalignments for subsequent interpolation, but will be more sensitive to noise influences. Alternatively, a lower number of blocks will provide coarser information for the interpolation but will be less sensitive to noise.
- **Min similarity** The block-wise cross-correlation will give a similarity score from 0-1 for the best performing registration of each block analysed (where 1 is a perfect match). If the similarity of score of a block is below the threshold set in this field then the registration vector from this block will be discounted and instead the misalignment in this region of

the image will be estimated by interpolating from neighbouring blocks.

### 6.3 Output of Register SR Images to Reference - Elastic

There are three output images from the elastic registration plugin.

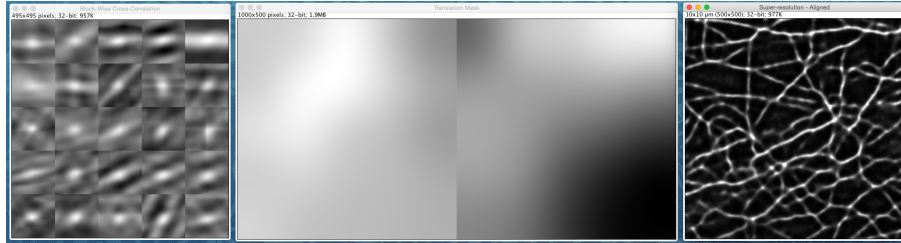


Figure 24: Output images from Elastic Registration

- **Blockwise cross-correlation** The normalized cross correlation maps produced for each block of the selected super-resolution frame against the reference.
- **Translation mask** The left half of the image is the x-translation applied to each pixel, and the right half of the image is the y-translation applied to each pixel.
- **{Image name} aligned** The input super-resolution image stack following pixel-wise translation as depicted in the translation mask image.

Note that here this has only been shown for a single super-resolution frame; this plugin also works for a super-resolution stack and will output the above images in stacks, one for each frame of the super-resolution image. Each super-resolution frame will be aligned to the original reference frame. The output aligned super-resolution stack can then be used as, for example, an input into the Error Mapping plugin.

## 7 Test data

Test data for getting yourself familiarised with the NanoJ-SQUIRREL package is available to download from our [Bitbucket repository](#). The available datasets are:

- **Simulated data for error mapping** The ground truth for this dataset was generated by tracing a real image of fixed actin filaments. A widefield reference image (pixel size 100nm) was generated through convolution with a PSF of width 135nm and addition of Poisson and Gaussian noise. The super-resolution image stack (pixel size 20nm) contains four images: the first image has one filament bent from its true position, and one filament displaced by 500nm from its true position; the second image has one filament displaced by 300nm; the third image has one large filament missing entirely; the fourth image is undersampled. Error maps and metrics from our running of SQUIRREL are also provided.
- **Real data for error mapping** The reference image is an LED-illuminated widefield image of Mitotracker Red-labelled mitochondria in a fixed cell (pixel size 86.7nm). The super-resolution image stack (pixel size 21.67nm) consists of five images of the same region taken using Andor SRRF-Stream at five different LED illuminations. Again, error maps and metrics that we generated by running SQUIRREL analysis are provided.

- **Real data for FRC** There are two datasets provided for FRC mapping. ‘Microtubules FRC stack.tif’ is two ThunderSTORM-generated images of fixed underlabelled microtubules where the first image comprises localisations from odd frames only and the second image comprises localisations from even frames only (pixel size 20nm). ‘Microtubules FRC stack 2.tif’ is the same as the previous dataset, except here with fixed overlabelled microtubules (pixel size 21.33nm). FRC maps and values which we generated are also included.

# Glossary

**blocks** Small spatial subset of an image upon which analysis is performed.

**frame** A single image.

**FRC** Fourier Ring Correlation - a method which uses correlations in Fourier space to estimate resolution. It requires two identically acquired images of the same structure.

**FWHM** Full Width at Half Maximum - a common measurement used for describing the width of a Gaussian distribution. It is equal to  $2\sqrt{2\ln 2}\sigma$ .

**GPU** Graphics Processing Unit - this is the graphics card inside your computer. It has many more cores than the CPU and so computation time can be significantly shortened by massively parallelising calculations to run on the GPU.

**image stack** Image containing  $n \geq 1$  frames.

**meta-optimisation** Quick calculations performed to provide an initial estimate for the main PSO optimisers.

**PSF** Point Spread Function - the mathematical description of the spatial spreading of a point source of light through the microscope.

**PSO** Particle Swarm Optimiser - a stochastic method for minimising an N-dimensional problem based on animal swarming behaviour. See [https://en.m.wikipedia.org/wiki/Particle\\_swarm\\_optimization](https://en.m.wikipedia.org/wiki/Particle_swarm_optimization) for a more detailed explanation..

**reference image** Low-resolution high-confidence image of structure.

**RSE** Resolution Scaled Error - a metric describing the whole-image root mean square error between the reference image and the resolution- and intensity-scaled super-resolution image. It exhibits intensity-dependence and as such is useful for assessing any non-linear intensity scaling. Lower values indicate better agreement.

**RSF** Resolution Scaling Function - the mathematical kernel required to transform a super-resolution image into its reference image equivalent.

**RSP** Resolution Scaled Pearson's coefficient - a metric describing the whole-image Pearson correlation coefficient between the reference image and the resolution- and intensity-scaled super-resolution image. This is a normalised metric between  $-1$  and  $1$ , and is independent of image intensity. Higher values indicate better agreement.