

# **Single Molecule Light Microscopy ImageJ Plugins**

**Alex Herbert**

MRC Genome Damage and Stability Centre  
School of Life Sciences  
University of Sussex  
Science Road  
Falmer  
BN1 9RQ

[a.herbert@sussex.ac.uk](mailto:a.herbert@sussex.ac.uk)

## Table of Contents

|  |    |
|--|----|
| 1 Introduction.....                            | 8  |
| 1.1 Fitting plugins.....                       | 9  |
| 1.2 Results plugins.....                       | 10 |
| 1.3 Analysis plugins.....                      | 11 |
| 1.4 Model plugins.....                         | 12 |
| 1.5 Calibration plugins.....                   | 13 |
| 1.6 Tool plugins.....                          | 14 |
| 2 Background.....                              | 15 |
| 2.1 Diffraction limit of light microscopy..... | 15 |
| 2.2 Approximation using a Gaussian.....        | 16 |
| 2.3 Localisation Fitting Method.....           | 16 |
| 3 Installation.....                            | 18 |
| 3.1 Install using ImageJ2/Fiji.....            | 18 |
| 3.2 Install using ImageJ version 1.....        | 18 |
| 4 ImageJ Plugins.....                          | 19 |
| 5 Fitting Plugins.....                         | 20 |
| 5.1 Simple Fit.....                            | 20 |
| 5.1.1 Parameters.....                          | 20 |
| 5.1.2 Calibration.....                         | 21 |
| 5.1.3 Advanced Settings.....                   | 25 |
| 5.2 Peak Fit.....                              | 26 |
| 5.2.1 Imaging Calibration Parameters.....      | 27 |
| 5.2.2 Gaussian PSF Parameters.....             | 27 |
| 5.2.3 Maxima Identification Parameters.....    | 28 |
| 5.2.4 Fitting Parameters.....                  | 30 |
| 5.2.5 Multiple Peak Fitting Parameters.....    | 38 |
| 5.2.6 Filtering Parameters.....                | 39 |
| 5.2.7 Results Parameters.....                  | 40 |
| 5.2.8 Interactive Results Table.....           | 43 |
| 5.2.9 Live image display.....                  | 43 |
| 5.2.10 Image examples.....                     | 44 |
| 5.2.11 Running Peak Fit.....                   | 45 |
| 5.2.12 Additional Fitting Options.....         | 46 |
| 5.2.13 Interlaced Data.....                    | 47 |
| 5.3 Template Configuration.....                | 48 |
| 5.4 Fit Configuration.....                     | 48 |
| 5.5 Peak Fit (Series).....                     | 49 |
| 5.6 Batch Fit.....                             | 51 |
| 5.6.1 Configuration File.....                  | 51 |
| 5.6.2 Running the batch.....                   | 53 |
| 5.7 Spot Finder.....                           | 53 |
| 5.8 Spot Finder (Series).....                  | 53 |
| 5.9 Fit Maxima.....                            | 53 |
| 5.10 Gaussian Fit.....                         | 54 |
| 5.10.1 Maxima Identification.....              | 55 |
| 5.10.2 Gaussian Fitting.....                   | 56 |
| 6 Results Plugins.....                         | 59 |

|   |     |
|---|-----|
| 6.1 Results Manager.....                              | 59  |
| 6.1.1 Input options.....                              | 59  |
| 6.1.2 Output options.....                             | 59  |
| 6.2 Summarise Results.....                            | 60  |
| 6.3 Clear Memory Results.....                         | 60  |
| 6.4 Clear Memory Results (Multi).....                 | 60  |
| 6.5 Rename Results.....                               | 61  |
| 6.6 Resequence Results.....                           | 61  |
| 6.7 Calibrate Results.....                            | 62  |
| 6.8 Show Results Header.....                          | 63  |
| 6.9 Filter Results.....                               | 64  |
| 6.10 Free Filter Results.....                         | 64  |
| 6.11 Split Results.....                               | 66  |
| 6.11.1 Analysis.....                                  | 67  |
| 6.12 Results Match Calculator.....                    | 67  |
| 6.12.1 Interactive Results Table.....                 | 68  |
| 6.13 Trace Match Calculator.....                      | 69  |
| 6.14 Spot Inspector.....                              | 70  |
| 7 Analysis Plugins.....                               | 72  |
| 7.1 Drift Calculator.....                             | 72  |
| 7.1.1 Localisation Sub-images.....                    | 76  |
| 7.1.2 Drift File.....                                 | 77  |
| 7.1.3 Reference Stack Alignment.....                  | 77  |
| 7.1.4 Marked ROIs.....                                | 78  |
| 7.1.5 Image alignment using correlation analysis..... | 78  |
| 7.2 Trace Molecules.....                              | 79  |
| 7.2.1 Background.....                                 | 79  |
| 7.2.2 Trace Molecules Plugin.....                     | 79  |
| 7.2.3 Optimisation.....                               | 82  |
| 7.2.4 Memory Output.....                              | 84  |
| 7.2.5 File output.....                                | 85  |
| 7.2.6 Refit option.....                               | 86  |
| 7.3 Cluster Molecules.....                            | 87  |
| 7.3.1 Clustering Algorithms.....                      | 88  |
| 7.4 Draw Clusters.....                                | 90  |
| 7.4.1 Drawing cluster centroids.....                  | 92  |
| 7.5 Density Image.....                                | 92  |
| 7.5.1 Available Density Score functions.....          | 94  |
| 7.5.2 Ripley's L-plot.....                            | 95  |
| 7.6 Dark Time Analysis.....                           | 96  |
| 7.7 Blink Estimator.....                              | 97  |
| 7.8 Neighbour Analysis.....                           | 99  |
| 7.9 Filter Analysis.....                              | 100 |
| 7.9.1 Input Data.....                                 | 100 |
| 7.9.2 Available Filters.....                          | 100 |
| 7.9.3 Additional parameters.....                      | 103 |
| 7.9.4 Output.....                                     | 104 |
| 7.10 Create Filters.....                              | 105 |

|   |     |
|---|-----|
| 7.11 Filter Analysis (File).....                | 107 |
| 7.12 Spot Analysis.....                         | 107 |
| 7.12.1 Input Images.....                        | 107 |
| 7.12.2 Plugin Interface.....                    | 108 |
| 7.12.3 Analysis Workflow.....                   | 113 |
| 7.12.4 Results Files.....                       | 113 |
| 7.13 Spot Analysis (Add).....                   | 114 |
| 7.14 Fourier Image Resolution.....              | 114 |
| 7.14.1 Input Data.....                          | 115 |
| 7.14.2 Threshold methods.....                   | 116 |
| 7.14.3 Spurious correlation (Q) correction..... | 117 |
| 7.14.4 Data Selection.....                      | 118 |
| 7.14.5 Parameters.....                          | 118 |
| 7.14.6 Output.....                              | 120 |
| 7.15 FIRE Q Estimation.....                     | 122 |
| 7.15.1 Estimating Q.....                        | 123 |
| 7.15.2 Parameters.....                          | 124 |
| 7.15.3 Output.....                              | 126 |
| 8 PC PALM Plugins.....                          | 131 |
| 8.1 PC-PALM Molecules.....                      | 131 |
| 8.2 PC-PALM Analysis.....                       | 131 |
| 8.3 PC-PALM Spatial Analysis.....               | 131 |
| 8.4 PC-PALM Save Results.....                   | 131 |
| 8.5 PC-PALM Load Results.....                   | 131 |
| 8.6 PC-PALM Fitting.....                        | 132 |
| 8.7 PC-PALM Clustering.....                     | 132 |
| 9 Model Plugins.....                            | 133 |
| 9.1.1 Benchmarking.....                         | 133 |
| 9.2 PSF Creator.....                            | 133 |
| 9.2.1 Input image.....                          | 134 |
| 9.2.2 Analysis.....                             | 134 |
| 9.2.3 Parameters.....                           | 137 |
| 9.2.4 Output.....                               | 138 |
| 9.3 PSF Drift.....                              | 140 |
| 9.3.1 Drift Calculation.....                    | 141 |
| 9.3.2 Parameters.....                           | 142 |
| 9.3.3 Output.....                               | 143 |
| 9.3.4 Saving the Drift.....                     | 145 |
| 9.4 PSF Combiner.....                           | 146 |
| 9.5 Create Data.....                            | 147 |
| 9.5.1 Simulation.....                           | 147 |
| 9.5.2 Point Spread Function.....                | 149 |
| 9.5.3 Noise model.....                          | 150 |
| 9.5.4 Particle distribution.....                | 151 |
| 9.5.5 Parameters.....                           | 153 |
| 9.5.6 Data Summary.....                         | 157 |
| 9.5.7 Compound molecules.....                   | 158 |
| 9.5.8 Sampled Statistics.....                   | 159 |

|  |     |
|--|-----|
| 9.5.9 Memory Output.....               | 159 |
| 9.6 Create Simple Data.....            | 160 |
| 9.7 Create Benchmark Data.....         | 162 |
| 9.7.1 Fitting Limits.....              | 164 |
| 9.8 Fit Benchmark Data.....            | 164 |
| 9.8.1 Data Summary.....                | 165 |
| 9.9 Benchmark Analysis.....            | 168 |
| 9.10 Create Spot Data.....             | 168 |
| 9.11 Filter Spot Data.....             | 170 |
| 9.11.1 Data Summary.....               | 173 |
| 9.12 Fit Spot Data.....                | 179 |
| 9.12.1 Analysis.....                   | 179 |
| 9.12.2 Parameters.....                 | 180 |
| 9.12.3 Results.....                    | 182 |
| 9.13 Benchmark Filter Analysis.....    | 186 |
| 9.13.1 Input Filters.....              | 186 |
| 9.13.2 Analysis.....                   | 188 |
| 9.13.3 Parameters.....                 | 191 |
| 9.13.4 Results.....                    | 197 |
| 9.14 Doublet Analysis.....             | 202 |
| 9.14.1 Analysis.....                   | 202 |
| 9.14.2 Parameters.....                 | 203 |
| 9.14.3 Results.....                    | 206 |
| 9.15 Doublet Filter Analysis.....      | 213 |
| 9.15.1 Selecting a two spot model..... | 213 |
| 9.15.2 Parameters.....                 | 214 |
| 9.15.3 Results.....                    | 216 |
| 9.16 Image Background.....             | 219 |
| 9.16.1 Image analysis.....             | 219 |
| 9.17 Load Localisations.....           | 219 |
| 10 Calibration Plugins.....            | 221 |
| 10.1 PSF Calculator.....               | 221 |
| 10.2 PSF Estimator.....                | 223 |
| 10.3 Mean-Variance Test.....           | 226 |
| 10.3.1 Multiple Input Images.....      | 226 |
| 10.3.2 Analysis.....                   | 227 |
| 10.3.3 Output.....                     | 227 |
| 10.3.4 Single Image Mode.....          | 229 |
| 10.4 Mean-Variance Test (EM-CCD).....  | 229 |
| 10.4.1 Multiple Input Images.....      | 229 |
| 10.4.2 Analysis.....                   | 230 |
| 10.4.3 Output.....                     | 230 |
| 10.4.4 Single Image Mode.....          | 231 |
| 10.5 EM-Gain Analysis.....             | 231 |
| 10.5.1 EM-CCD Probability Model.....   | 231 |
| 10.5.2 Input image.....                | 232 |
| 10.5.3 Parameters.....                 | 233 |
| 10.5.4 Simulation Mode.....            | 233 |

|  |     |
|--|-----|
| 10.5.5 Results.....  | 234 |
| 10.6 EM-Gain PMF.....  | 235 |
| 10.7 Diffusion Rate Test.....                                  | 237 |
| 10.7.1 Grid Walk simulation.....                               | 237 |
| 10.7.2 Random Move simulation.....                             | 237 |
| 10.7.3 Confined Diffusion.....                                 | 237 |
| 10.7.4 Analysis.....   | 238 |
| 10.7.5 Output.....   | 238 |
| 10.7.6 Extra options.....                                      | 241 |
| 10.7.7 Aggregate steps parameter.....                          | 241 |
| 10.7.8 MSD Analysis parameter.....                             | 241 |
| 10.7.9 Precision parameter.....                                | 242 |
| 10.8 Trace Diffusion.....                                      | 243 |
| 10.8.1 Analysis.....   | 243 |
| 10.8.2 Jump Distance analysis.....                             | 244 |
| 10.8.3 MSD Correction.....                                     | 245 |
| 10.8.4 Precision Correction.....                               | 246 |
| 10.8.5 MSD and Precision correction.....                       | 246 |
| 10.8.6 Selecting the best fit.....                             | 247 |
| 10.8.7 Parameters.....   | 247 |
| 10.8.8 Output.....   | 249 |
| 10.9 Trace Diffusion (Multi).....                              | 253 |
| 11 Tools Plugins.....  | 255 |
| 11.1 Smooth Image.....   | 255 |
| 11.1.1 Smoothing within the Peak Fit plugin.....               | 256 |
| 11.2 Binary Display.....                                       | 257 |
| 11.3 Reset Display.....  | 257 |
| 11.4 Pixel Filter.....   | 257 |
| 11.5 Noise Estimator.....                                      | 258 |
| 11.5.1 Image Residuals.....                                    | 259 |
| 11.5.2 Noise estimation within the Peak Fit plugin.....        | 259 |
| 11.6 Median Filter.....  | 260 |
| 11.7 Overlay Image.....  | 261 |
| 12 Toolset Plugins.....  | 262 |
| 12.1 Install SMLM Toolset.....                                 | 262 |
| 12.2 Show SMLM Tools.....                                      | 262 |
| 12.3 Create SMLM Tools Config.....                             | 264 |
| Appendix A Localisation Precision.....                         | 265 |
| A.1 Signal Precision.....                                      | 265 |
| A.2 Localisation Precision for Least Squares Fitting.....      | 265 |
| A.3 Localisation Precision for Maximum Likelihood Fitting..... | 266 |
| Appendix B Comparison Metrics.....                             | 268 |
| B.1 Recall.....  | 268 |
| B.2 Precision.....   | 268 |
| B.3 Jaccard.....   | 268 |
| B.4 F-score.....   | 269 |
| B.5 FNR.....   | 269 |
| B.6 FDR.....   | 269 |

## *GDSC SMLM ImageJ Plugins*

|  |     |
|--|-----|
| B.7 Defining the Actual or Predicted Points.....   | 269 |
| B.8 Metrics requiring the True Negative count..... | 269 |
| Appendix C MSD Correction.....                     | 271 |
| C.1 $n < 1$ .....                                  | 273 |
| C.2 $n > 1$ .....                                  | 273 |
| C.3 Fitting the plot of MSD verses N frames.....   | 275 |
| 13 References.....                                 | 276 |

## 1 Introduction

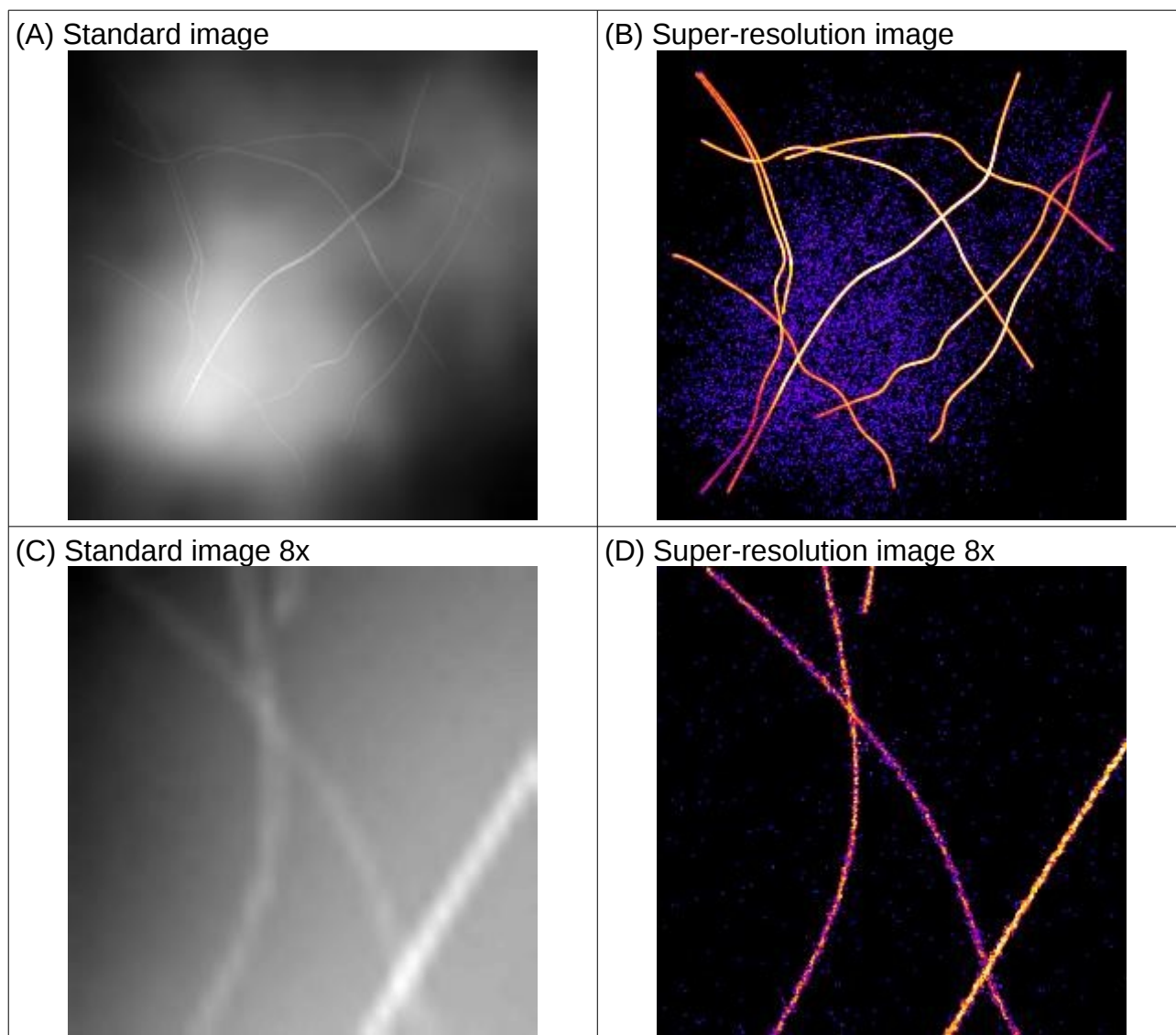
Super-resolution microscopy can be used to obtain a higher level of resolution from an image than conventional light microscopy.

In conventional light microscopy the subject is illuminated and all the light interacting with the subject is captured simultaneously into an image. The resolution is limited since the optics of the microscope cannot focus the light perfectly.

Super-resolution microscopy uses small molecules that can be activated to emit light. If a single molecule is activated its position can be localised using software analysis. Using thousands of localisations it is possible to reconstruct a virtual image of the subject. This increases the image resolution to the precision of the localisation method. The improvement depends on the imaging conditions but is often 5-10 times higher.

Illustration 1 shows the increased resolution of single molecule localisation microscopy. The difference is most noticeable where structures meet since the single molecules have a cleaner signal when activated individually than the combined signal of all the molecules together.





*Illustration 1: Example images of standard and single molecule localisation microscopy*

The GDSC Single Molecule Light Microscopy (SMLM) plugins provide various tools for single molecule localisation analysis. The following plugins are available:

### 1.1 Fitting plugins

|                        |   |
|------------------------|---|
| Simple Fit             | Performs fitting on an image to generate a table of molecule localisations and an image reconstruction.<br>Provides a step through guide for initial configuration of microscope parameters |
| Peak Fit               | Performs fitting on an image to generate a list of molecule localisations   |
| Template Configuration | Allows the user to load custom configuration templates from a directory   |

|                      |   |
|----------------------|---|
| Fit Configuration    | Allows setting the fitting parameters and saving them to file                     |
| Peak Fit (Series)    | Performs fitting on folder of images to generate a list of molecule localisations |
| Batch Fit            | Runs the Peak Fit analysis on an image for each configuration option in the batch |
| Spot Finder          | Identifies candidate maxima in an image   |
| Spot Finder (Series) | Identifies candidate maxima in a folder of images                                 |
| Fit Maxima           | Performs PSF fitting on candidate maxima  |
| Gaussian Fit         | Allows interactive identification of maxima and 2D Gaussian fitting               |

## 1.2 Results plugins

|                              |  |
|------------------------------|--|
| Result Manager               | Allows conversion of the localisation results into different formats, e.g. files or images   |
| Summarise Results            | Displays a summary of the results held in memory   |
| Clear Memory Results         | Removes all results held in memory   |
| Clear Memory Results (Multi) | Removes selected results held in memory  |
| Rename Results               | Allows the results sets to be renamed  |
| Resequenece Results          | Allows the frame number of results to be rebuilt assuming a repeating pattern of data and non-data frames  |
| Calibrate Results            | Allows results held in memory to be calibrated   |
| Show Results Header          | Shows the header information from any support localisation results file format   |
| Filter Results               | Filters a list of localisations using signal strength, coordinate shift and PSF width  |
| Free Filter Results          | Filters a list of localisations using a custom filter specified using a text description. Multiple filters can be combined with AND/OR operators |
| Split Results                | Splits a set of localisation results using a 2D mask   |
| Results Match Calculator     | Calculate the match statistics between two results sets  |

|                        |   |
|------------------------|---|
| Trace Match Calculator | Calculate the match statistics between two sets of traced molecules                     |
| Spot Inspector         | Extracts the fitted spots from an image into a stack ordered by the user-selected score |

### 1.3 Analysis plugins

|                          |   |
|--------------------------|---|
| Drift Calculator         | Corrects stage drift using: sub-image alignment; fiducial markers within an image; reference stack alignment; or a drift file                               |
| Trace Molecules          | Traces molecules through time using time and distance thresholds (using a type of single-linkage clustering)  |
| Cluster Molecules        | Clusters molecules through time using time and distance thresholds (using centroid-linkage clustering)  |
| Draw Clusters            | Draws collections of localisations with the same ID on an image, for example the output from <code>TRACE MOLECULES</code> or <code>CLUSTER MOLECULES</code> |
| Density Image            | Calculates the local density around localisations and displays an image coloured using the density score  |
| Dark Time Analysis       | Determines the maximum dark time for a fluorophore from localisation data   |
| Blink Estimator          | Estimate the blinking rate of fluorophores in a results set   |
| Neighbour Analysis       | Saves all localisations paired with their neighbour (if present) to file  |
| Filter Analysis          | Performs filtering on a set of categorised localisation results and computes match statistics for each filter   |
| Create Filters           | Used to prepare a large set of filters for use in the <code>FILTER ANALYSIS (FILE)</code> plugin  |
| Filter Analysis (File)   | Performs filtering on a set of categorised localisation results and computes match statistics for each filter defined in the input file                     |
| Spot Analysis            | Allows analysis of the signal and on/off times for fixed fluorophore spots in an image stack  |
| Spot Analysis (Add)      | This plugin provides a named plugin command for the <code>ADD</code> button of the <code>SPOT ANALYSIS</code> plugin  |
| Fourier Image Resolution | Analyses the resolution of an image using Fourier Ring Correlation  |
| PC-PALM Molecules        | Prepare a set of localisations for Pair Correlation analysis  |

|                          |   |
|--------------------------|---|
|                          |   |
| PC-PALM Analysis         | Produce Pair Correlation curve for a set of localisations selected from a super-resolution image                                |
| PC-PALM Spatial Analysis | Performs spatial analysis to plot the molecule density around each localisation as a function of distance from the localisation |
| PC-PALM Save Results     | Saves all the PC-PALM results held in memory to a results folder  |
| PC-PALM Load Results     | Load all the PC-PALM results from a results folder to memory  |
| PC-PALM Fitting          | Combine multiple Pair Correlation curves and fit them using models for random or clustered distributions                        |
| PC-PALM Clusters         | Find clusters of molecules using a partial centroid-linkage hierarchical clustering algorithm                                   |

## 1.4 Model plugins

|                       |   |
|-----------------------|---|
| PSF Creator           | Extracts the PSF from a Z-stack image of markers, e.g. quantum dots or fluorescent beads  |
| PSF Drift             | Computes the drift in a PSF image   |
| PSF Combiner          | Combines several PSF images into a single average PSF   |
| Create Data           | Creates an image by simulating single molecule localisations using a model of photoactivated diffusing fluorophore complexes. The PSF can be simulated or real using an input PSF image |
| Create Simple Data    | Creates an image by simulating single molecule localisations at a specified density   |
| Create Benchmark Data | Creates an image by simulating single molecule localisations in a fixed location  |
| Fit Benchmark Data    | Fit the image created by <code>CREATE BENCHMARK DATA</code> and compute statistics on the accuracy and precision of fitting   |
| Benchmark Analysis    | Compute statistics on the accuracy and precision of fitting using different methods. Statistics are only computed for all the localisations that were fit successfully by each method   |
| Create Spot Data      | Creates a sparse image by simulating zero or one localisation per frame   |

|                           |  |
|---------------------------|--|
| Filter Spot Data          | Filter the image created by <code>CREATE SIMPLE DATA</code> or <code>CREATE SPOT DATA</code> and compute statistics on the accuracy and precision of identifying spot candidates |
| Fit Spot Data             | Fits all the candidate spots identified by the <code>FILTER SPOT DATA</code> plugin  |
| Benchmark Filter Analysis | Run different filtering methods on a set of benchmark fitting results outputting performance statistics on the success of the filter   |
| Doublet Analysis          | Fits candidate spots identified by the <code>FILTER SPOT DATA</code> plugin as a single or double spot (doublet) and scores the results  |
| Doublet Filter Analysis   | Filters all the fit results produced by the <code>DOUBLET ANALYSIS</code> plugin using the specified filter settings and scores the results                                      |
| Image Background          | Produces a background intensity image and a mask from a sample <i>in vivo</i> image. This can be used to simulate realistic fluorophore distributions                            |
| Load Localisations        | Loads the XYZ location file from <code>CREATE DATA</code> into memory for analysis   |

## 1.5 Calibration plugins

|  |   |
|--|---|
| PSF Calculator                                   | Allows calculation of the Point Spread Function (PSF) size given the microscope imaging parameters  |
| PSF Estimator                                    | Estimates the PSF by performing fitting on a sample image   |
| Mean-Variance Test & Mean-Variance Test (EM-CCD) | Opens a folder of images and computes a Mean-Variance test to determine the gain and read noise of the microscope camera.<br>Gain can be calculated for standard or Electron Multiplying (EM) cameras |
| EM-Gain Analysis                                 | Analysis a white light image from an EM-CCD camera and fits a model to obtain the bias, EM-gain, read noise and average photons per pixel   |
| EM-Gain PMF                                      | Displays a plot of the probability mass function (PMF) of the expected value of a pixel on an EM-CCD camera given an average number of photons  |
| Diffusion Rate Test                              | Simulate molecule diffusion and fit a graph of mean-squared displacement to determine the diffusion coefficient   |
| Trace Diffusion                                  | Trace molecules through consecutive frames. Mean-squared displacement analysis is performed on the traces to calculate a  |

|                         |  |
|-------------------------|--|
|                         | diffusion coefficient  |
| Trace Diffusion (Multi) | Allows the TRACE DIFFUSION plugin to be run with multiple input datasets |

## 1.6 Tool plugins

In addition to the principle plugins for localisation fitting and analysis there are several utility plugins provided.

|                          |   |
|--------------------------|---|
| Install SMLM Toolset     | Adds an ImageJ Toolset containing common commands that can be used from the ImageJ toolbar    |
| Show SMLM Tools          | Display a SMLM TOOLS window with buttons to run plugins                                       |
| Create SMLM Tools config | Create a configuration file allowing the SMLM TOOLS panel to be customised                    |
| Smooth Image             | Performs smoothing on an image (identical to that performed when identifying maxima)          |
| Binary Display           | Switches an image to binary (white/black) to allow quick visualisation of localisations       |
| Reset Display            | Resets a binary image back to the standard display  |
| Pixel Filter             | Perform filtering to remove hot pixels from an image  |
| Noise Estimator          | Estimate the noise in an image using various methods  |
| Median Filter            | Compute the median of an image, on a per-pixel basis, using a rolling window at set intervals |
| Overlay Image            | Allow an image to be added as an overlay with a transparent background                        |

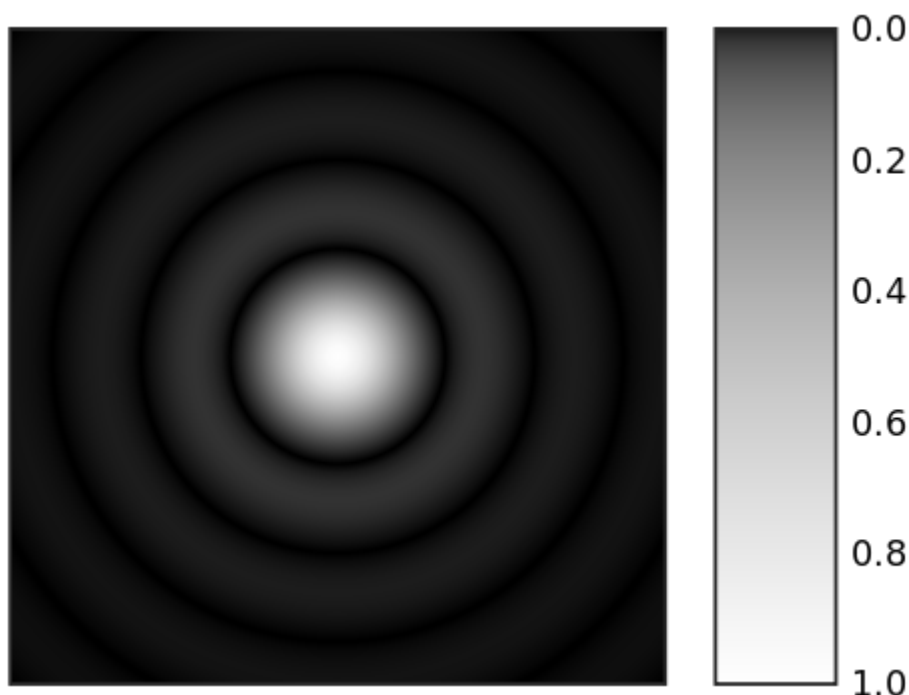
All of the plugins can be incorporated into ImageJ macros to allow automation of image analysis workflows.

## 2 Background

### 2.1 Diffraction limit of light microscopy

Light microscopy uses lens optics to focus light from a sample onto an imaging plane. However even with a perfect set of optics it is not possible to bring all the light into perfect focus. This is because light is diffracted when passing through an opening.

The diffraction pattern resulting from a uniformly-illuminated circular aperture has a bright region in the centre, known as the Airy disk which together with the series of concentric bright rings around is called the Airy pattern (see Illustration 2).



*Illustration 2: A computer-generated image of an Airy pattern. Image taken from Wikimedia Commons.*

The inability to perfectly focus light sets a limit on the resolution that can be achieved using conventional light microscopy. The Rayleigh criterion for barely resolving two objects that are point sources of light is that the centre of the Airy disk for the first object occurs at the first minimum of the Airy disk of the second. This separation can be calculated from the following formula:

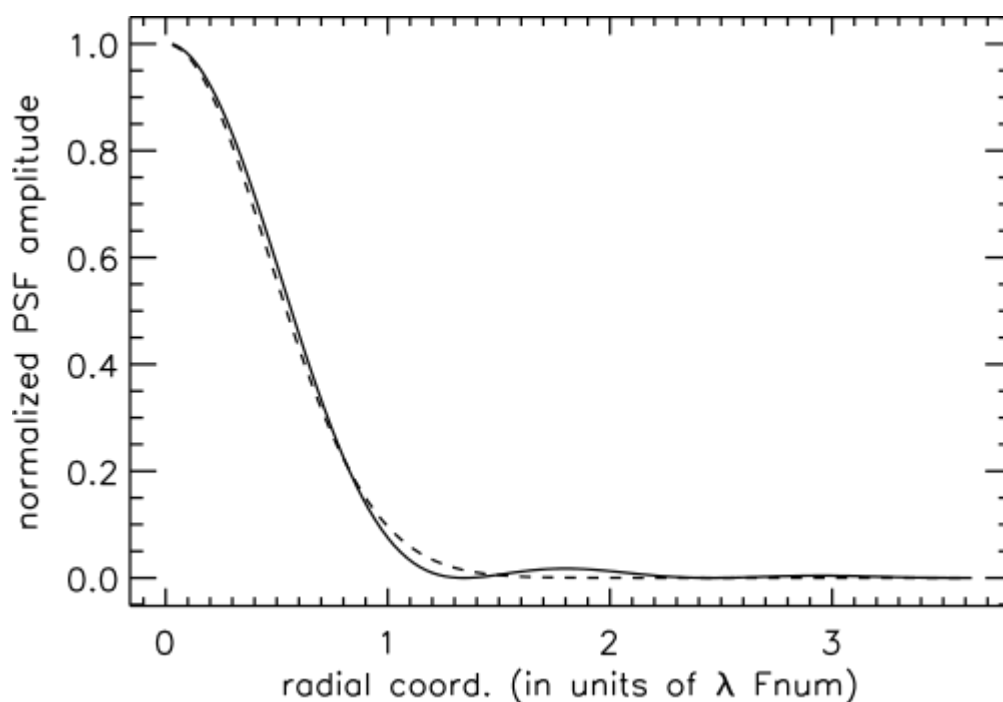
$$\sin(\theta) = 1.22 \frac{\lambda}{d}$$

where  $\theta$  is the angle at which the first minimum occurs,  $\lambda$  is the wavelength of the light and  $d$  is the diameter of the aperture. Therefore the resolution that can be achieved is a function of the wavelength of the light and the size of the aperture used in the optical system.

## 2.2 Approximation using a Gaussian

The Airy pattern falls rather slowly to zero with increasing distance from the centre such that the outer rings contain a significant portion of the integrated intensity. As a result the spot size is undefined (i.e. infinite) when using the entire pattern.

An alternative measure of the spot size is to ignore the relatively small outer rings of the Airy pattern and to approximate the central lobe with a Gaussian profile. This is shown in Illustration 1.



*Illustration 3: Approximation of the Airy pattern (solid line) using a Gaussian (dashed line). Image taken from Wikimedia Commons.*

The Gaussian profile closely approximates the profile of the Airy pattern in the plane of focus. This allows the location of a point source of light to be found by 2D Gaussian fitting. However this approximation breaks down if the focal plane is above or below the imaging plane. In this instance the pattern still exists as a series of rings but the relative strength of central spot is reduced. Consequently the Gaussian fit will be poor, or not possible, and the localisation precision will be reduced.

## 2.3 Localisation Fitting Method

The single-molecule localisation method used in the GDSC Single Molecule Light Microscopy (SMLM) plugins is based on 2D Gaussian fitting.

For each frame candidate spots are identified and then fit using a local region surrounding each candidate [Wolter *et al*, 2010]. An estimate of the PSF width is required which can be obtained from the wavelength and optical system used for acquisition, or estimated from the image.

Candidates are identified using a smoothing filter on the image followed by non-maximal suppression. Smoothing is done using a single filter such or a difference filter where the



second smoothed image is subtracted from the first. Various filters are available such as mean, median or Gaussian. Local maxima are identified within a configured search region which is at least 3x3 pixels.

Peak candidates are processed in descending height order and a 2D Gaussian is fit to the peak using a region around the candidate of typically 2 to 5 times the estimated Gaussian standard deviation. Fitting is performed using either non-linear least squares Levenberg-Marquardt method or Maximum Likelihood fitting until convergence or the maximum iterations is exceeded. Fitted spots are filtered using signal-to-noise, width, precision and coordinate shift criteria. Processing is stopped when a consecutive number of fits fail.

To account for high density samples neighbour peaks within the block region are included in the fit if they are within a fraction of the height of the candidate peak, typically 30%. If multiple peak fitting fails then single peak fitting is attempted. Additionally the candidate can be fit using a two peak model if the fit residuals show an asymmetric distribution around the centre. The doublet fit is selected if it improves the corrected Akaike Information Criterion (AIC) [Hurvich & Tsai, 1989]. To prevent over-counting when fitting multiple peaks a check is made for duplicates using a distance criterion before adding to the results. The method is applicable to all types of localisation microscopy data and results are suitable input for filtering methods using structural models.

Image frames are processed independently allowing multi-threaded operation. Results can be output to memory, file or a rendered image using various methods.

### 3 Installation

The plugin is designed to run within ImageJ. This can be the original ImageJ (version 1) or any modified version of ImageJ such as ImageJ2 or Fiji.

#### 3.1 Install using ImageJ2/Fiji

The SMLM plugins are distributed using an ImageJ2/Fiji update site. This allows the plugins to be easily installed and kept up-to-date. To install the plugins using Fiji just follow the instructions here:

[http://fiji.sc/How\\_to\\_follow\\_a\\_3rd\\_party\\_update\\_site](http://fiji.sc/How_to_follow_a_3rd_party_update_site)

and add the GDSC SMLM update site. Fiji will automatically check for a new version during start-up and install it if desired.

All the plugins will appear under the 'Plugins > GDSC SMLM' menu.

#### 3.2 Install using ImageJ version 1

The plugin is designed to run within ImageJ. You can obtain the latest version of ImageJ from:

<http://rsbweb.nih.gov/ij/download.html>

To get the plugins you can download the latest Jar files from the update site and put them in your ImageJ plugins folder. The jars can be found here:

<http://sites.imagej.net/GDSC-SMLM/>

You will also need to install the additional Apache Commons Math library. This is already included in Fiji so is not on the update site. You can get the jar file here:

<http://www.sussex.ac.uk/gdsc/intranet/files/commons-math3-3.2.jar>

Place all of the following Jar files into the ImageJ plugins directory:

| Jar File              | Description                                      |
|-----------------------|--|
| GDSC SMLM             | Contains the SMLM plugins                        |
| Apache Commons Math 3 | Contains Math routines                           |
| EJML                  | Efficient Java Matrix Library for linear algebra |
| Jtransforms           | Library for multi-threaded Fourier transforms    |
| Xstream               | A library for reading/writing XML                |

The plugins will be visible under the 'PLUGINS > GDSC SMLM' menu.

An ImageJ toolset can be installed using the 'GDSC SMLM > TOOLSET > INSTALL SMLM TOOLSET' plugin. When selected the toolset adds a set of buttons on the ImageJ toolbar for commonly used plugins. More details can be found in section 12: Toolset Plugins.

## 4 ImageJ Plugins

The various ImageJ plugins allow the processing of single-molecule light microscopy images and analysis of the results. This includes finding and fitting spots on the image; reconstructing an image from the list of localisations; analysis of the blinking rate of fluorophores; and tracing of fluorophore molecules through time.

The plugins used to analyse a set of localisations only require that the localisations be loaded into memory. The localisations do not have to be computed by the SMLM fitting plugins and can be generated by another program. For example it is possible to read the following file formats for analysis:

- RapidSTORM
- Nikon NSTORM

The plugins have been divided into the following sub-sets which are detailed in the remaining sections:

| Jar File    | Description  |
|-------------|--|
| Fitting     | For identification of localisations on an image  |
| Results     | Allow loading and saving results in different formats. Results can be filtered to subsets and compared to a reference set, e.g. for benchmarking |
| Analysis    | Perform analysis on localisations for example blinking rate estimation, molecule tracing and Fourier image resolution                            |
| PC PALM     | Plugins for Pair Correlation (PC) analysis   |
| Model       | Simulate single-molecule images  |
| Calibration | Estimate PSF widths and allow calibration of the imaging camera noise and gain   |
| Tools       | Utility plugins for image manipulation   |
| Toolset     | For install of the SMLM Toolset and configuration of the SMLM Tools window   |

## 5 Fitting Plugins

The following plugins use the SMLM fitting engine to find and fit spots on an image.

It is vital that the method used by the software is understood when adjusting the parameters from the defaults. Please refer to section 2.3: Localisation Fitting Method for more details.

The plugins are described in the following sections using the order presented on the **PLUGINS > GDSC SMLM > FITTING** menu.

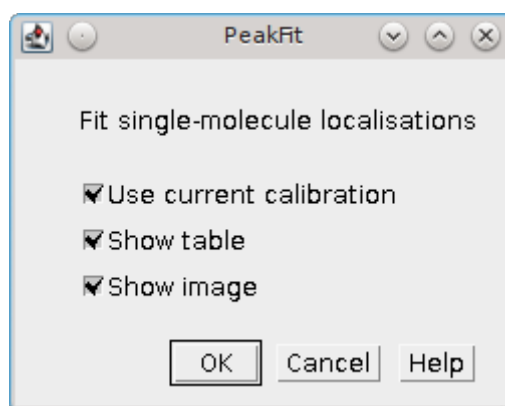
### 5.1 Simple Fit

The **SIMPLE FIT** plugin provides a single plugin to fit localisations on an image and produce a table and image reconstruction of the results. The fitting is performed using the fitting defaults, i.e. no fitting options are presented. This simplifies the fitting process to a single-click operation.

The plugin must be run when a single-molecule image is the currently active window in ImageJ. Each frame of the image will be analysed and the localisations recorded in a table and/or drawn on a high resolution image reconstruction.

The fitting is performed using multi-threaded code with a thread analysing each frame. Consequently results can appear out-of-order in the results table.

The plugin dialog has a simple appearance as shown in Illustration 47.



*Illustration 4: Simple Fit dialog*

#### 5.1.1 Parameters

The plugin offers the following parameters.

| Parameter               | Description  |
|-------------------------|--|
| USE CURRENT CALIBRATION | <p>If selected use the current SMLM configuration. Otherwise run the configuration wizard.</p> <p>This option is only shown if a SMLM configuration file can be found. If no file is found then the configuration wizard is run by default</p> |

|            |  |
|------------|--|
| SHOW TABLE | Show a table containing the localisations  |
| SHOW IMAGE | Show a super-resolution image of the localisations. The image will be 1024 pixels on the long edge |

Note that the plugin will run if no output options are selected. This is because the fitting results are also stored in memory. The results can be accessed and manipulated using the RESULTS plugins (see section 6).

It is possible to stop the fitting process using the ESCAPE key. All current results will be kept but the fitting process will end.

### 5.1.2 Calibration

The SMLM plugins require information about the system used to capture the images. This information is used in many of the analysis plugins. The information that is required is detailed below:

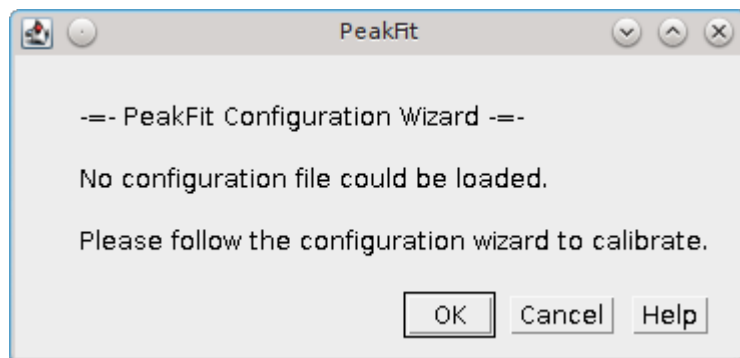
| Information | Description   |
|-------------|---|
| Pixel Pitch | <p>Specify the size of each pixel in the image.</p> <p>This is used to set the scale of the image and allows determination of the distance between localisations. This is used in many analysis plugins and for the localisation precision calculation.</p> <p>It is expected that the value would be in the range around 100nm. Input of an incorrect value will lead to incorrect precision estimates and require the user to adapt the analysis plugins and their results</p>  |
| Gain        | <p>Specify how many pixel values are equal to a photon of light. The units are Analogue-to-Digital Units (ADUs)/photon.</p> <p>This allows conversion of the pixel values to photons. It is used to convert the volume of the fitted 2D Gaussian to a photon count (the localisation signal).</p> <p>The gain is the total gain of the system (ADU/photon) and is equal to:<br/> <math display="block">[\text{Camera gain (ADU/e-)}] \times [\text{EM-gain}] \times [\text{Quantum Efficiency (e-/photon)}]</math></p> <p>Note: Check the units for this calculation as the camera gain can often be represented as electrons/ADU. In this case a reciprocal must be used. EM-gain has no units. Quantum Efficiency should be in the range 0-1; the units are electrons/photon.</p> <p>For an EM-CCD camera with an EM-gain of 250 the total gain may be in the range around 40. Camera gain and EM-gain can be calculated for your camera using the Mean-variance Test plugins (see section 10.3). The gain values and Q.E. may also</p> |

|               |   |
|---------------|---|
|               | <p>have been provided on a specification sheet with the camera.</p> <p>Input of an incorrect value will lead to incorrect precision estimates. These estimates are used in various analysis plugins to set parameters relative to the average fitting precision.</p>  |
| Exposure Time | <p>This is the length of time captured by each frame in milliseconds</p> <p>The exposure time is used in various analysis plugins. Input of an incorrect value will only effect the time scales reported in the results.</p>  |
| Peak Width    | <p>Specify the expected width of the Gaussian function that approximates the Point Spread Function Airy disk. This is how spread out the spot of light is when in focus. The width is specified in pixels.</p> <p>This value is used as an input to the fitting process to avoid having to guess the initial width for each spot processed. Since the width is approximately constant for the microscope it is valuable to input the expected width. The width is updated during the fitting process allowing fitting of out-of-focus spots.</p> <p>The width can be calculated using knowledge of the microscope objective and the wavelength of light. It can also be estimated from an image (see section 10.1: PSF Calculator).</p> <p>It is expected that this should be in the range around 1 pixel. Input of an incorrect value will lead to poor fitting performance since by default peaks that are too wide/narrow are discarded.</p> |

When the SIMPLE FIT plugin is run it attempts to load the SMLM configuration file. This is located in the user's home directory and is named gdsc.smlm.settings.xml. If no configuration file can be found then a configuration wizard is run to guide the user through calibration. If any dialog is cancelled then the SIMPLE FIT plugin terminates.

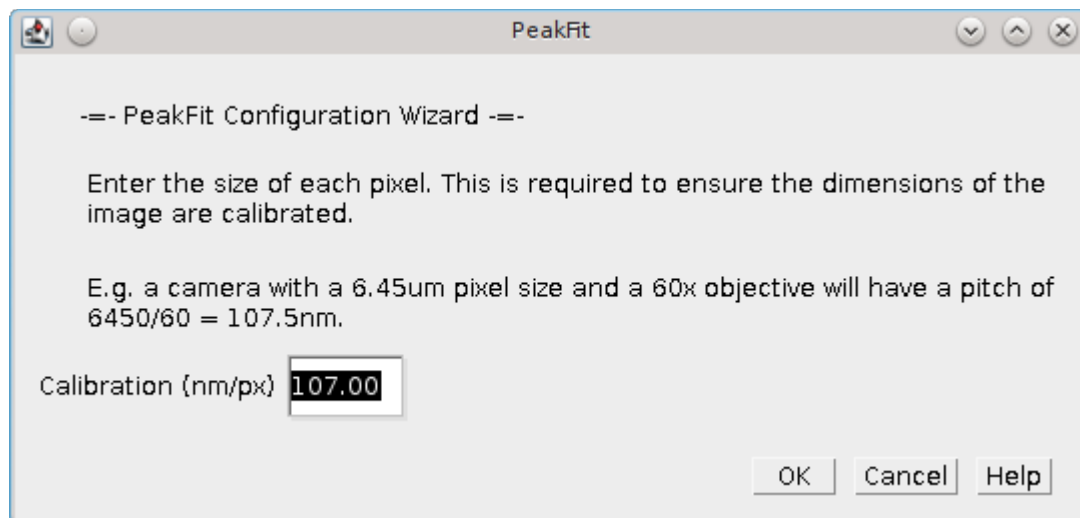
#### 5.1.2.1 Configuration Wizard 1: Introduction

The first dialog of the wizard warns the user that no configuration file could be loaded:



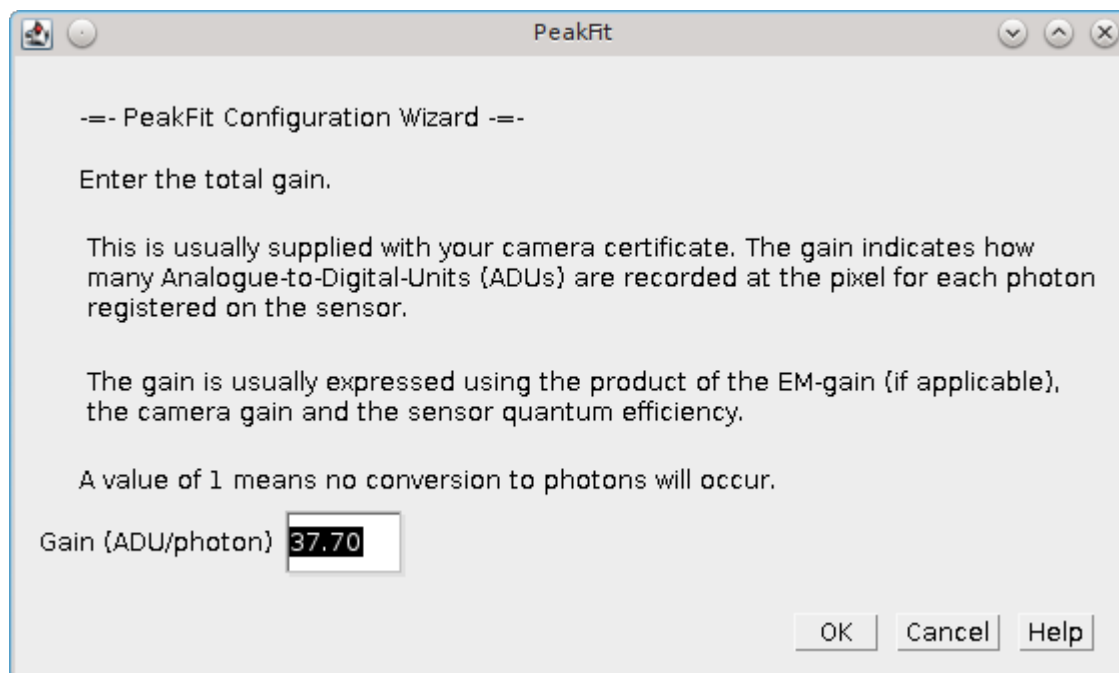
### 5.1.2.2 Configuration Wizard 2: Pixel Pitch

The second dialog of the wizard request the pixel pitch:



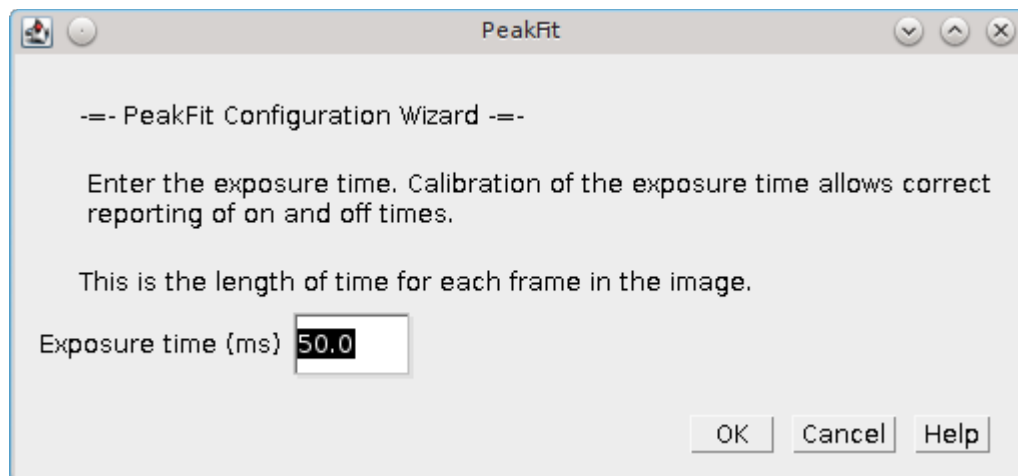
### 5.1.2.3 Configuration Wizard 3: Gain

The third dialog of the wizard requests the total gain:



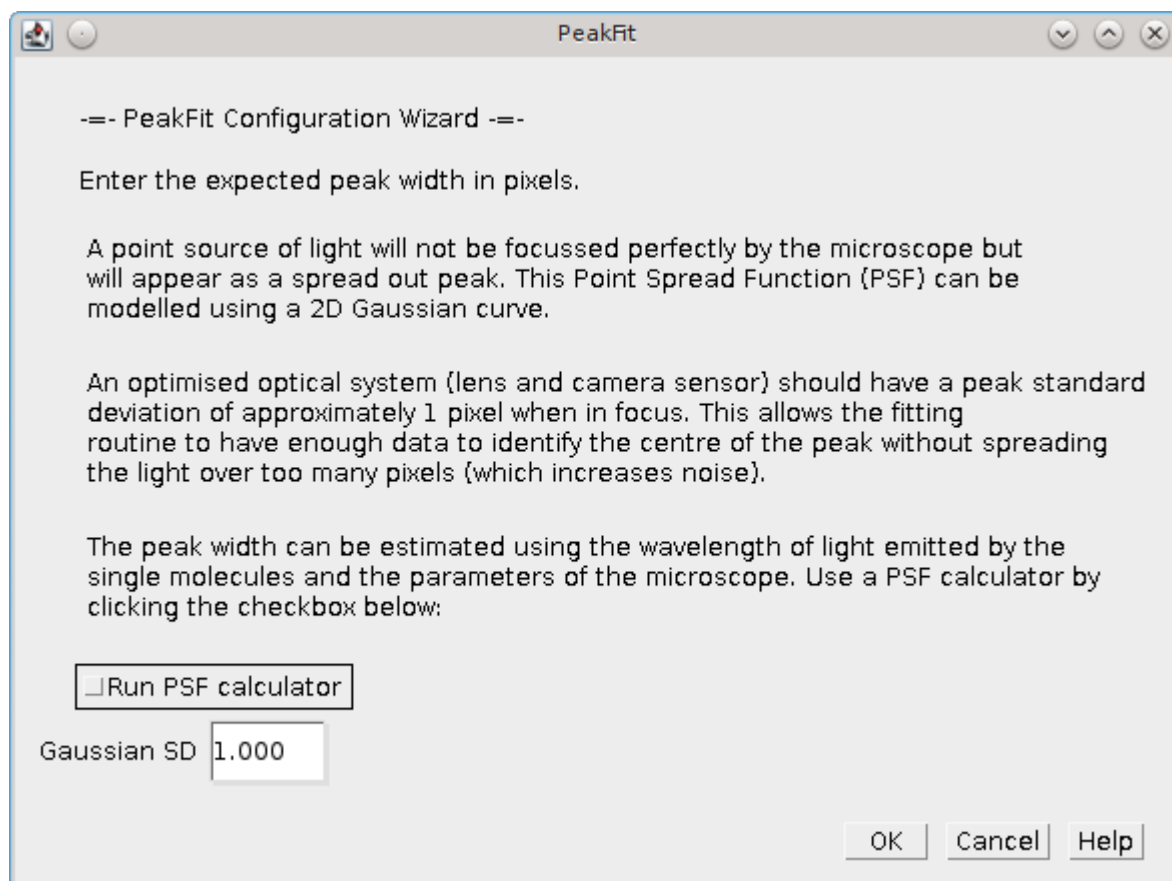
#### 5.1.2.4 Configuration Wizard 4: Exposure Time

The fourth dialog of the wizard requests the exposure time:



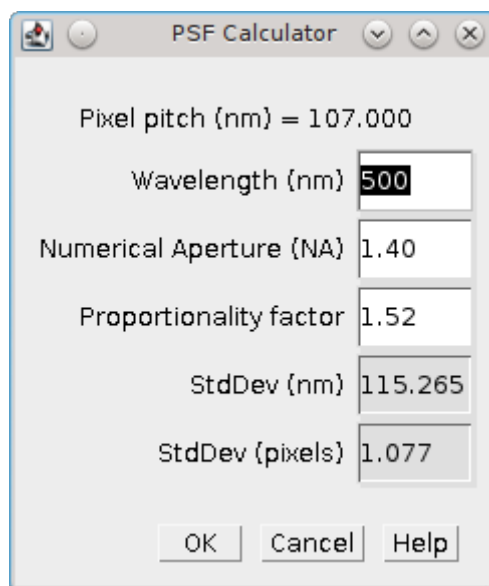
#### 5.1.2.5 Configuration Wizard 5: Peak Width

The fifth dialog of the wizard requests the expected peak width for the 2D Gaussian:





A checkbox is provided which allows the user to run the PSF Calculator. The calculator will compute an expected Gaussian standard deviation using the microscope optics:



The calculator uses the following formula:

$$SD = p \times 1.323 \times \frac{\lambda}{2\pi NA}$$

Where:

SD The standard deviation of the Gaussian approximation to the Airy pattern

$\lambda$  The wavelength (in nm)

NA The Numerical Aperture

p The proportionality factor.

Using a value of 1 gives the theoretical lower bounds on the peak width. However the microscope optics are not perfect and the fluorophore may move on a small scale so the fitted width is often wider than this limit. The factor of 1.52 in the calculator matches the results obtained from the PSF Estimator plugin on many calibration images made with the GDSC optical set-up.

When the configuration wizard is finished the user is presented with the SIMPLE FIT dialog shown in Illustration 4. The settings are saved to the settings file.

If the wizard is cancelled at any point then the settings file is not created. The user will be forced to go through the wizard again the next time they run the plugin.

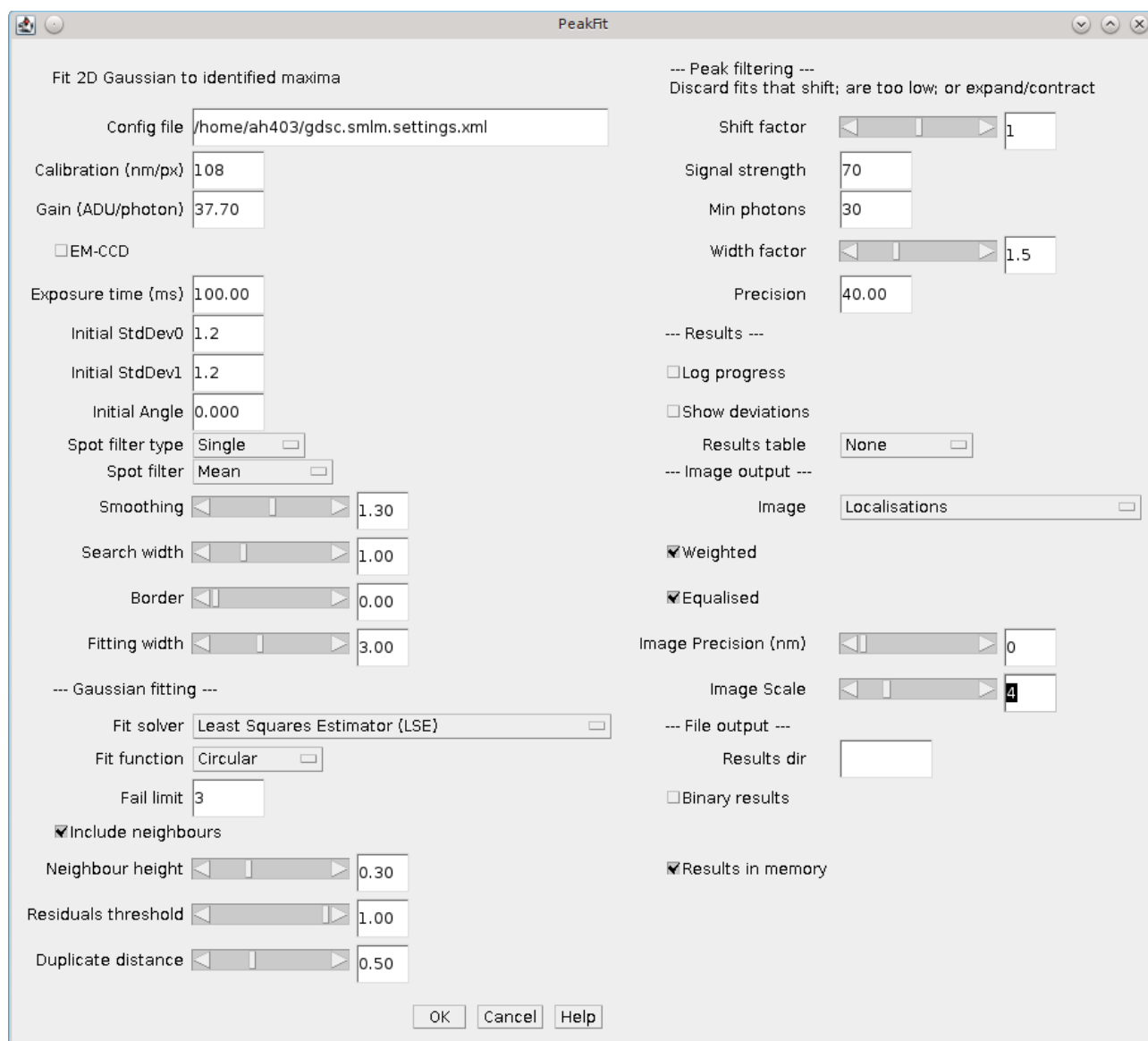
### 5.1.3 Advanced Settings

The SIMPLE FIT plugin is a simplified interface to the PEAK FIT plugin that uses default values for fitting parameters. All the fitting parameters can be adjusted only by using the PEAK FIT plugin. Since the SIMPLE FIT plugin resets the SMLM configuration file to the fitting defaults when the PEAK FIT plugin is run immediately after the SIMPLE FIT plugin the results will be the same. This allows the user to reset the fitting parameters with SIMPLE FIT and then

repeatedly make changes to the parameters with the **PEAK FIT** plugin to see how the results are affected. This can be a useful learning tool to experiment with the fitting parameters.

## 5.2 Peak Fit

Finds all the candidate maxima in an image and fits them using a 2D Gaussian. The **PEAK FIT** dialog is shown in Illustration 5.



*Illustration 5: The **PEAK FIT** dialogue.*

The plugin will initialise using the previously selected configuration file. If no file exists then a default set of settings will be created.

A different file can be selected by double clicking in the text box. This will open a file selection dialogue.

If the file name is changed and the new file exists the plugin will provide the option to reload the settings from the new configuration file.

When the plugin runs all the settings will be saved to the configuration file, overwriting any

existing file.

The dialogue contains setting for the imaging conditions and then various parts of the fitting algorithm: Imaging Calibration; Gaussian PSF; Maxima Identification; Fitting; Multiple Peak Fitting; and Peak Filtering. Additional parameters are used to control the output. Each of the sections is described below.

### Note:

The fitting algorithm is described in section 2.3: Localisation Fitting Method. Understanding the method will ensure that the parameters can be adjusted to achieve the desired fitting result. Using the `SIMPLE FIT` plugin will reset the parameters to their defaults.

### 5.2.1 Imaging Calibration Parameters

The imaging parameters describe the conditions used to acquire the image. The pixel size is used to define distances in nm. The gain is used to convert the Analogue to Digital Units (ADUs) to photons.

| Parameter           | Description  |
|---------------------|--|
| CALIBRATION (NM/PX) | The size of the image pixels in nm   |
| GAIN                | The total gain. Used to convert the camera ADUs to photons   |
| EM-CCD              | Select this if you used an EM-CCD camera. This is stored with the results and used in precision calculations |
| EXPOSURE TIME (MS)  | This is the length of time captured by each frame in milliseconds  |

Note that a bias (offset zero level) is not always needed since the fitting process fits the background which will include the bias offset. The number of photons in the peak can then be calculated without needing to know the camera bias. If Maximum Likelihood fitting is used then the bias is required and the plugin will prompt the user to enter it.

### 5.2.2 Gaussian PSF Parameters

The Point Spread Function (PSF) of the microscope is approximated using a 2D Gaussian function. The Gaussian can have the same width in the X and Y dimensions or separate widths. If the widths are different then the Gaussian will be elliptical in shape. In this case the ellipse can be rotated by an angle. The parameters allow the initial shape of the Gaussian PSF to be specified. These parameters are only an initial guess and the Gaussian shape will be optimised to fit each identified spot in the image.

Note that the Gaussian function is defined in units of pixels.

| Parameter       | Description   |
|-----------------|---|
| INITIAL STDDEV0 | Set the initial parameters for the 2D Gaussian  |
| INITIAL STDDEV1 | Set the initial parameters for the 2D Gaussian. Used for independent X/Y width fitting ( <code>FREE CIRCULAR</code> , <code>FREE</code> ) |

|               |  |
|---------------|--|
| INITIAL ANGLE | Set the initial parameters for the 2D Gaussian. Used for elliptical fitting (FREE) |
|---------------|--|

### 5.2.3 Maxima Identification Parameters

Note that the smoothing, search, border and fitting width parameters are factors applied to the Gaussian function width (INITIAL STDDEV). They have no units.

| Parameter        | Description   |
|------------------|---|
| SPOT FILTER TYPE | <p>The type of filter to use. The default is a SINGLE filter.</p> <p>If a DIFFERENCE or JURY filter is selected then the plugin will present an additional dialogue to configure each additional spot filter.</p> <p>More details are given in the next section.</p>  |
| SPOT FILTER      | <p>The name of the first spot filter:</p> <p>Mean: Compute the mean in a square region. The region can be any size as the final edge pixels are given a weight using the region width</p> <p>Block mean: Compute the mean in a square region. The region is rounded to integer pixels</p> <p>Circular mean: Compute the mean in an approximate circular region. The circle is drawn using square pixels. To see the circle mask use PROCESS &gt; FILTERS &gt; SHOW CIRCULAR MASKS...</p> <p>Gaussian: Perform Gaussian convolution. The convolution kernel standard deviation is set to the SMOOTH parameter. The total region width around each point that is used will be <math>2n+1</math> with <math>n=\lceil 2.8\sigma \rceil</math> where <math>\lceil x \rceil</math> is the ceiling function.</p> <p>Median: Compute the median in a square region. The region is rounded to integer pixels</p> |
| SMOOTHING        | <p>Controls the size of the first smoothing filter:</p> <p><math>SMOOTH = INITIAL\_STDDEV * SMOOTHING</math>.</p> <p>Filtering can be disabled using a SMOOTHING value of 0.</p>  |
| SEARCH WIDTH     | <p>Controls the size of the region used for finding local maxima:</p> <p><math>Width = integer(INITIAL\_STDDEV * SEARCH)</math>.</p> <p>Note that ideally localisation spots should be well separated (over 5 pixels) and so increasing this parameter will reduce the number of false maxima identified as fitting candidates by eliminating noisy pixels.</p>   |

| Parameter     | Description  |
|---------------|--|
| BORDER        | Define the number of border pixels to ignore. No maxima are allowed in the border<br><br>Width = integer(INITIAL STDDEV * FITTING)   |
| FITTING WIDTH | Controls the size of the region used for fitting each peak.<br><br>Width = integer(INITIAL STDDEV * FITTING)<br><br>The width should be large enough to cover a localisation spot so the function can fit the entire spot data. 3 standard deviations should cover 99% of a Gaussian function. |

### 5.2.3.1 Spot Filter Type

The PEAK FIT plugin can perform initial filtering on the image using a SINGLE, DIFFERENCE or JURY filter. The filtered image is then analysed for local maxima. The purpose is to remove noise from the image to prevent identification of false candidate maxima that waste time during the fitting process and may create bad localisation data.

A SINGLE filter will process the image once with the selected filter.

A DIFFERENCE filter will process the image twice with the two configured filters. The second filtered image is then subtracted from the first. This acts as a band-pass filter that allows any frequency between the two filters to pass but remove the other frequencies. For example a PSF with an approximate standard deviation ( $s_0$ ) of 1 could be filtered with a difference of Gaussians filter using a filter standard deviation of 0.5 and 2.

The DIFFERENCE filter is useful when there is a large background variation in the image since the subtraction of the second image is performing a local background subtraction. The spots are then ranked using their relative height over background. This would rank a spot with a height of 10 over a background of 50 as lower than a spot with a height of 30 over a background of 20. The SINGLE filter would put the height 10 spot first as its total height is 60 compared to 50 for the other brighter spot.

A JURY filter will apply many filters to the image. Each filtered image is used to identify maxima. The pixel value from the filtered image from each maxima is added to a sum image. When all filters have been processed the maxima are then identified in the sum image. The JURY filter is therefore finding maxima in a combined image composed of pixels that are maxima. Analysis of simple JURY filters has shown that they have high recall but lower precision than single filters (e.g. a SINGLE Mean with smoothing 1.3 verses a JURY of Mean 1, Mean 2 and Mean 3).

### 5.2.4 Fitting Parameters

| Parameter    | Description  |
|--------------|--|
| FIT SOLVER   | <p>Specify the method used to fit the maxima.</p> <ul style="list-style-type: none"> <li>LSE: Use the GDSC Least Squares Estimator (LSE) fitting algorithm (based on the Hessian matrix of partial derivatives)</li> <li>WLSE: Use the GDSC Weighted Least-Squares Estimation</li> <li>LSEqn: Use the Apache Commons LSE fitting algorithm (based on the Jacobian matrix of partial derivatives)</li> <li>MLE: Use Maximum Likelihood Estimation</li> </ul> <p>Note: The LVM method is fastest, MLE is the most precise (smallest error).</p> <p>Individual FIT SOLVER methods may require further parameters which are detailed in the following section.</p> |
| FIT FUNCTION | <p>Specify the type of 2D Gaussian function to fit.</p> <ul style="list-style-type: none"> <li>FIXED: Fits X,Y centre and amplitude</li> <li>CIRCULAR: Fits X,Y centre, combined X/Y deviation and amplitude</li> <li>FREE CIRCULAR: Fits X,Y centre, individual X/Y deviation and amplitude</li> <li>FREE: Fits X,Y centre, individual X/Y deviation, rotation angle and amplitude</li> </ul>   |
| FAIL LIMIT   | <p>Stop processing the image frame when N consecutive fits fail. This prevents attempting to fit the remaining candidates that have a lower signal.</p>  |

#### 5.2.4.1 Gaussian PSF Equation

The following equation specifies the elliptical 2D Gaussian (named FREE in the FIT FUNCTION options):

$$f(x, y) = B + \frac{\text{Signal}}{2\pi\sigma_x\sigma_y} e^{-a(x-x_0)^2 - 2b(x-x_0)(y-y_0) + c(y-y_0)^2}$$

where

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

is positive-definite and

$$a = \frac{\cos^2 \theta}{2 \sigma_x^2} + \frac{\sin^2 \theta}{2 \sigma_y^2}$$

$$b = -\frac{\sin^2 2\theta}{4 \sigma_x^2} + \frac{\sin^2 2\theta}{4 \sigma_y^2}$$

$$c = \frac{\sin^2 \theta}{2 \sigma_x^2} + \frac{\cos^2 \theta}{2 \sigma_y^2}$$

with

|               |                                      |
|---------------|--------------------------------------|
| B             | The background level                 |
| <i>Signal</i> | The total volume of the Gaussian     |
| $x_0$         | The X centre of the Gaussian         |
| $y_0$         | The Y centre of the Gaussian         |
| $\sigma_x$    | The X standard deviation             |
| $\sigma_y$    | The Y standard deviation             |
| $\theta$      | The angle of rotation of the ellipse |

If  $\theta$  is zero this reduces to an elliptical 2D Gaussian with no rotation (named `FREE CIRCULAR` in the `FIT FUNCTION` options):

$$f(x, y) = B + \frac{Signal}{2\pi \sigma_x \sigma_y} e^{-\left(\frac{(x-x_0)^2}{2\sigma_x^2}\right) - \left(\frac{(y-y_0)^2}{2\sigma_y^2}\right)}$$

If  $\theta$  is zero and  $\sigma_x$  equals  $\sigma_y$  this reduces to a circular 2D Gaussian:

$$f(x, y) = B + \frac{Signal}{2\pi \sigma^2} e^{-\left(\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma^2}\right)}$$

Note: A previous of the software did not fit the Gaussian volume (Signal) but the height of the Gaussian (Amplitude). To convert the Signal to the Amplitude use the following conversion:

$$Amplitude = \frac{Signal}{2\pi \sigma_x \sigma_y}$$

#### 5.2.4.2 Least Squares Estimation

Least-squares estimation is the processes of fitting a function (expected values) to a set of observed values. The fit attempts to minimise the sum of the squared difference between observed (O) and expected (E):

$$SS = \sum (O - E)^2$$

The parameters for the expected function are updated until no improvement can be made. The estimation process uses the popular Levenberg-Marquardt (LVM) algorithm which uses the gradient of the function, i.e. how the function value will change with a change to the parameters, to choose how to modify the parameters. The `PEAK FIT` plugin offers two version of the algorithm. One version uses the Hessian matrix of partial derivatives (i.e. it

is a Full Newton method for finding roots of equations). This is a matrix of the gradient of the function with respect to two parameters for all combinations of parameters. The method must compute the matrix and invert it and this can lead to numerical instability with floating-point numbers, for example if one parameter has a very small gradient with respect to any of the other parameters. The second uses only the Jacobian matrix of partial derivatives. This is a matrix of the gradient with respect to each parameter at each data point. The Jacobian is used to approximate and update the inverted Hessian (i.e. it is a quasi-Newton method) and is less prone to instability. This method is provided by an external library (Apache Commons Maths). The custom made GDSC routines are faster and both methods return very similar results.

The weighted least-squares estimator provides a weight for each point using the estimated value. The fit attempts to minimise:

$$SS = \sum (O - E)^2 / E^2$$

Note that when the estimated value is small the weight will have a destabilising effect on the sum by significantly over-weighting data. The weight is thus limited to a minimum of 1. However weighted least squares is not recommended as it has not been proven in testing on simulated single molecule data to outperform the standard LSE.

Currently the quasi-Newton LSE requires no additional parameters to be configured.

The GDSC Least Squares Estimator requires the following additional parameters:

| Parameter    | Description   |
|--------------|---|
| FIT CRITERIA | <p>The fit uses a non-linear least squares routine until convergence. If convergence is not achieved by the maximum iterations then the fit fails. Convergence can be defined using:</p> <ul style="list-style-type: none"> <li>Least squared error: convergence of the least-squared error of the fit to a given number of SIGNIFICANT DIGITS</li> <li>Least squared error plus: convergence of the least-squared error of the fit to a given number of SIGNIFICANT DIGITS. Convergence is also achieved if 3 consecutive improvements are the same relative improvement and half of the maximum iterations has passed. This avoids slowly converging fits</li> <li>Co-ordinates: convergence of the X,Y centre coordinates to within a specified COORD DELTA</li> <li>Parameters: convergence of each of the Gaussian parameters (e.g. centre, width and signal) to a set number of SIGNIFICANT DIGITS</li> </ul> |



| Parameter          | Description   |
|--------------------|---|
| SIGNIFICANT DIGITS | When comparing two numbers, defines the significant digit for equality, e.g. specify 3 to recognise 1 and 1.001 as equal.<br><br>Note that Java floating points have a limited precision preventing comparisons to high numbers of significant digits. Using a value over 15 will generally not allow convergence   |
| COORD DELTA        | Define the smallest shift (in pixels) that specifies a movement of the X/Y coordinates. Used in the COORDINATES fit criteria  |
| LAMBDA             | The initial lambda for the Levenberg-Marquardt algorithm. Higher favours the gradients of the parameters. Lower favours the Hessian matrix gradients (second partial derivatives). Lower is used when very close to the solution.<br><br>Note that the algorithm updates the lambda during fitting to refine the improvement to the fit. A value of 10 is a good initial value. |
| MAX ITERATIONS     | Stop the fit when this is reached and return a failure  |

#### 5.2.4.3 Maximum Likelihood Estimation

Maximum Likelihood estimation is the processes of fitting a function (expected values) to a set of observed values by maximising the probability of the observed values. MLE requires that there is a probability model for each data point. The function is used to predict the expected value (E) of the data point and the probability model is used to specify the probability (likelihood) of the observed value (O) given the expected value. The total probability is computed by multiplying all the probabilities for all points together:

$$likelihood = \prod p(O|E)$$

or by summing their logarithms:

$$loglikelihood = \sum \ln(p(O|E))$$

The maximum likelihood returns the fit that is the most probable given the model for the data.

#### Poisson noise model

This model is suitable for modelling objects with a lot of signal.

The standard model for the image data is a Poisson model. This models the fluctuation of light emitted from a light source (photon shot noise). This is based on the fact that gaps between individual photons can vary even though the average emission rate of the photons is constant. The Poisson model will work when the amount of shot noise is much higher than all other noise in the data, i.e. when the localisations are very bright. If the other noise is significant then a more detailed model is required.

The Poisson probability model is:

$$p(k|\lambda) = \frac{\lambda^k}{k!} e^{-\lambda}$$

with  $k$  equal to the pixel count and  $\lambda$  equal to the expected pixel count. Note that when we take a logarithm of this we can remove the factorial since it is constant and will not affect optimising the sum:

$$\begin{aligned} \ln(p(k|\lambda)) &= \ln\left(\frac{\lambda^k}{k!} e^{-\lambda}\right) \\ &= \ln\left(\frac{\lambda^k}{k!}\right) + \ln(e^{-\lambda}) \\ &= \ln(\lambda^k) - \ln(k!) - \lambda \\ &= k \ln(\lambda) - \lambda \end{aligned}$$

The final log-likelihood function is fast to evaluate and since it can be differentiated the formula can be used with derivative based function solvers.

### Poisson-Gaussian noise model

This model is suitable for modelling a standard CCD camera.

This model accounts for the photon shot noise and the read noise of the camera, i.e. when the number of electrons is read from the camera chip there can be mistakes. The read noise is normally distributed with a mean of zero.

The two noise distributions can be combined by convolution of a Poisson and a Gaussian function to produce the following model:

$$p(k|\lambda, \sigma) = \sum_{n=0}^{\infty} \frac{1}{n!} \lambda^n e^{-\lambda} \times \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(k-n)^2}$$

with  $k$  equal to the pixel count,  $\lambda$  equal to the expected pixel count and  $\sigma$  equal to the standard deviation of the Gaussian read noise. This model is evaluated using a saddle-point approximation as described in Snyder *et al* (1995); the implementation is adapted from the authors example source code.

No gradient is available for the function and so non-derivative based methods must be used during fitting.

### Poisson-Gamma-Gaussian noise model

This model is suitable for modelling a Electron Multiplying (EM) CCD camera.

This model accounts for the photon shot noise, the electron multiplication gain of the EM register and the read noise of the camera. The EM-gain is modelled using a Gamma distribution. The read noise is normally distributed with a mean of zero.

The convolution of the Poisson and Gamma distribution can be expressed as:

$$G_{p,m}(c) = e^{-p} \delta(c) + \sqrt{\frac{p}{cm}} e^{-\frac{c}{m}-p} \text{Bessel}I_1\left(2\sqrt{\frac{cp}{m}}\right)$$

where:

$p$                       The average number of photons

|              |   |
|--------------|---|
| m            | The EM-gain multiplication factor                     |
| c            | The observed pixel count                              |
| $\delta(c)$  | The Dirac delta function (1 when $c=0$ , 0 otherwise) |
| BesselI1     | Modified Bessel function of the 1 <sup>st</sup> kind  |
| $G_{p,m}(c)$ | The probability of observing the pixel count c        |

This is taken from Ulbrich and Isacoff (2007). The output of this function is subsequently convolved with a Gaussian function with standard deviation equal to the camera read noise and mean zero. This must be done numerically since no algebraic solution exists. However Mortensen et al (2010) provide example Python code that computes an approximation to the full convolution using the Error function to model the cumulative Gaussian distribution applied to the Poisson-Gamma convolution at low pixel counts. This approximation closely matches the correct convolution with a Gaussian but is faster to compute.

No gradient is available for the function and so non-derivative based methods must be used during fitting.

## Parameters

Since MLE requires that we know the value of the data at each point the **PEAK FIT** plugin requires the camera bias. This is subtracted from the data before fitting so that the probability model can accurately return the likelihood of the pixel values given the expected values from the function.

The Maximum Likelihood Estimator requires the following additional parameters:

| Parameter          | Description   |
|--------------------|---|
| CAMERA BIAS        | The value added to all pixels by the camera   |
| MODEL CAMERA NOISE | Select this option to model the camera noise (read noise and EM-gain (if applicable)). If unselected the MLE will use the Poisson noise model.  |
| READ NOISE         | The camera read noise (in ADUs). Only applicable if using MODEL CAMERA NOISE.   |
| GAIN               | The total camera gain. Only applicable if using MODEL CAMERA NOISE.   |
| EM-CCD             | Select this if using an EM-CCD camera. The Poisson-Gamma-Gaussian function will be used to model camera noise. The alternative is the Poisson-Gaussian function. Only applicable if using MODEL CAMERA NOISE. |

| Parameter                | Description   |
|--------------------------|---|
| SEARCH METHOD            | The search method to use. It is recommended to use the Powell algorithm for any model. The BFGS algorithm is a good alternative for the Poisson noise model.<br><br>SEARCH METHODS are detailed in the following section. |
| RELATIVE THRESHOLD       | Define the relative threshold for convergence   |
| ABSOLUTION THRESHOLD     | Define the absolute threshold for convergence   |
| MAX ITERATIONS           | The maximum number of iterations. Note that in contrast to the least-squares estimators the function is evaluated multiple times per iteration  |
| MAX FUNCTION EVALUATIONS | The maximum number of times to evaluate the model   |

#### 5.2.4.4 Search Methods

Brief notes on the different algorithms and where to find more information are shown below for completeness. It is recommended to use the Powell or BFGS methods.

Note that some algorithms support a bounded search. This is a way to constrain the values for the parameters to a range, for example keep the XY coordinates of the localisation within the pixel region used for fitting. When using a bounded search the bounds are set at the following limits:

The lower bounds on the background and signal are set at zero. The upper bounds are set at the maximum pixel value for the background and twice the sum of the data for the signal.

The coordinates are limited to the range of the fitted data.

The width is allowed to change by a value of 2-fold from the initial standard deviation.

#### Powell

Search using Powell's conjugate direction method.

This method does not require derivatives. It is the recommended method for the camera noise models.

[http://en.wikipedia.org/wiki/Powell%27s\\_method](http://en.wikipedia.org/wiki/Powell%27s_method)

#### Powell (bounded)

Search using Powell's conjugate direction method using a mapping adapter to ensure a bounded search.

This method maps the parameters from a bounded space to infinite space and then uses the Powell method.

## BOBYQA

Search using Powell's Bound Optimisation BY Quadratic Approximation (BOBYQA) algorithm.

BOBYQA could also be considered as a replacement of any derivative-based optimiser when the derivatives are approximated by finite differences. This is a bounded search.

This method does not require derivatives.

<http://en.wikipedia.org/wiki/BOBYQA>

## CMAES

Search using active Covariance Matrix Adaptation Evolution Strategy (CMA-ES). The CMA-ES is a reliable stochastic optimization method which should be applied if derivative-based methods, e.g. conjugate gradient, fail due to a rugged search landscape. This is a bounded search and does not require derivatives.

<http://en.wikipedia.org/wiki/CMA-ES>

## Conjugate Gradient

Search using a non-linear conjugate gradient optimiser. Two variants are provided for the update of the search direction: Fletcher-Reeves and Polak-Ribière, the later is the preferred option due to improved convergence properties.

This is a bounded search using simple truncation of coordinates at the bounds of the search space. Note that this method has poor robustness (fails to converge) on test data and is not recommended.

This method requires derivatives.

[http://en.wikipedia.org/wiki/Conjugate\\_gradient\\_method](http://en.wikipedia.org/wiki/Conjugate_gradient_method)

## BFGS

Search using a Broyden-Fletcher-Goldfarb-Shanno (BFGS) gradient optimiser.

This method requires derivatives. This is a good alternative to the Powell method for the Poisson noise model.

[http://en.wikipedia.org/wiki/Broyden%E2%80%93Fletcher%E2%80%93Goldfarb%E2%80%93Shanno\\_algorithm](http://en.wikipedia.org/wiki/Broyden%E2%80%93Fletcher%E2%80%93Goldfarb%E2%80%93Shanno_algorithm)

### 5.2.4.5 Which Fit Solver to Choose

The most general fit solver is the least-squares estimator. It does not require any specific information about the camera to perform fitting. For this reason this is the default fitting engine. It is also the fastest method.

Maximum Likelihood estimation should return a solution that is more precise than least-squares estimation, i.e. has less variation between the fitted result and the actual answer. MLE should be operating at the theoretical limit for fitting given how much information is actually present in the pixels. This limit is the Cramér-Rao lower bound which expresses a lower bound on the variance of estimators of a deterministic parameter. MLE has also been proven to be robust to the position of the localisation within the pixel whereas least-

squares estimation is less precise the further the localisation is from the pixel centre [Abraham *et al* (2009)]. Therefore MLE should be used if you would like the best possible fitting. However it requires additional parameters which if configured incorrectly will lead to fitting results that are not as precise as the least-squares estimators.

If you are fitting localisations with a high signal then the Poisson model will work. This model also has the advantage of requiring only the camera bias. If unknown then this can be guessed from a low-light image using the darkest part of the frame. The Poisson model allows use of the derivative-based BFGS algorithm and is fast.

At low signal levels other sources of noise beyond shot noise become more significant and the fitting will produce better results if they are included in the model. This would mean choosing the Poisson-Gaussian function (slow) for a standard CCD camera or the Poisson-Gamma-Gaussian function (very slow) for an EM-CCD camera.

Methods for determining the bias, read noise and gain of a camera can be found in the sections describing the Mean-Variance Test and The plugin provides a plot of gain verses slice and a histogram of the values. These can be used to determine if the gain is constant throughout the image and so is a good estimate. plugins.

### 5.2.5 Multiple Peak Fitting Parameters

These parameters control how the algorithm handles fitting high density localisations where the region surrounding a maxima used for fitting may contain other maxima. If only the single target peak is fitted it is possible the fitting engine will move the Gaussian centre to the higher peak (causing a duplicate fit). This can be avoided by fitting multiple peaks but with the penalty of increased computation time and likelihood of failure.

In the event that multiple fitting fails the algorithm reverts to fitting a single peak. In this case duplicates fits can be eliminated using the `DUPLICATE_DISTANCE` parameter.

Note that peaks are processed in height order. Thus any candidate maxima with neighbours that are higher will be able to use the exact fit parameters of the neighbour. If they are not available then fitting of the neighbour failed. In this case, as with lower neighbour peaks, the initial parameters for the neighbour are estimated.

| Parameter          | Description  |
|--------------------|--|
| INCLUDE_NEIGHBOURS | Include any neighbour peaks within the fitting region in the fit (multiple peak fitting) |

| Parameter           | Description   |
|---------------------|---|
| NEIGHBOUR HEIGHT    | <p>Define the height for a neighbour peak to be included as a fraction of the peak to be fitted. The height is taken relative to an estimate of the background value of the image (the image mean). If the target peak is below the background then only higher neighbour peaks are included.</p> <p>Since neighbours that are higher than the maxima may cause the fit procedure to drift to a different position this setting allows higher peaks to be included and lower neighbour peaks to be ignored.</p> <p>A value of 1 will only include peaks higher than the target peak. A value of 0 will include all neighbours</p>   |
| RESIDUALS THRESHOLD | <p>Set a threshold for refitting a single peak as a double peak. A value of 1 disables this feature.</p> <p>The residuals (difference of the fitted function to the data) are analysed to determine if there is a skewed arrangement around the centre point. The skew is calculated by dividing the region into quadrants (clockwise labelled ABCD), summing each quadrant and then calculating the difference of opposite quadrants divided by the sum of the absolute residuals: <math> A-B+C-D  / \sum( r )</math>. If this value is zero then the residuals are evenly spread in each quadrant. If it is one then the residuals are entirely above zero in one pair of opposing quadrants and below zero in the other, i.e. the spot is not circular and may be a doublet (two spots close together).</p> <p>If the residuals analysis is above the threshold then it is refitted as a doublet. The doublet fit is compared to the single fit and only selected if the fit is significantly improved.</p> <p>Note the residuals threshold only controls when doublet fitting is performed and not the selection of a doublet over a single. Lowering the threshold will increase computation time.</p> |
| DUPLICATE DISTANCE  | <p>Each new fit is compared to the current results for the frame. If any existing fits are within this distance then the fit is discarded. This avoids duplicate results when multiple peak fitting has refit an existing result peak.</p> <p>Note that doublets are allowed to be closer than this distance since the results of the latest fitting are only compared to all existing results.</p>   |

## 5.2.6 Filtering Parameters

These parameters control the fitted peaks that will be discarded.

| Parameter        | Description  |
|------------------|--|
| SHIFT FACTOR     | Any peak that shifts more than a factor of the initial peak standard deviation is discarded  |
| SIGNAL STRENGTH  | Any peak with a signal/noise below this level is discarded.<br>The signal is the calculated volume under the Gaussian.<br>The image noise is calculated per frame using the least mean of squares of the image residuals. This is a method that returns a value close to the image standard deviation but is robust to outliers. Note: The noise method can be changed using the extra options available by holding the <code>SHIFT</code> key when running the plugin |
| MIN PHOTONS      | The minimum number of photons in a fitted peak. This requires a correctly calibrated gain to convert ADUs to photons.  |
| MIN WIDTH FACTOR | Any peak whose final fitted width is a factor smaller than the start width is discarded (e.g. 0.5x fold)   |
| MAX WIDTH FACTOR | Any peak whose final fitted width is a factor larger than the start width is discarded (e.g. 2x fold)  |
| PRECISION        | Any peak with a precision above this level is discarded, i.e. not very good precision  |

If a precision threshold is specified then the plugin will calculate the precision of the localisation using the Mortensen *et al* (2010) formula, see Appendix A : Localisation Precision. The appropriate formula for either the Maximum Likelihood and Least Squares Estimator is used.

The precision calculation requires the expected background noise at each pixel. The noise can be estimated two ways. The first method is to use the noise estimate for the entire frame. This is computed automatically during fitting for each frame or can be provided using the additional options (see section below). The second is to use the local background level that is computed when fitting the localisation. This background level is the background number of photons at the localisation position that will contribute photon shot noise to the pixels. The global noise estimate will be a composite of the average photon shot noise over the entire frame and the read noise of the camera. The local background will provide more contextual information about the localisation precision and may be preferred if fitting localisations where the image background is highly variable.

If using precision filtering the plugin will ask the user if they wish to perform the calculation using the local background. If using a local background then the camera bias must be provided so that the background photons can be correctly determined.

## 5.2.7 Results Parameters

Control where the list of localisations will be recorded. Parameters have been grouped in the following table by background colour for different outputs: Table; Image; File; and Memory.



| Parameter            | Description  |
|----------------------|--|
| LOG PROGRESS         | Outputs a lot of logging information to the ImageJ log window. Used for debugging the fitting algorithm. Logging slows down the program and should normally be disabled.   |
| SHOW DEVIATIONS      | <p>Calculate the estimated deviations for the fitted parameters. These are shown in the table output and saved to the results files.</p> <p>Note that the deviations are not used for filtering bad fits so should be disabled to improve speed performance.</p>   |
| <i>Table Results</i> |  |
| RESULTS TABLE        | <p>Show all the fitting results in an ImageJ result table:</p> <ul style="list-style-type: none"> <li>• None: No results table</li> <li>• Calibrated: Output the result positions and widths in nm and values in photons</li> <li>• Uncalibrated: Output the result positions and widths in pixels and values in ADUs</li> </ul> |
| <i>Image Results</i> |  |

| Parameter           | Description  |
|---------------------|--|
| IMAGE               | <p>Show a reconstructed image using the localisations:</p> <ul style="list-style-type: none"> <li>• None: No image</li> <li>• Localisations: Use a value of 1 for each spot</li> <li>• Signal intensity: Use the signal strength for each spot</li> <li>• Frame number: Use the frame number for each spot</li> <li>• PSF: Plot the fitted Gaussian PSF</li> <li>• Localisations (width=precision): Plot a Gaussian at the centre of each spot using the localisation precision for the width</li> <li>• Signal (width=precision): Plot a Gaussian at the centre of each spot using the localisation precision for the standard deviation and the signal intensity for the height</li> <li>• Localisations (width=av.precision): Plot a Gaussian at the centre of each spot using the configured image precision for the width</li> <li>• Signal (width=av.precision): Plot a Gaussian at the centre of each spot using the configured image precision for the standard deviation and the signal intensity for the height</li> <li>• Fit error: Use the fitting error for each spot</li> </ul> |
| WEIGHTED            | <p>If selected the exact spot coordinates are used to distribute the value on the surrounding 2x2 integer pixel grid using bilinear weighting.</p> <p>If not selected the spot is plotted on the nearest pixel.</p>  |
| EQUALISED           | Use histogram equalisation on the image to enhance contrast. Allows viewing large dynamic range images.  |
| IMAGE PRECISION     | The Gaussian standard deviation to use for the average precision plotting options  |
| IMAGE SCALE         | The factor used to enlarge the image   |
| <b>File Results</b> |  |
| RESULTS DIR         | <p>The directory used to save the results. The result file will be named using the input image title plus a suffix.</p> <p>Leave empty for no results file.</p>  |

| Parameter             | Description  |
|-----------------------|--|
| BINARY RESULTS        | <p>If selected save the results in binary format. The suffix is .results.bin.</p> <p>If not selected save the results in text format. The suffix is .results.xls. The results are tab delimited and can be opened in a spreadsheet application.</p> <p>Saving the results in binary format provides very fast read and write performance. It is preferred when using large datasets. The data can be read using the RESULTS MANAGER plugin.</p>    |
| <i>Memory Results</i> |  |
| RESULTS IN MEMORY     | <p>Store all results in memory.</p> <p>This is very fast and is the default option applied when no other results outputs are chosen (preventing the loss of results). Results in memory can be accessed by other plugins, for example the RESULT MANAGER can convert them to file or images.</p> <p>The memory results will be named using the input image title. If a results set already exists with the same name then it will be replaced.</p> |

## 5.2.8 Interactive Results Table

The results table will show the coordinates and frame for each localisation. To assist in viewing the localisations the table supports mouse click interaction. If the original source image is open in ImageJ the table can draw ROI points on the image:

- Double-clicking a line in the results table will draw a single point overlay on the frame and at the coordinates identified
- Highlighting multiple lines with a mouse click while holding the shift key will draw multiple point overlay on the coordinates identified. Each point will only be displayed on the relevant frame in the image. The frame will be set to the first identified frame in the selection

The coordinates for each point are taken from the X & Y columns for the fitted centre (not the original candidate maxima position).

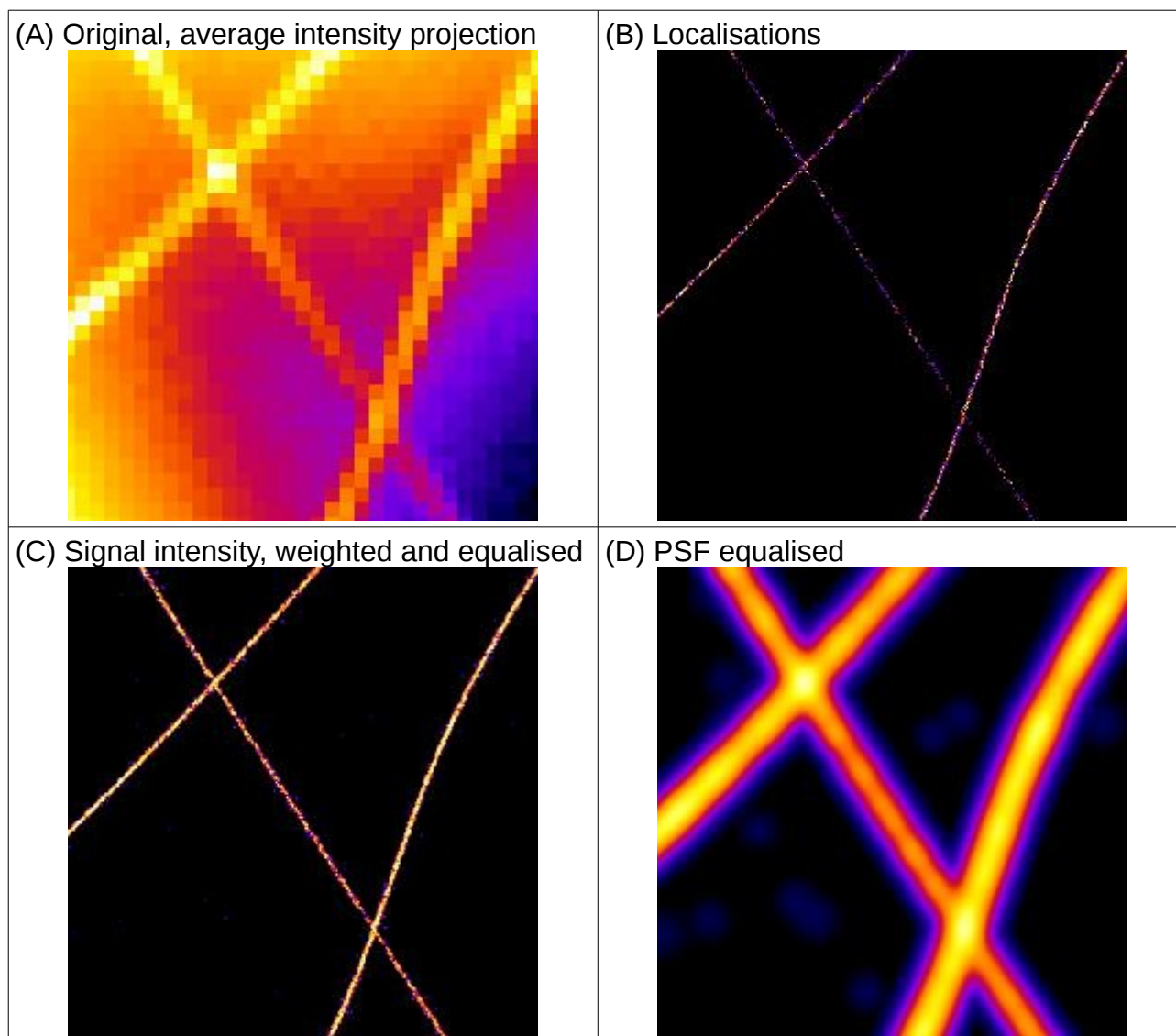
## 5.2.9 Live image display

The super-resolution image is computed in memory and displayed live during the fitting process. To reduce the work load on ImageJ the displayed image is updated at set intervals as more results become available. The image is initially created using a blank frame; the size is defined by the input image. The image is first drawn when 20 localisations have been recorded. The image is then redrawn each time the number of localisations increases by 10%. Finally the image is redrawn when the fitting process is complete.

### 5.2.10 Image examples

Illustration 6 shows examples of different image rendering methods. The `LOCALISATIONS` and `SIGNAL INTENSITY` methods are able to plot the location of the fibres to a higher resolution than the original average intensity projection. The `POINT SPREAD FUNCTION (PSF)` plot shows a very similar width for the fibres as the original image. However there has been a significant reduction in background noise since any signals not identified as a localisation are removed.

The `LOCALISATIONS` image method can be used to directly count localisations in an area, for example counting localisations in regions of a cell. This is only valid if the image has not been rendered using equalisation since that adjusts the pixels values to increase contrast. A region can be marked on the image using any of ImageJ's area ROI tools. The localisation count can be measured by summing the pixel intensity in the region. This is performed using the `ANALYZE > MEASURE` command (`CTRL+M`). Note: Ensure that the `INTEGRATED DENSITY` measurement is selected in the ImageJ `ANALYZE > SET MEASUREMENTS...` dialogue.



*Illustration 6: Example super-resolution images using different rendering methods. Images were generated from a sequence of 2401 frames using the Tubulins 1 dataset from the Localisation Microscopy Challenge 2013 (<http://bigwww.epfl.ch/smlm/challenge/>). The original image has been enlarged using 8x magnification and part of the image has been extracted using a region of 256x256 pixels at point (x=1348,y=1002). The region contains 3855 localisations.*

### 5.2.11 Running Peak Fit

When the plugin is run it will process the image using multi-threaded code. Each frame will be added to a queue and then processed when the next worker is free. The number of workers is configured using the ImageJ preferences: EDIT > OPTIONS > MEMORY & THREADS. The PARALLEL THREADS FOR STACKS parameter controls the number of threads. Note that the image is not processed using ImageJ's standard multi-threaded plugin architecture for processing stacks. The SMLM fitting engine code is written so it can run outside of ImageJ as a Java library. The plugin just uses the configured ImageJ parameter for the thread count.

Progress is shown on the ImageJ progress bar. The plugin can be stopped using the `ESCAPE` key. If stopped early the plugin will still correctly close any open output files and the partial results will be saved.

### 5.2.12 Additional Fitting Options

The standard `PEAK FIT` plugin allows the user to set all the parameters that control the fitting algorithm. However there are some additional options (disabled by default) that provide extra functionality. These can be set by running the `PEAK FIT` plugin with the `SHIFT` or `ALT` key down. This will add some extra fields to the plugin dialogue:

| Parameter        | Description  |
|------------------|--|
| INTERLACED DATA  | Select this option if the localisations only occur in some of the image frames, for example in the case where 2 channel imaging was performed or alternating white light and localisation imaging. If selected the program will ask for additional parameters to specify which frames to include in the analysis (see section below).  |
| INTEGRATE FRAMES | Combine N consecutive frames into a single frame for fitting. This allows the <code>PEAK FIT</code> plugin to simulate the result of running the image acquisition at a slower frame rate (exposure time).<br><br>The results will be slightly different from a long exposure experiment due to the cumulative read noise of multiple frames differing from the read noise of a single long exposure frame.<br><br>Note that the results will be entered into the results table with a start and end frame representing all the frames that were integrated. |
| MIN ITERATIONS   | The minimum number of iterations to run the fitting optimiser. Prevents the optimiser stopping too soon but may prevent the optimiser recognising convergence since later iterations may not improve the fit. An experimental feature that should be used with caution.  |
| NOISE            | Set a constant noise for all frames in the image. This overrides the per-frame noise calculation in the default mode.  |
| NOISE METHOD     | Specify the method used to calculate the noise. See section 11.5 (Noise Estimator plugin) for details of the methods.  |
| IMAGE WINDOW     | Applies to output images.<br><br>The Image Window specifies the number of consecutive frames from the results that should be plotted on a single ImageJ stack frame.<br><br>By default this parameter is zero. All localisations are plotted on the same output frame.   |

|                       |   |
|-----------------------|---|
|                       | <p>If this is set to 1 then each frame will be output to a new frame in the output image. Use this option to allow the input and output images to be directly compared frame-by-frame.</p> <p>If set higher than 1 then N frames will be collated together into one output image. Use this option to produce a time-slice stack through your results at a specified collation interval.</p> <p>This option is not recommended during live fitting since the results must be sorted. This is not possible with multi-threaded code and the results can appear out of order. In this case any result that is part of a frame that has already been drawn will be ignored.</p> <p>The option is also available using the RESULTS MANAGER plugin which can plot all results in order.</p> |
| SHOW PROCESSED FRAMES | <p>Show a new image stack (labelled 'Processed frames') that show the images that were passed to the fitting engine.</p> <p>This option is useful when using the INTERLACED DATA or INTEGRATE FRAMES options. Each slice will be labelled with the start and end frame of the original image used to produce the image data.</p>  |

### 5.2.13 Interlaced Data

The additional fitting options allow for interlaced data where not all the frames in the image should be analysed. Interlaced data must follow a regular pattern where a repeating block of frames should be processed followed by a block of frames to ignore. The plugin must know the size of each block and the first frame that must be precessed. If the INTERLACED DATA option is selected then an addition dialogue will be shown:

Illustration 7: Peak Fit Interlaced Data dialog

| Parameter | Description   |
|-----------|---|
| START     | The first frame containing data                                       |
| BLOCK     | The number of continuous frames that contain data                     |
| SKIP      | The number of continuous frames to skip before the next block of data |

The INTERLACED DATA option is fully compatible with the INTEGRATE FRAMES option. However note that the data is read from the interlaced frames and then aggregated. None of the skipped frames will be aggregated. The user must simply select how many consecutive data frames to integrate.

The use of the interlaced and integrate options together can produce results that have a larger gap between the start and end frame than the number of frames that were integrated. For example if the plugin is set to fit 2 out of 3 frames but integrate 4 frames then any fit results from the first processed image will have a start frame of 1 and an end frame of 5.

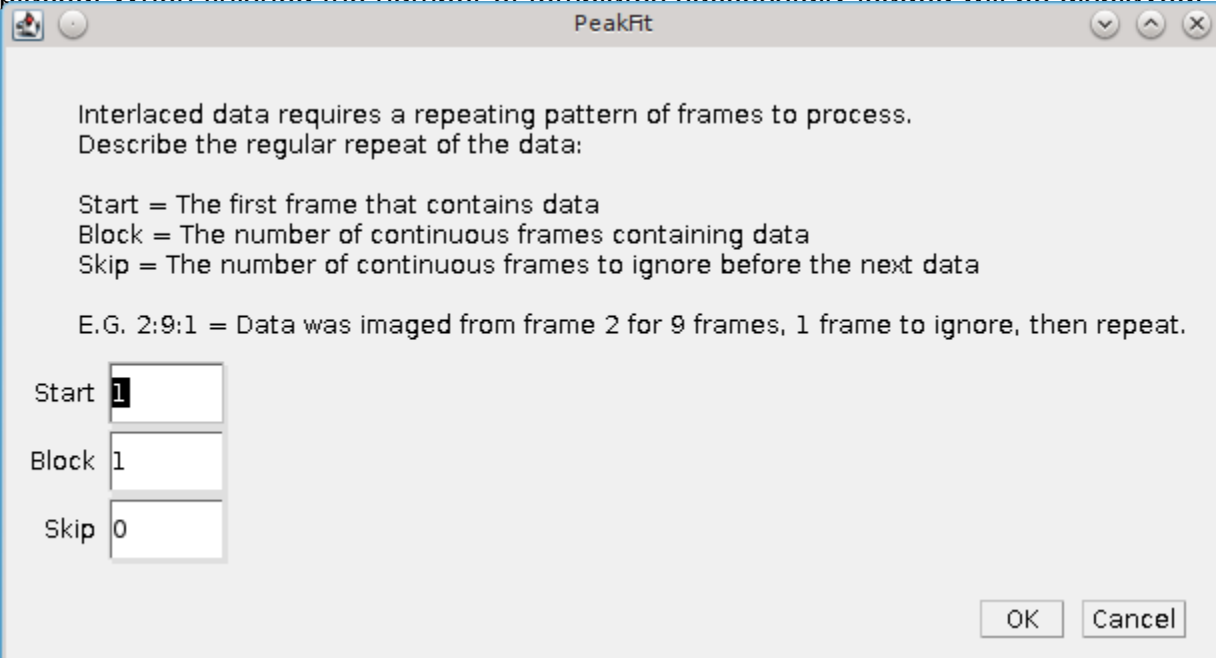
### 5.3 Template Configuration

Allows the user to load custom configuration templates from a directory.

The GDSC SMLM configuration is stored in a configuration file. This file is used by many of the plugins to save and load settings. Copies of this file can be edited and saved as templates to a directory. This allows settings to be applied from pre-configured templates within various plugins, for example the fitting configuration of the PEAK FIT plugin.

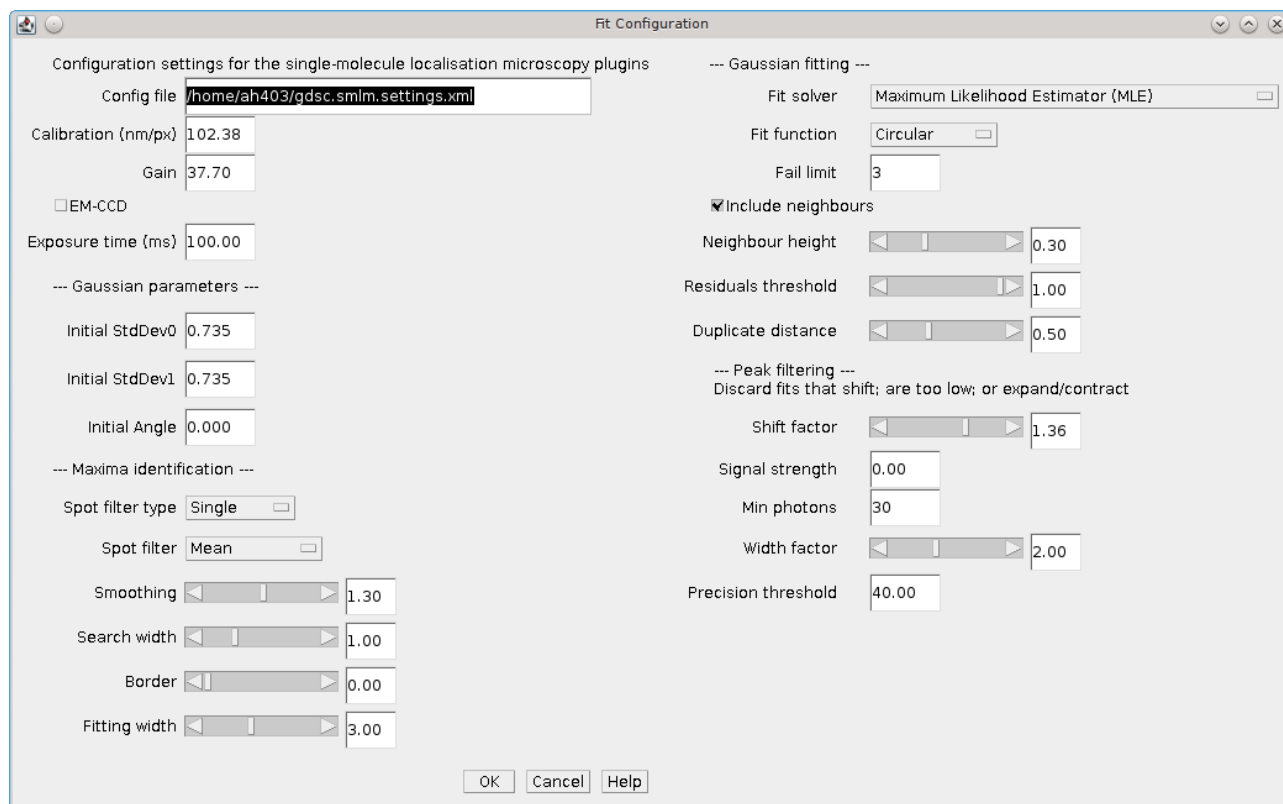
To load a set of saved templates run the TEMPLATE CONFIGURATION plugin. The user will be asked to select a template directory. By default this is a directory inside the user's home directory named 'gdsc.smlm'.

When the directory is selected all files with the XML extension will be processed. If the file can be successfully loaded as a GDSC SMLM settings file then the template will be added to the list of available templates. The template will be named using the filename without the .xml file extension. Any existing templates with the same name will be replaced. When finished the number of templates successfully loaded will be displayed.



The screenshot shows a dialog box titled "PeakFit". Inside, it says: "Interlaced data requires a repeating pattern of frames to process. Describe the regular repeat of the data:". Below this, it lists three parameters: "Start = The first frame that contains data", "Block = The number of continuous frames containing data", and "Skip = The number of continuous frames to ignore before the next data". An example is given: "E.G. 2:9:1 = Data was imaged from frame 2 for 9 frames, 1 frame to ignore, then repeat." At the bottom, there are three input fields: "Start" with the value "1", "Block" with the value "1", and "Skip" with the value "0". There are "OK" and "Cancel" buttons at the bottom right.





*Illustration 8: The FIT CONFIGURATION dialogue.*

As with the P<sub>EAK</sub> FIT plugin the settings contained in the configuration file are loaded when the plugin is initialised. If no file exists then a default set of settings will be created.

A different file can be selected by double clicking in the CONFIG FILE text box. This will open a file selection dialogue.

If the file name is changed and the new file exists the plugin will provide the option to reload the settings from the new configuration file.

When the plugin runs all the settings will be saved to the configuration file, overwriting any existing file.

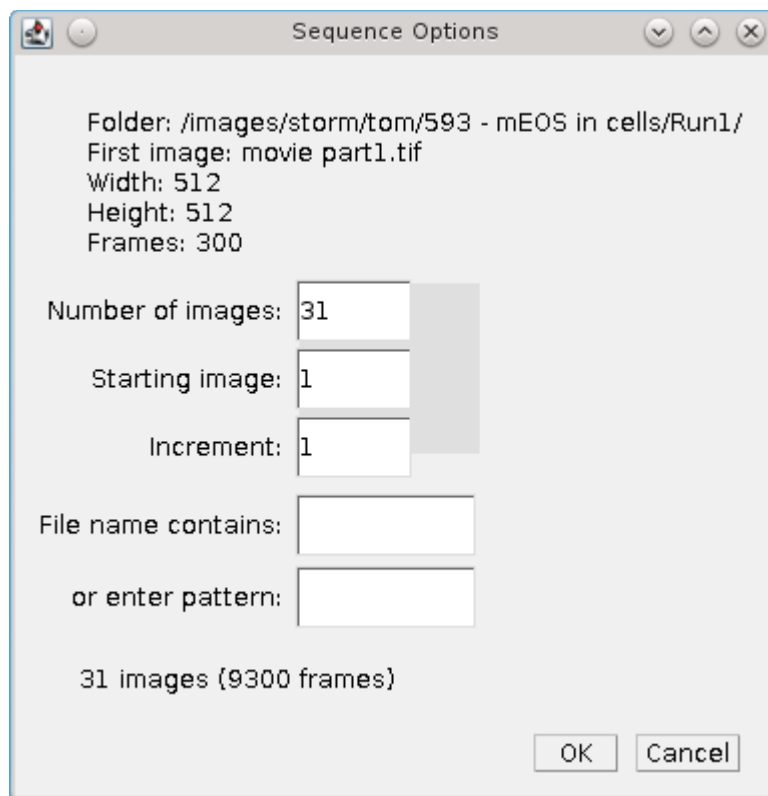
The FIT CONFIGURATION plugin allows the configuration to be viewed and updated without the need to have an image open. Since all plugins can be called from ImageJ scripts this also allows creation of a batch macro to change the configuration file.

### 5.5 Peak Fit (Series)

Allows the P<sub>EAK</sub> FIT plugin to be run on a folder containing many images. This allows the code to run on images that are too large to fit into memory or that may have been imaged in a sequence.

When the P<sub>EAK</sub> FIT SERIES plugin is executed it shows a folder selection dialogue where the user can select a folder containing a set of images. The plugin then scans the folder for images and sorts them numerically into a list, i.e. the first sequence of numeric digits in the filename are used to sort images. E.g. image2.tif is before image10.tif. The plugin then provides a dialogue to control how the series is loaded (see Illustration 9). The dialogue

shows the name and dimensions of the first image in the series. It is assumed that all images in the folder have the same dimensions (with the exception of the last image which may be truncated). The dialogue summarises at the bottom the total number of images and frames that will be read in the series.



*Illustration 9: Peak Fit (Series) sequence options dialogue.*

When different options are selected the plugin updates the count of the number of images and frames that will be processed. The parameters that effect what images are loaded are show below:

| Parameter          | Description  |
|--------------------|--|
| NUMBER OF IMAGES   | Specify the maximum number of images to load   |
| STARING IMAGE      | The first image in the sequence. The sequence begins at 1.   |
| INCREMENT          | The gap between images of the sequence. Use a number higher than 1 to miss out images in a sequence. |
| FILE NAME CONTAINS | Specify the text that the image filename must contain  |
| OR ENTER PATTERN   | Specify a pattern (regular expression) that the image filename must match                            |

If the user selects OK then the P<sub>EAK</sub> F<sub>IT</sub> plugin is run and must be configured as described in section 5.2. The only difference is that the plugin is not running on a single image but on

a series of images that are loaded sequentially and passed to the **PEAK FIT** plugin. The names of each image loaded in the image series will be saved with the results. This allows other plugins to access the original data associated with the results.

Note that if the directory contains a mixed collection of images then the results will not make sense.

### 5.6 Batch Fit

The **BATCH FIT** plugin runs the **PEAK FIT** plugin for each configuration setting within a batch file. The batch file allows the user to specify the input image files, the values for each of the **PEAK FIT** parameters and a results directory.

Running the **BATCH FIT** plugin presents a simple dialogue window, as shown in Illustration 10. The configuration file can be entered into the text box. Double-clicking in the text box will open a file selection dialogue to allow the file to be chosen.

Clicking on the 'CREATE CONFIG FILE' checkbox will open a file selection dialogue to allow a location to be chosen for a new default configuration file. If an existing file is selected it will be overwritten.

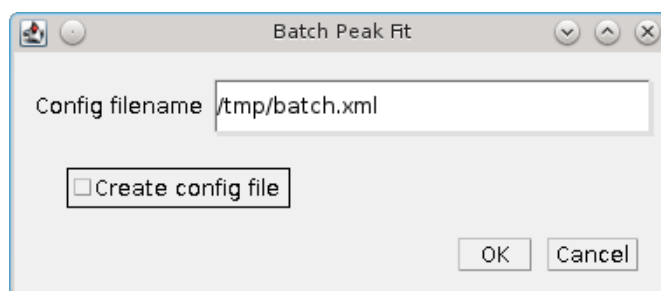


Illustration 10: Batch Peak Fit dialogue.

#### 5.6.1 Configuration File

The default configuration file is an XML file with the following format (note that the full list of parameters is not shown):

```
<gdsc.fitting.batchSettings>
  <images>
    <string>/path/to/image.tif</string>
    <!-- more images ... -->
  </images>
  <parameters>
    <parameter>
      <name>signalStrength</name>
      <value>30.0</value>
    </parameter>
    <!-- more parameters ... -->
  </parameters>
  <resultsDirectory>G:\Temp\</resultsDirectory>
  <runPeakFit>true</runPeakFit>
</gdsc.fitting.batchSettings>
```

#### <images>

The **<images>** tag is used to specify the full path to the input images. Each image should be within a **<string>** tag.

**<parameters>**

The <parameters> tag can contain any named parameter for the P<sub>EAK</sub> F<sub>IT</sub> plugin. Generating a new configuration file will create all the valid parameters with the default values. Multiple values for a parameter are specified using a comma delimited list in the <value> tag, e.g.

```
<parameter>
  <name>signalStrength</name>
  <value>30,40,50</value>
</parameter>
```

will run the P<sub>EAK</sub> F<sub>IT</sub> plugin using a signal strength of 30, 40 and 50.

If a parameter is removed then the default value will be used. This allows redundant parameters to be removed from the batch file for simplicity.

The values that are allowed for each parameter can be deduced from the P<sub>EAK</sub> F<sub>IT</sub> dialogue as follows:

| Field Type                    | Allowed values   | Example Field      | Example Value |
|-------------------------------|--|--------------------|---------------|
| NUMBER WITH DECIMAL PLACES    | Any floating-point number  | INITIAL STDDEV0    | 1.2           |
| NUMBER WITHOUT DECIMAL PLACES | Any integer  | SIGNIFICANT DIGITS | 5             |
| SLIDER                        | Any floating-point number  | SMOOTHING          | 0.5           |
| CHECKBOX                      | True/False   | FIT BACKGROUND     | TRUE          |
| DROP-DOWN LIST                | Use the upper-case text.<br>In some case the allowed value does not match the user-friendly name used in the dialogues. Run the F <sub>IT</sub> CONFIGURATION plugin, select the desired option and check the gdsc.smlm.settings.xml file for the allowed value. | FIT FUNCTION       | CIRCULAR      |

**<resultsDirectory>**

Provide the full path of the results directory. This is the directory used to save the P<sub>EAK</sub> F<sub>IT</sub> settings and the fit results for each single run. By default this initialises using the Java temporary directory.

**<runPeakFit>**

Set this to true to run the P<sub>EAK</sub> F<sub>IT</sub> plugin. Set to false to only create the configuration files in the results directory. This can be used to ensure that the batch file has correctly enumerated all the settings by checking the output configuration files.

### 5.6.2 Running the batch

Click OK to parse and run the batch file.

For each valid combination of parameters the **PEAK FIT** plugin will be executed on each input image. The results will be saved to a file named using the image filename and a sequential number from 1, e.g. InputImage.00001. The **PEAK FIT** settings will be written to an XML file with the suffix .xml. The **PEAK FIT** results will be saved to a text file of the same name with the suffix .xls. The fit results can be read using the **RESULTS MANAGER** plugin.

Note that the output options for a batch are limited to results files. The intended purpose for the **BATCH FIT** plugin is to allow multiple settings to be run on input test images. The results can then be further processed to determine which combinations of settings are the best parameters for fitting.

### 5.7 Spot Finder

Finds all the candidate maxima in an image.

This plugin uses the same algorithm as the **PEAK FIT** plugin to identify maxima. However all the candidates are saved to the output. No 2D Gaussian fitting or peak filtering is performed.

The fit configuration is the same as in the **Peak FIT** plugin. As with the **Peak FIT** plugin the settings contained in the configuration file are loaded when the plugin is initialised. If no file exists then a default set of settings will be created.

A different file can be selected by double clicking in the text box. This will open a file selection dialogue.

If the file name is changed and the new file exists the plugin will provide the option to reload the settings from the new configuration file.

When the plugin runs all the settings will be saved to the configuration file, overwriting any existing file.

### 5.8 Spot Finder (Series)

Allows the **SPOT FINDER** plugin to be run on a folder containing many images. This allows the code to run on images that are too large to fit into memory or have been imaged in a sequence.

The plugin allows the user to select a folder containing source images. This works using the same method as the **PEAK FIT (SERIES)** plugin. More details on the selection options can be found in section 5.5: **Peak Fit (Series)**.

### 5.9 Fit Maxima

Fits a 2D Gaussian to identified maxima.

This plugin uses the same algorithm as the **PEAK FIT** plugin to fit maxima. Candidates are taken from any results set held in memory with a valid image source (i.e. fitting a list of selected maxima can be performed on the original data). Candidates are collated per time frame and processed in ranked order until a number of successive fits fails or no candidates remain.

Candidates can be identified using the **SPOT FINDER** plugin. Running the **SPOT FINDER** and **FIT MAXIMA** plugins will produce the same results as using the **PEAK FIT** plugin. However separating the two steps allows processing to be performed on the candidates. For example the **TRACE MOLECULES** plugin can be used to join up candidates in successive frames and fit the combined stack. These are identified internally as spanning multiple frames by tagging an end frame onto the result. The **FIT MAXIMA** plugin will not fit any results that span multiple frames; these will be send directly through to the result output.

The fit configuration is the same as in the **PEAK FIT** plugin. As with the **PEAK FIT** plugin the settings contained in the configuration file are loaded when the plugin is initialised. If no file exists then a default set of settings will be created.

A different file can be selected by double clicking in the text box. This will open a file selection dialogue.

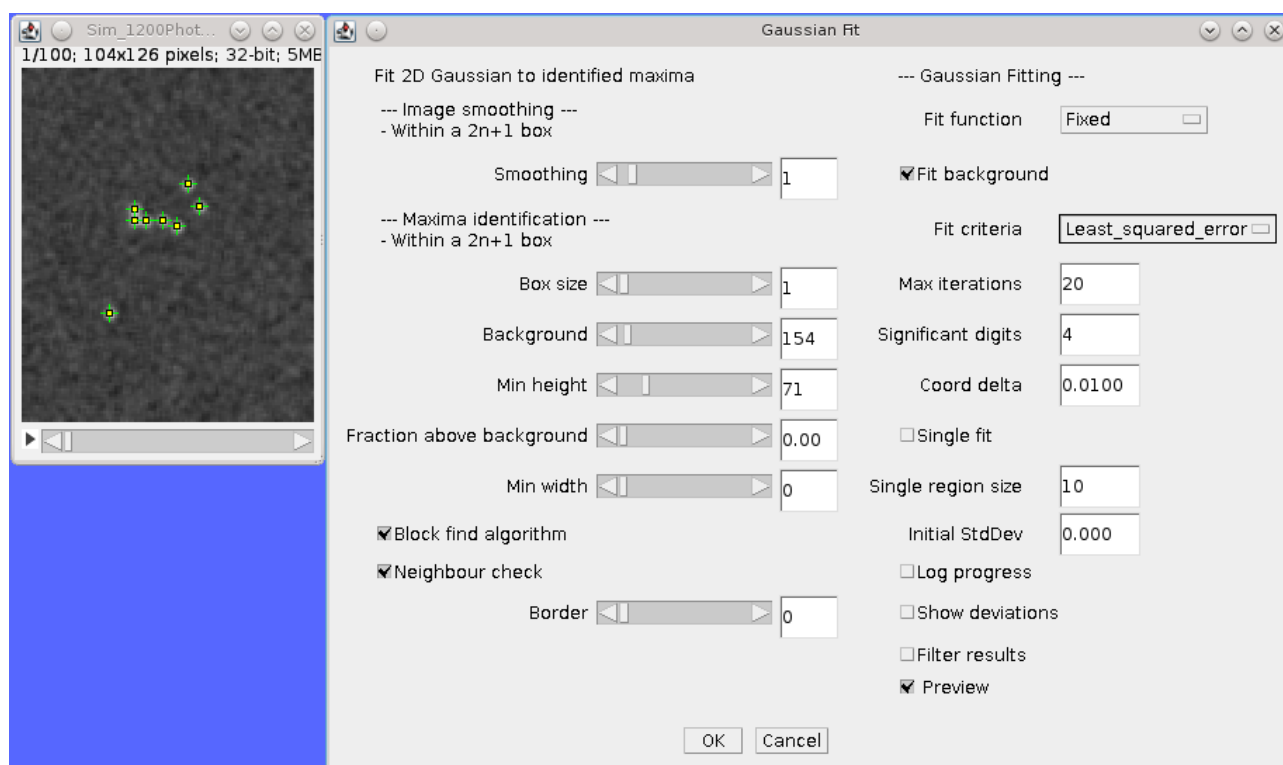
If the file name is changed and the new file exists the plugin will provide the option to reload the settings from the new configuration file.

When the plugin runs all the settings will be saved to the configuration file, overwriting any existing file.

### 5.10 **Gaussian Fit**

Uses the Gaussian fitting code to fit peaks. Since it requires no calibration parameters, the least-squares estimation is used. It is currently not possible to use Maximum Likelihood estimation.

The plugin identifies peaks on a smoothed image using non-maximal suppression and a set of size filters. It supports an interactive preview of the candidate peaks (see Illustration 11). These are then fitted using a 2D Gaussian and the results are output to a table.



*Illustration 11: Gaussian Fit plugin with live preview of the candidate peaks on the active image.*

The plugin is designed to work on rectangular regions of an image (or the whole image). It will work best in the following situation:

1. The background level is constant across the image
2. The peaks are distinct and well separated, e.g. over 40% of the peak height is clearly visible away from any other peak

The plugin works using a two stage process: (1) Identify peaks on a smoothed image using non-maximal suppression; (2) Fit all the peaks simultaneously using a 2D Gaussian with a global background.

### 5.10.1 Maxima Identification

Pixels are smoothed using a box filter of  $2n+1$  square dimensions around each pixel. Non-integer smoothing sizes are supported using a weight for the edge pixels.

Peaks are selected if they are higher than all other pixels in a  $2n+1$  box region. The box can be a different size from the smoothing window. The peaks must also satisfy the criteria of minimum height above the defined background, minimum width and a certain distance from the edge of the ROI.

When you run the plugin you can enable a preview checkbox at the bottom of the dialogue. This will draw on the image the currently identified maxima. This will update as you adjust the parameters such as the smoothing and the minimum peak height.

The following parameters are available:

| Parameter                 | Description  |
|---------------------------|--|
| SMOOTHING                 | The size of the smoothing window   |
| BOX SIZE                  | Identify maxima within a $2n+1$ box  |
| BACKGROUND                | Set the background for the image   |
| MIN HEIGHT                | Set the minimum height above the background  |
| FRACTION ABOVE BACKGROUND | Set the minimum height above the background as a fraction of the total peak height   |
| MIN WIDTH                 | The minimum peak-width at half-maxima  |
| BLOCK FIND ALGORITHM      | Use the block-find algorithm for non-maximal suppression [Neubeck and Van Gool, 2006].<br><br>This is much faster than a standard search of each $2N+1$ region around every pixel. The algorithm finds the maximum in each of the non-overlapping $N+1$ sized blocks in the image. Only the single maxima from each block is compared to the remaining pixels in the $2N+1$ region |
| NEIGHBOUR CHECK           | The block-find algorithm defines a maxima as any pixel with no other higher pixels within a radius. This can over-count maxima if they are equal height. Check adjacent blocks for equal height maxima, eliminating any maxima with neighbours that are already a maxima.  |
| BORDER                    | The border size in pixels to ignore when identifying maxima  |
| PREVIEW                   | Draw on the image the currently identified maxima  |

### 5.10.2 Gaussian Fitting

When peaks have been identified they are fit using a 2D Gaussian. This uses the same fitting algorithm as the `PEAK FIT` plugin.



The following parameters are available:

| Parameter          | Description   |
|--------------------|---|
| FIT FUNCTION       | <ul style="list-style-type: none"> <li>Fixed: Fits X,Y centre and amplitude</li> <li>Circular: Fits X,Y centre, combined X/Y deviation and amplitude</li> <li>Free Circular: Fits X,Y centre, individual X/Y deviation and amplitude</li> <li>Free: Fits X,Y centre, individual X/Y deviation, rotation angle and amplitude</li> </ul>  |
| FIT BACKGROUND     | <p>Enable background fitting. If disabled then the background is assumed to be zero.</p> <p>Note that cellular images contain background fluorescence and cameras may have a bias offset to allow characterisation of noise. This setting is best left on.</p>  |
| FIT CRITERIA       | <p>The fit uses a non-linear least squares routine until convergence. If convergence is not achieved by the maximum iterations then the fit fails. Convergence can be using:</p> <p>Least squares: convergence of the mean-squared error of the fit to a given number of significant digits</p> <p>Least squares plus: convergence of the mean-squared error of the fit to a given number of significant digits. Convergence is also achieved if 3 consecutive improvements are the same relative improvement and half of the maximum iterations has passed. This avoids slowly converging fits</p> <p>Coordinates: convergence of the X,Y centre coordinates to within a specified delta</p> <p>Parameters: convergence of each of the Gaussian parameters (e.g. centre, width and height) to a set number of significant digits</p> |
| MAX ITERATIONS     | Stop the fit when this is reached   |
| SIGNIFICANT DIGITS | <p>When comparing two numbers, defines the significant digit for equality, e.g. specify 3 to recognise 1 and 1.001 as equal.</p> <p>Note that Java floating points have a limited precision preventing comparisons to high numbers of significant digits. Using a value over 6 will not work (no convergence) for the GAUSSIAN or PARAMETER fitting criteria options and is not recommended for the LEAST SQUARED ERROR option.</p>   |
| COORD DELTA        | Define the smallest shift (in pixels) that specifies a movement of the X/Y coordinates. Used in the COORDINATES fit criteria  |

| Parameter          | Description  |
|--------------------|--|
| SINGLE FIT         | <p>Peaks are fit all at once using the entire ROI or can be fitted individually (Single fit). If using single fit then a region is drawn around the peak and fitted.</p> <p>Individual fitting works well if the box does not include any other peaks, otherwise a neighbour peak can interfere and it is best to fit all at once.</p> |
| SINGLE REGION SIZE | The size around each peak to use for fitting   |
| INITIAL STDDEV     | The initial standard deviation in pixels for the Gaussian. Set to zero to estimate this from the peak-width  |
| LOG PROGRESS       | Write progress messages to the ImageJ log  |
| SHOW DEVIATIONS    | The show deviations output produces an estimated error for each of the parameters in the results table.  |
| FILTER RESULTS     | <p>Removes fits that are far away from the initial guess of the Gaussian. Only valid for single fitting as peaks are filtered individually.</p> <p>Peaks are removed if they drift more than half the width of the smoothing window or if the width changes more than 3-fold from the initial estimate</p>                             |

Click OK to start the fitting. The fit uses a non-linear least squares routine until convergence. If convergence is not achieved by the maximum number of iterations the fit fails. The fitted results are output to a results table.

## 6 Results Plugins

The following plugins allow localisation results to be opened, converted and saved using various formats.

The plugins are described in the following sections using the order presented on the `PLUGINS > GDSC SMLM > RESULTS` menu.

### 6.1 Results Manager

Allows results to be output to a results table, an image, or to file.

The `RESULTS MANAGER` allows the `PEAK FIT` plugin to be run in the fastest mode with no output results (results are saved to memory). The results can then be visualised with different options using the `RESULTS MANAGER` and saved to file. Reconstructed images can be saved using the standard ImageJ `FILE > SAVE AS` options.

The plugin can be used to convert text results to binary results and vice-versa. Binary results save and load very fast but are not human readable. They are a good option for storing large results sets (10,000 or more localisations) for repeat analysis.

#### 6.1.1 Input options

Input can be from results in memory or a result file. Using the result file option allows the user to load results that were generated in a previous session.

The plugin can load data generated by the `PEAK FIT` plugin and saved as results file. It also loads results that have been saved from the `PEAK FIT` results table to a text file. External file formats are also supported. Currently the following alternative formats are supported:

- rapidSTORM
- Nikon NSTORM

##### 6.1.1.1 Image calibration

The SMLM plugins require that an image have a calibration to allow certain plugins to function. This calibration includes the pixel pitch, the total gain and the exposure time (as described in section 5.1.2). Certain file types do not have an image calibration. In this case the plugin will present a dialog where the user can enter the calibration for the results. This is added to the results and will be saved if using a SMLM file format.

#### 6.1.2 Output options

The output options are the same as in the `PEAK FIT` plugin. The only difference is that the file output requires that the name of the output file be provided. In the `PEAK FIT` plugin only the results directory was specified and the filename was taken from the image. Allowing the full filename to be specified provides greater flexibility in saving results.

Note that the `INPUT FILE` and `RESULTS FILE` fields support a mouse double-click action to open a file selection dialogue. As with the `PEAK FIT` plugin the results file suffix will be changed to `.xls` for text results and `.bin` for binary results.

One additional parameter is available for Image output:

| Parameter    | Description  |
|--------------|--|
| IMAGE WINDOW | <p>Applies to output images</p> <p>The Image Window specifies the number of consecutive frames from the results that should be plotted on a single ImageJ stack frame.</p> <p>By default this parameter is zero. All localisations are plotted on the same output frame.</p> <p>If this is set to 1 then each frame will be output to a new frame in the output image. Use this option to allow the input and output images to be directly compared frame-by-frame.</p> <p>If set higher than 1 then N frames will be collated together into one output image. Use this option to produce a time-slice stack through your results at a specified collation interval</p> <p>This option is not available during live fitting since the result must be sorted. This is not possible with multi-threaded code as the results can appear out of order.</p> |

As with all of the SMLM plugins, the RESULTS MANAGER is fully supported by the ImageJ macro recorder. This allows the use of macros to bulk convert many results sets, for example to render images and save them using the FILE > SAVE AS command.

## 6.2 Summarise Results

Shows a summary of all the results currently in memory.

The summary table shows the name of the results dataset and then the following statistics:

- The number of results
- The size of the results in memory
- The bounds of the results (minimum and maximum x and y coordinates of the source)
- The average, median, minimum and maximum of the precision and Signal-to-noise ratio (SNR)

## 6.3 Clear Memory Results

Removes all the results currently stored in memory. Presents a confirmation dialog showing the number of results that will be removed if the user selects OK.

## 6.4 Clear Memory Results (Multi)

Removes selected results currently stored in memory. Presents a dialog where the results to be removed can be selected. Then presents a confirmation dialog showing the number of results that will be removed if the user selects OK.

## 6.5 Rename Results

Allows the name of a results set held in memory to be changed.

By default the results are named using the name of the input image or results set appended with details of how the new results were generated, e.g. the name of the fitting engine or in the case of tracing analysis (Traced). The name is associated with the results set and used by analysis plugins or when reconstructing super-resolution images. It is also saved in the SMLM file format.

The names can be updated using the `RENAME RESULTS` plugin. The plugin presents a dialog with a single large text field. All the existing names are entered on the left side of the equals (=) symbol. The new name will be entered on the right side of the equals symbol followed by a semi-colon (;). The semi-colon is needed to support this plugin within the ImageJ macro language.

When the plugin is run all the target results set are identified using the names on the left. Any missing names are ignored allowing the user to delete many entries that should be unchanged. If the left and right side are identical then the name will be unchanged. Any invalid names not corresponding to an existing dataset cause an error to be displayed.

The destination names are then checked, any duplicates cause an error to be displayed. If no errors occurred then the datasets are renamed. Renaming may cause an existing dataset to be over-written if that dataset is not also renamed. This is allowed behaviour as it may be desirable to over-write a set of named results with the latest analysis results.

## 6.6 Resequence Results

Allows the frame number of results to be rebuilt assuming a repeating pattern of data and non-data frames.

The `PEAK FIT` plugin will fit a stack of images using a continuous frame number starting at 1. However this image may have been extracted from a larger image with interlaced data or been taken with a custom image acquisition workflow. In this case the frame number will be an incorrect representation of time. This is relevant for any analysis using the time gaps between localisations.

For example if every 20 images is a white light image and this was removed before fitting the frame number can be restored to add blank frames at 1, 21, 41, etc. Or the image may represent 1000 frames of imaging interspersed with 5 second gaps. Resequencing the results can put an appropriate gap between localisations in frame 1000 and 1001.

This plugin can resequence the results using the regular repeat of the original image. The following parameters are required:

| Parameter   | Description   |
|-------------|---|
| INPUT       | The results to process. Results will be directly updated (and there is no Undo operation)   |
| START       | The first frame containing data in the original image                                       |
| BLOCK       | The number of continuous frames that contain data in the original image                     |
| SKIP        | The number of continuous frames to skip before the next block of data in the original image |
| LOG MAPPING | Log to the Image log the mapping between the current and the new frame number               |

It is not possible to undo the RESEQUENCE RESULTS plugin. Before running the plugin you can save the results to file using the RESULTS MANAGER. These can be reloaded if the resequence operation produced an incorrect frame-by-frame mapping.

Note: If the data is interlaced it can be directly handled by the PEAK FIT plugin using the extra options (hold the SHIFT or ALT key down when running the plugin). There is no need to extra all the relevant data frames from the source image before running PEAK FIT.

## 6.7 Calibrate Results

Allows results held in memory to be calibrated (e.g. the pixel pitch and camera gain can be adjusted). Note that the raw data held in memory is stored using pixel units and image frames. Many of the plugins require calibrated units such as nanometers, micrometres, milliseconds and photons. Thus it is possible to store a calibration within the results. This calibration is added automatically when the results are generated inside the SMLM plugins. However the results may be loaded from file where a calibration is not present or the calibration was incorrect when the results were generated. This plugin allows the calibration to be updated.

When the plugin is run it presents a selection of the current result that are held in memory. If no results are available then an error is displayed. The user must select the results to update.

The following options are then available:

| Parameter                 | Description  |
|---------------------------|--|
| UPDATE ALL LINKED RESULTS | When new results are created from existing results they copy the calibration from the source results. Select this to update all the results in memory that share the same calibration.<br><br>If unselected then the other results will remain unchanged |
| CALIBRATION (NM/PX)       | The size of the pixels in nanometers   |
| GAIN (ADU/PHOTON)         | The number of Analogue-to-Digital Units (ADU) that represent   |

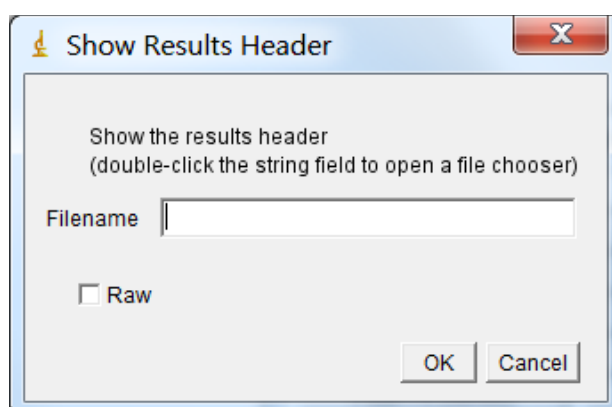
|                    |  |
|--------------------|--|
|                    | one photon   |
| EM-CCD             | A flag to indicate if the image was taken using an EM-CCD camera |
| EXPOSURE TIME (MS) | The exposure time for a single frame in milliseconds             |
| CAMERA BIAS (ADUs) | The bias offset added to each pixel by the camera                |
| READ NOISE (ADUs)  | The camera read noise in ADUs                                    |

Note that not all the calibration parameters have to be configured. Not all the plugins require every parameter. The most common parameters used are CALIBRATION and EXPOSURE TIME.

## 6.8 Show Results Header

Shows the header information from any supported localisation results file format. This is particularly useful for reading the header from GDSC SMLM binary format results files.

When the plugin is run the user is presented with a dialog where the results file can be selected:



The FILENAME field can be double-clicked to open a file selection window.

When the plugin runs it will attempt to open the selected file and read it as a localisation results file. The header will be extracted and reported to the IMAGEJ log window.

If the RAW option is selected then the header will be written directly. Note that the IMAGEJ log window does not show tab characters. However if a line containing tab characters is copied from the log window and pasted into a text editor these characters are maintained.

If the RAW option is not selected then the plugin will attempt to extract the standard information stored in a GDSC SMLM results file header:

| Field  | Description                    |
|--------|--------------------------------|
| FORMAT | The GDSC SMLM file format code |
| NAME   | The name of the results        |

|               |  |
|---------------|--|
| BOUNDS        | The bounds of the results data (minx, miny, width, height) |
| CALIBRATION   | The calibration (nm/pixel, exposure time, etc)             |
| CONFIGURATION | The fitting configuration used to produce the results      |

## 6.9 Filter Results

Filters a set of localisations using various criteria.

Requires the fitting results to be loaded into memory. When the plugin is run the user is presented with a selection dialogue of the available results. The user can select the results to filter.

The plugin analyses the selected results and computes limits for each of the filters based on the data. A second dialogue is then shown that allows the filters to be adjusted. The following filters are available:

| Parameter     | Description   |
|---------------|---|
| MAX DRIFT     | The maximum distance the fitted centre is allowed to be from the original maxima location   |
| MIN SIGNAL    | The minimum signal strength   |
| MIN SNR       | The minimum signal-to-noise ratio   |
| MIN PRECISION | The minimum precision   |
| MIN WIDTH     | The minimum width for a localisation. Width is defined as the peak width at half maxima (PWHM)  |
| MAX WIDTH     | The maximum width for a localisation. Width is defined as the peak width at half maxima (PWHM)  |
| MASK          | Select an image to use as a mask. Only localisations that occur in non-zero mask pixels will be included in the results. The mask is scaled to match the width and height of the result source. |

If any parameter is set to zero it will be ignored.

## 6.10 Free Filter Results

Filters a set of localisations using various criteria.

Requires the fitting results to be loaded into memory. When the plugin is run the user is presented with a selection dialogue of the available results. The user can select the results to filter.

The dialogue also contains a text area. This is used to construct filters using eXtensible Markup Language (XML). A set of example filters can be shown by clicking on the **SHOW DEMO FILTERS** checkbox. This will record the available filters to the ImageJ log. Note: To see



the filters and copy them to the clipboard as examples you will first have to cancel the plugin dialogue.

The following filters are available:

| Filter                    | Description  |
|---------------------------|--|
| WIDTHFILTER               | Filters the results using an upper width factor. Width is relative to the initial peak width   |
| WIDTHFILTER2              | Filter the results using a width range. Width is relative to the initial peak width  |
| SIGNALFILTER              | Filter results using a lower signal threshold. The threshold is applied in photons   |
| SNRFILTER                 | Filter results using a lower SNR threshold   |
| SNRFILTER2                | Filter results using a lower SNR threshold and width range. Width is relative to initial peak width  |
| PRECISIONFILTER           | Filter the results using an upper precision threshold  |
| SNRHYSTERESISFILTER       | Filter results using a signal-to-noise (SNR) threshold. Any results above the upper SNR limit are included. Any results below the lower SNR limit are excluded. Any results between the limits (candidates) are included only if they can be traced through time, potentially via other candidates, to a valid result.<br>The distance used for tracing is the search distance multiplied by the average precision of the candidates |
| PRECISIONHYSTERESISFILTER | Filter results using a precision threshold. Any results below the lower precision limit are included. Any results above the upper precision limit are excluded. Any results between the limits (candidates) are included only if they can be traced through time, potentially via other candidates, to a valid result.<br>The distance used for tracing is the search distance multiplied by the average precision of the candidates |
| TRACEFILTER               | Filter results that can be traced over time frames. The trace distance is specified in pixels and the time threshold in frames   |
| COORDINATEFILTER          | Filter results using a coordinate range. This can be used to crop the results to a rectangular region, for example when batch processing results subsets in a macro  |
| ANDFILTER                 | Filter results using the combination of two filters. Results must pass both filters  |
| ORFILTER                  | Filter results using the combination of two filters. Results can   |

|  |                    |
|--|--------------------|
|  | pass either filter |
|--|--------------------|

The **FREE FILTER** plugin provides a powerful tool for customising the subset of results that are extracted. For example it is possible to extract only the results that have either (1) a Signal-to-noise ratio above 10 and a Precision of less than 30nm; or (2) that can be traced to another localisation within 0.5 pixels and 1 time frame using the following combined filter:

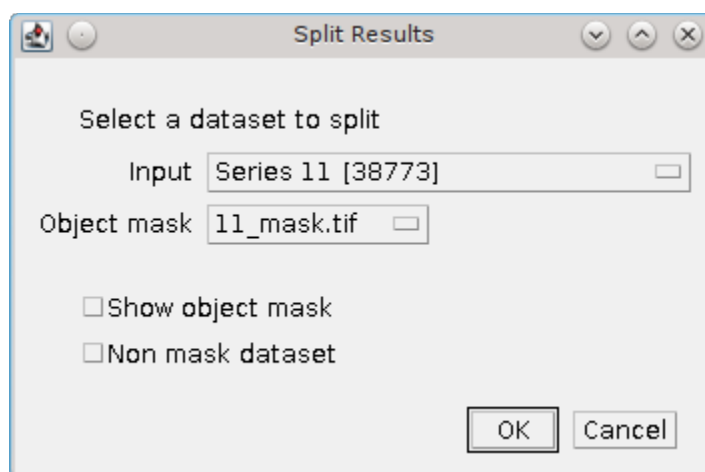
```
<OrFilter>
  <filter1 class="AndFilter">
    <filter1 class="SNRFilter" snr="10.0"/>
    <filter2 class="PrecisionFilter" precision="30.0"/>
  </filter1>
  <filter2 class="TraceFilter" d="0.5" t="1"/>
</OrFilter>
```

Note how the combined filters require that the contained filters are specified in a **<FILTER>** tag and the type of filter is specified using the **CLASS** attribute. The filter parameters are specified using attributes.

## 6.11 Split Results

Splits a set of localisation results using a 2D mask. This allows the results to be divided into subsets, for example one results set per cell.

When the plugin is run the user is presented with a dialog allowing the results set and the mask to be selected.



If no results are currently held in memory, or no images are open then an error is displayed.

The following parameters can be set:

| Parameters       | Description  |
|------------------|--|
| INPUT            | Select the results set to split  |
| OBJECT MASK      | Select the mask image used to identify objects                               |
| SHOW OBJECT MASK | Display an image with a unique pixel value for each object                   |
| NON MASK DATASET | Include a results set for all the results that do not occur within an object |

### 6.11.1 Analysis

The selected object mask is analysed for objects. An object is assigned using contiguous pixels with the same value. Object pixels are joined using 4-connected edges, not 8-connected neighbours. This allows a single line of diagonal pixels to divide two objects.

The 2D dimensions from the bounds of the results set are then mapped onto the object mask dimensions to create scaling factors. For example a results set with dimensions 512x512 can be mapped to a mask of 1024x1024 with a scaling factor of 2. The mapping is performed in each dimension allowing use of an object mask that does not match the original rectangle ratio of the results set. (Note that the dimensions of the results set can be displayed using the `SUMMARISE RESULTS` plugin.) However it is normal for a mask to be created using an image taken on the same camera as the source image for the super-resolution dataset to allow masking image objects, for example cells.

A new dataset is created for each mask object. The results are then mapped to the object pixels using the scaling factors and added to the appropriate dataset. The plugin will save any non-empty dataset to memory. The dataset is named using the input results set name plus the object ID (starting from 1). To see the objects use the `SHOW OBJECT MASK` option.

Optionally the plugin can save a results set containing all the results that do not map to any objects. This has an ID of zero. To create this dataset use the `NON MASK DATASET` option.

## 6.12 Results Match Calculator

Calculate the match statistics between two results sets.

The `RESULTS MATCH CALCULATOR` allows two sets of localisations to be compared. The results are processed per time frame. The plugin can identify results that span multiple time frames, e.g. trace results produced by the `TRACE MOLECULES` plugin. These will be split into a single localisation for each frame, all with identical coordinates.

Localisations are identified as a match if they are within a set distance. The plugin computes matches iteratively allocating the closest pairs first until no more matches can be made. The matches are used to compute comparison score metrics to show the similarity between the two results sets. The available metrics are Precision, Recall, F-score and Jaccard. Details of the comparison metrics can be found in Appendix B : Comparison Metrics.

The score metrics are shown in a results table. Optionally a table of the matched pairs can be displayed showing the matched and unmatched localisations. The pairs table supports interactive identification of the selected points on the source image (see 6.12.1). Any previous results in the pairs table will be cleared.

Since matches are computed at a set distance threshold the plugin provides the ability to perform analysis at many distances. In this case the pairs are matched at the largest distance threshold. Then the scores for lower distance thresholds can be computed by eliminating pairs that are too far apart.

The following parameters can be set:

| Parameters           | Description   |
|----------------------|---|
| RESULTS1             | The first results set   |
| RESULTS2             | The second results set  |
| DISTANCE             | The minimum distance for a match  |
| INCREMENTS           | The number of times to increment the distance threshold   |
| DELTA                | The value to increment the distance threshold   |
| BETA                 | Controls the weighting between Precision and Recall for the custom F-score  |
| SHOW TABLE           | Display a table of the match statistics   |
| SHOW PAIRS           | Display a table of the matched pairs with coordinates and distances   |
| SAVE CLASSIFICATIONS | The data from results set 2 will be saved to a PeakResults file. If the point matches a point in results set 1 then the original value will be set to 1, otherwise the original value is set to 0   |
| ID ANALYSIS          | <p>If the results in the results set have an Id label for each localisation the plugin will compute the number of molecules that were matched. The TP, FN and recall for each results set will be added to the results table as additional columns.</p> <p>Note: Ids are added to the results by various plugins, e.g. TRACE MOLECULES, CREATE DATA</p> |

### 6.12.1 Interactive Results Table

The results SHOW PAIRS table will show the coordinates and distance between matched pairs. Unmatched paris will be added to the table at the end of the matches for the same time frame.

To assist in viewing the localisations that are matches the table supports mouse click interaction. The table is linked to the results source for the RESULTS1 input. If this is an

image open in ImageJ the table can draw ROI points on the image:

- Double-clicking a line in the results table will draw a single point ROI at the coordinates identified. The stack position will be set to the correct frame
- Highlighting multiple lines with a mouse click while holding the shift key will draw multiple point ROI on the coordinates identified. The frame will be set to the last identified frame in the selection

The coordinates for each point are taken from the X1 & Y1 columns, or if they are unavailable, the X2 & Y2 columns.

### 6.13 Trace Match Calculator

Calculate the match statistics between two sets of traced molecules.

The TRACE MATCH CALCULATOR allows sets of traced localisations to be compared. The plugin scans the results held in memory and only allows results to be selected where they contain an entry that spans multiple time frames. Such results can be generated using the TRACE MOLECULES plugin.

The plugin compares traces using the following distance weighted score:

$$Score = overlap \cdot 1 / (1 + d^2 / d_t^2)$$

where *overlap* is the number of frames where both traces are present, *d* is the distance between the two points and *d<sub>t</sub>* is a threshold distance. The score is composed of two parts: the overlap and the distance weighting. The distance weighting has a maximum value of 1 and reduces to zero. The weighting is 0.5 when *d* equals *d<sub>t</sub>*. Thus the score will favour a match between close traces and those with the largest overlap in time. Since any overlapping traces will be scored a maximum allowed distance between the traces is set. This is currently configured at 2*d<sub>t</sub>*.

The plugin computes matches iteratively allocating the highest scoring pairs first until no more matches can be made. The matches are used to compute comparison score metrics to show the similarity between the two results sets. The available metrics are Precision, Recall, F-score and Jaccard. Details of the comparison metrics can be found in Appendix B : Comparison Metrics.

The overall score is the sum of the scores for all of the matched pairs. This score is then normalised by the maximum number of time-points contained in either result set.

$$Total\ Score = \frac{\sum_i S_i}{\max(p1, p2)}$$

The number of time-points (*p*) is equal to the count of the number of individual localisations in the results before tracing.

$$p = \sum_i tEnd_i - tStart_i + 1$$

where *tStart* is the trace start time and *tEnd* is the trace end time. The normalisation penalises the score if either result set contains many unmatched or partially matched traces. The overall score should have a value between 0 and 1.

The score metrics are shown in a results table. Optionally a table of the matched pairs can be displayed showing the matched and unmatched localisations. The pairs table supports interactive identification of the selected points on the source image. This is the same functionality as the RESULTS MATCH CALCULATOR (see 6.12.1). Any previous results in the pairs table will be cleared.

The plugin can compare one or two results sets to the same reference. This allows the user to compare different tracing results to a benchmark, for example the results of tracing raw localisations can be compared to tracing filtered localisations. If two test sets are input then the matched pairs table will contain additional columns to display triples.

The following parameters can be set:

| Parameters | Description   |
|------------|---|
| RESULTS1   | The first results set (the reference)   |
| RESULTS2   | The second results set (test set 1)   |
| RESULTS3   | The third results set (test set 2; optional)  |
| DISTANCE   | The minimum distance for a match. This is equal to $2d_i$ in the score distance weighting formula |
| BETA       | Controls the weighting between Precision and Recall for the custom F-score                        |
| SHOW PAIRS | Display a table of the matched pairs with coordinates and distances                               |
| SORT PAIRS | Specify the sort method for the pairs: Time or Score  |

### 6.14 Spot Inspector

Extracts the fitted spots from an image into a stack ordered by the user-selected score.

The SPOT INSPECTOR plugin allows visualisation of the fitted spots from a result set held in memory. The results are ordered using a user-selected score. Then the pixels surrounding each spot centre are extracted into an image stack named using the plugin title.

The plugin also produces a results table containing all the results in their rank order. If a line on the table is double-clicked using the mouse then the appropriate slice of the image stack is selected. Any spot from the entire results set will be labelled on the image using a multi-point ROI. This can produce many labels on the image which can be dismissed by clicking on the image or using EDIT > SELECTION > SELECT NONE (CTRL+SHIFT+A).

The plugin will check if the original source can be located. This may be an image open in ImageJ or the original file or image series located on disk. This allows inspection of fitting results from an image series too large to fit into memory. If the original source cannot be located then the plugin will fail with an error message.

The following parameters can be configured:

| Parameters       | Description  |
|------------------|--|
| INPUT            | Select the input results set   |
| RANKING          | Select the score used to rank the results  |
| RADIUS           | Select the pixel radius around the localisation centre to extract  |
| CALIBRATED TABLE | Show the localisation sizes in nm (default is pixels)  |
| PLOT SCORE       | Show a plot of the score against the rank  |
| PLOT HISTOGRAM   | Show a histogram of the score  |
| HISTOGRAM BINS   | The number of bins to use for the histogram  |
| REMOVE OUTLIERS  | Remove any localisation from the plots that lies more than 1.5x the interquartile range above or below the 25 <sup>th</sup> and 75 <sup>th</sup> percentile (quartile boundaries). This can remove poor scoring results that skew the plot visualisation |

## 7 Analysis Plugins

The following plugins perform analysis on localisation results.

The plugins are described in the following sections using the order presented on the **PLUGINS > GDSC SMLM > ANALYSIS** menu.

### 7.1 Drift Calculator

Stabilises the drift in the image by calculating the drift for each frame and applying it to correct the positions of the localisations.

Drift can be calculated using:

- Sub-image alignment – Subsets of the localisation data are used to build images that are aligned to the global image
- Fiducial markers within an image – The positions of fiducial markers are aligned to their centre-of-mass over the entire image
- Reference stack alignment – Images from a reference stack (e.g. a white light image) are aligned to a global projection
- Drift file – The drift is loaded from file

Further details of the methods are shown in the sections below.

Each method will produce an X and Y offset (drift) for specified frames. (Not all frames must have a drift value.) The calculated values are smoothed using LOESS smoothing (a local regression of the points around each value). The drift error is calculated as the total sum of the drift per frame. The current drift is applied to the results and the drift calculation repeated. This process is iterated until the drift has converged (shown by a small relative change in the drift error). In the case of loading the curve from file, no iteration is performed but the drift points may be smoothed.

The calculated drift curve is interpolated to assign values for any frames without a drift value. Any frames outside the range of frames with drift values cannot be interpolated. These are assigned the same drift as the closest frame with a known drift value (i.e. no extrapolation of the drift curve is performed to avoid errors from poor data at the ends of the analysis range).

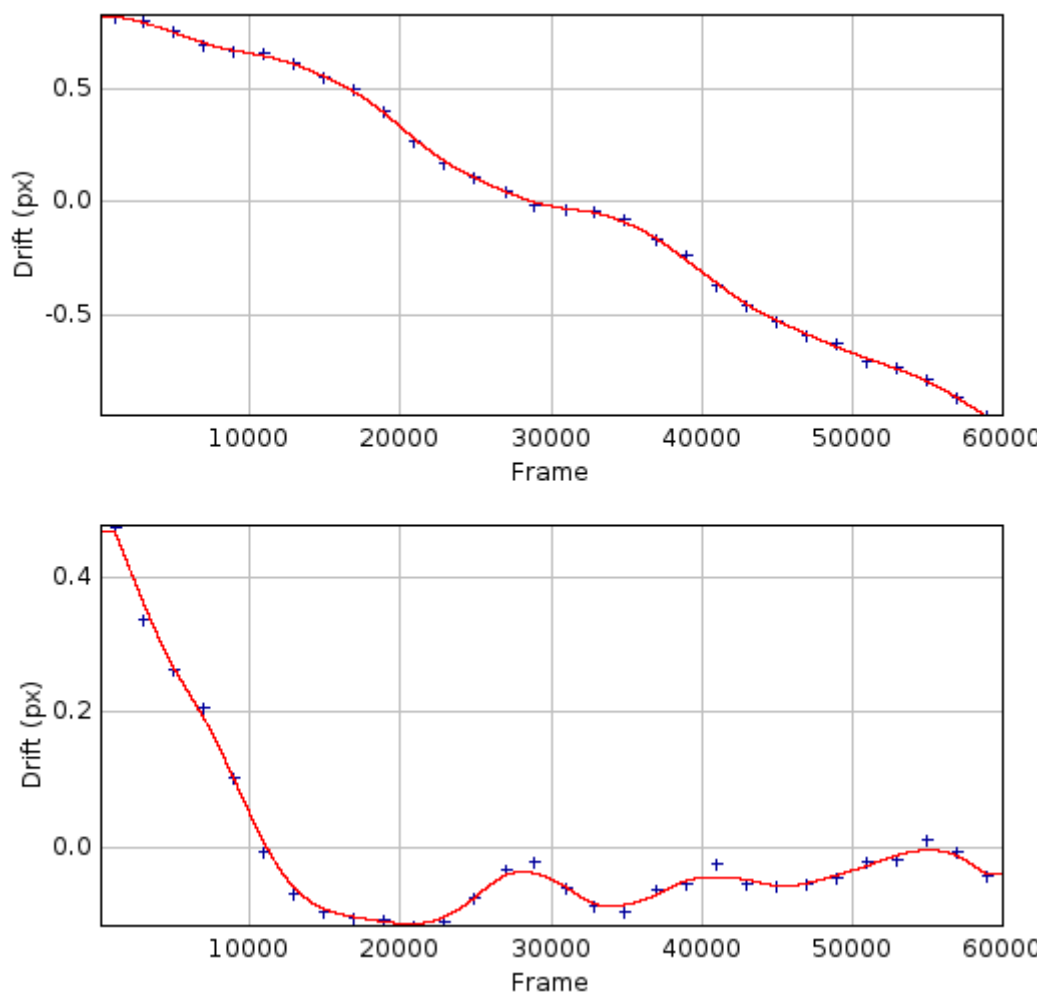
The plugin requires the following parameters:

| Parameter      | Description  |
|----------------|--|
| INPUT          | Select the results set to analyse.<br><br>Fitting results can be stored in memory by the <b>PEAK FIT</b> plugin or loaded from file using the <b>RESULTS MANAGER</b> |
| METHOD         | The method for calculating the drift curve   |
| MAX ITERATIONS | The maximum number of iterations when calculating the drift curve  |



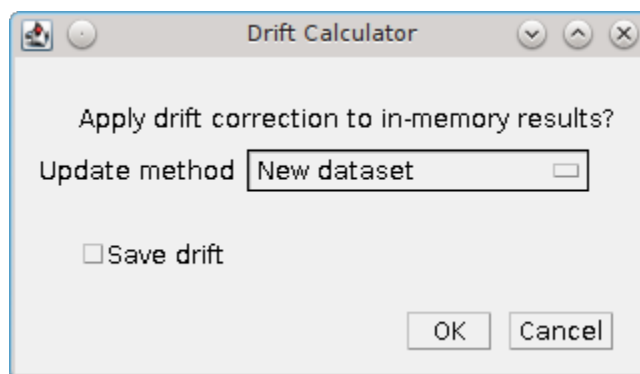
|                      |  |
|----------------------|--|
| RELATIVE ERROR       | Stop iterating when the relative change in the total drift is below this level   |
| SMOOTHING            | The window width to use for LOESS smoothing (values below 0.1 are unstable for small datasets). Set to zero to ignore smoothing  |
| LIMIT SMOOTHING      | Select this option to adjust the smoothing parameter so that smoothing uses a number of points within the configured minimum or maximum  |
| MIN SMOOTHING POINTS | The minimum number of points to use for LOESS smoothing  |
| MAX SMOOTHING POINTS | The maximum number of points to use for LOESS smoothing. A lower value will allow the drift curve to track the raw drift data more closely but may start to model noise  |
| SMOOTHING ITERATIONS | The number of iterations for LOESS smoothing. 1 is usually fine  |
| PLOT DRIFT           | Produce an output of the drift for the X and Y shifts. Shows the raw drift correction for each frame and the smoothed correction (see Illustration 12).<br><br>Use this option to see the effects of different smoothing parameters before applying the drift correction to the data |

An example of a calculated drift curve is shown in Illustration 12. The drift curves show that drift in the X and Y axis are independent and drift may be approximately constant (X-drift) or can vary during the course of the image acquisition (Y-drift).



*Illustration 12: Plot of the calculated X-drift (top) and Y-drift (bottom) following drift calculation using the localisation sub-images method. The STORM dataset consisted of 410,000 results over 60000 frames. Sub-images were produced using 2000 frames, raw drift calculated (blue) and smoothed using LOESS with a maximum of 4 points before interpolation to the drift curve (red).*

When the drift analysis is complete the plugin will present a dialog asking the user what to do with the drift curve:



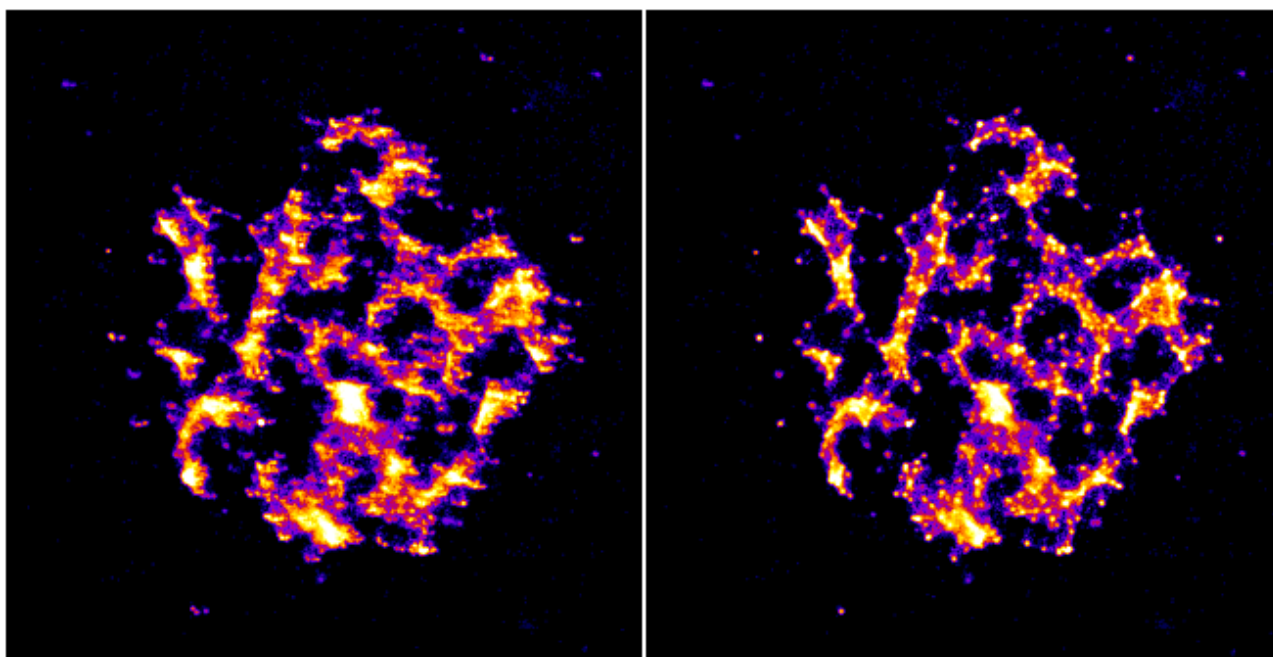
*Illustration 13: Drift calculator curve options dialogue*

The following options are available:

| Parameter     | Description  |
|---------------|--|
| UPDATE METHOD | <p>None – Do not use the drift curve</p> <p>Update – Update the input results' coordinates using the drift</p> <p>New dataset – Create a new dataset with updated coordinates using the drift</p> <p>New truncated dataset – Create a new dataset with updated coordinates but truncate the results to the interpolation range of the drift curve (points outside this range may have a poor drift correction)</p> |
| SAVE DRIFT    | <p>Save the drift curve to file. The plugin will prompt the user for a filename. The drift curve can later be loaded by the DRIFT CALCULATOR plugin using the FILE option for the METHOD parameter</p>   |

Note that if the drift correction is used to update the results then only the results held in memory will change. Any derived output, for example a table of the results or a reconstructed image, will have to be regenerated from the new results.

Example images showing the original and drift corrected localisations following sub-image alignment are shown in Illustration 14. The drift correction curve for the image data is shown in Illustration 12. Note how drift correction has removed the blur from the main image and resolved smeared spots into single points.



*Illustration 14: Example of drift correction. Left: Original localisations; Right: Drift corrected localisations.*

Details of the different drift calculation methods are shown below.

### 7.1.1 Localisation Sub-images

If the localisations represent a structural image then a subset of the localisations will also represent the same structure. Where there are a large number of localisations (for example in STORM images) it is possible to create sub-images from sub-sets of the data and align them. The LOCALISATION SUB-IMAGES method performs the following steps:

1. Initialise the drift for each time point to zero
2. Produce sub-images from N consecutive frames using the coordinates from the localisations plus the drift for that time point
3. Combine all sub-images to a master projection
4. Aligns each sub-image to the master projection using correlation analysis. The drift time point is set using the average frame from all the localisations in the sub-image
5. Smooth the drift curve
6. Calculate the change to the drift and repeats from step 2 until convergence

The following parameters can be specified:

| Parameter             | Description  |
|-----------------------|--|
| FRAMES                | The number of consecutive frames to use to construct each sub-image  |
| MINIMUM LOCALISATIONS | The minimum number of localisations used for a sub-image. If the frames for a sub-image do not contain enough localisations then |

|          |  |
|----------|--|
|          | they will be combined with the next set of frames until an image is produced with the minimum number of localisations  |
| FFT SIZE | Specify the size of the reconstructed image. Large images will (a) be slow to process; (b) if there are not enough localisations the image density will be spread out and the correlation between images may not work; and (c) may produce a better drift estimate if the number of localisations per sub-image is very high |

### 7.1.2 Drift File

The drift can be loaded directly from file. The file must contain delimited records of:

Time X Y

The fields can be delimited with tabs, spaces or commas. Any line not starting with a digit is ignored. Only the time-points that are within the time range of the input results are used.

The file is assumed to contain the final drift curve and no iterations are performed to update the curve. The curve may be smoothed using LOESS smoothing before being interpolated and applied to the data.

The DRIFT FILE method allows the same drift to be applied to multiple data sets. For example if an image is produced with a white light channel for drift tracking and two different colour channels with localisation data the same drift from the white light image can be used to correct both sets of localisations.

### 7.1.3 Reference Stack Alignment

The drift can be calculated using a reference stack image, for example this may be a white light image taken during the experiment. The reference stack must be a single stack image. Some microscopes may make a separate image during acquisition for the white light. However if all the frames are joined into a master image then you can extract reference stack slices from a master image using ImageJ's Substack Maker plugin (IMAGE > STACKS > TOOLS > MAKE SUBSTACK...).

The slice numbers in the reference stack will not correspond to the slices in the localisation results. Therefore the plugin allows the user to specify the actual slice number of the first slice in the reference stack (start frame), and then the frame spacing between slices in the stack. The actual frame for the stack is then calculated as the previous frame (starting from the start frame) plus the frame spacing.

The REFERENCE STACK ALIGNMENT method performs the following steps:

1. Initialise the drift for each time point to zero
2. Calculate an average projection of the slices shifted by the current drift
3. Aligns each slice to the average projection using correlation analysis to compute the drift
4. Smooth the drift curve
5. Calculate the change to the drift and repeats from step 2 until convergence

The following parameters can be specified:

| Parameter     | Description  |
|---------------|--|
| START FRAME   | The actual slice number from the original image for the first frame from the reference stack |
| FRAME SPACING | The number of frames from the original image between each slice in the reference stack       |

For example if a white light image was taken at the start and then every 20 frames then the method should be called with parameters 'start\_frame=1 frame\_spacing=20'. The drift will then be calculated as if the slices were at time points 1, 21, 41, etc.

### 7.1.4 Marked ROIs

This method uses constant fiducial markers that are placed within the image to allow the drift to be tracked (e.g. fluorescent beads). The method is only available in the drop-down options when there are ROIs listed in the ImageJ ROI manager.

Rectangular ROIs can be placed around the fiducial markers on the original image and then added to the ROI manager (press `CTRL+T`). All the ROIs in the manager will be used to calculate the drift. Ensure that you choose regions containing a constant bright spot that is present through the majority of frames. If multiple spots are within the ROI only the brightest one per frame will be used. Ideally these are fluorescent beads added to the image as fiducial markers. The Marked ROIs method performs the following steps:

1. Initialise the drift for each time point to zero
2. Calculate the centre-of-mass of all the spots selected within each ROI
3. For each frame and each ROI, calculate the shift from the spot to the centre-of-mass of the ROI
4. For each frame, produce a combined shift using a weighted average of the shift from each ROI. The weight is the spot signal (number of photons)
5. Smooth the drift curve
6. Calculate the change to the drift and repeats from step 2 until convergence

The Marked ROIs method requires no additional parameters, only ROIs within the ROI Manager. Note however that each ROI's bounds (x,y,width,height) are used to find spots within the input localisations and so the ROI should be selected using an image with the same scale and image bounds as the input data. Ideally this should be an average intensity projection of the original image but it can be a super-resolution reconstruction of the data made using an image scale of 1.

### 7.1.5 Image alignment using correlation analysis

Image alignment in the `LOCALISATION SUB-IMAGE` and `REFERENCE STACK ALIGNMENT` methods is done using the maximum correlation between images (including sub-pixel registration using cubic interpolation). This is computed in the frequency domain after a Fast Fourier Transform (FFT) of the image. The method of producing and then aligning images is computationally intensive so the plugin uses multi-threading to increase speed. The

number of threads to use is the ImageJ default set in `EDIT > OPTIONS > MEMORY & THREADS...`

## 7.2 Trace Molecules

Traces localisations through time and collates them into traces using time and distance thresholds. Each trace can only have one localisation per time frame. With the correct parameters a trace should represent all the localisations of a single fluorophore molecule including blinking events.

### 7.2.1 Background

The fluorophores used in single molecule imaging can exist in several states. When in an active state it can absorb light and emit it at a different frequency (fluorescence). The active state can move into a dark state where it does not fluoresce. The dark state can move back to the active state. Eventually the molecule moves into a bleached state where it will no longer fluoresce (photo-bleached). The rates of the transitions between states are random as are the number of times this can occur. This means that it is possible for the same molecule to turn on and off several times causing blinking.

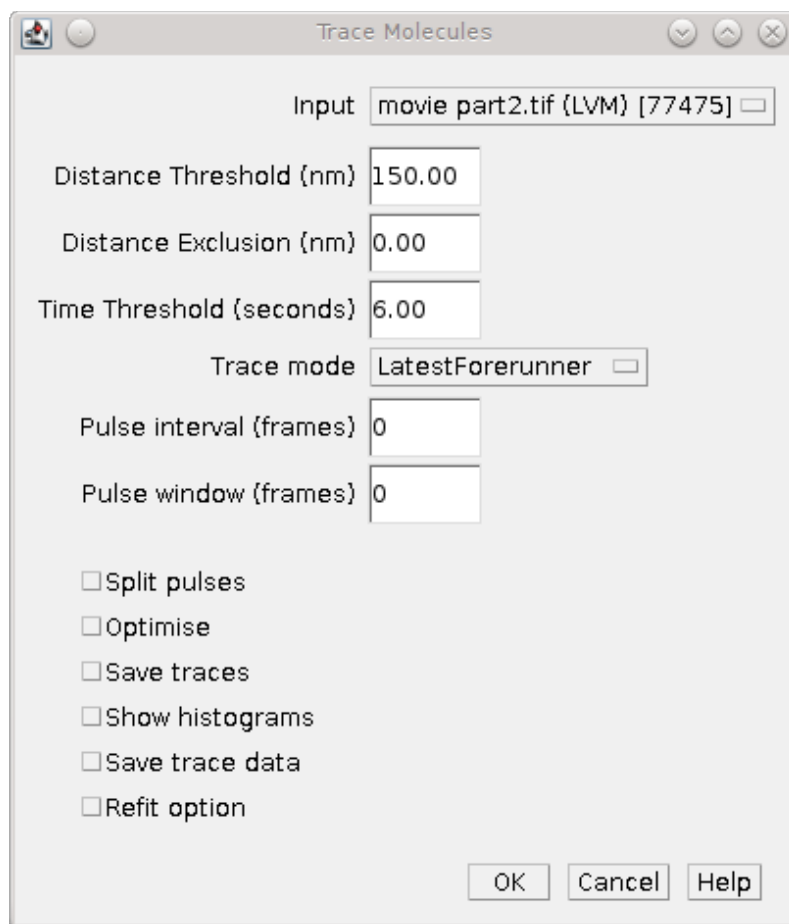
To prevent over-counting of molecules due to blinking it is possible to trace the localisations through time. Any localisation that occurs very close to another localisation from a different frame may be the same molecule. The distance between localisations can be spatial or temporal. Using two parameters it is possible to trace localisations using the following algorithm:

Any spot that occurred within time threshold and distance threshold of a previous spot is grouped into the same trace as that previous spot. When all frames are processed the resulting traces are assigned a spatial position equal to the centroid position of all the spots included in the trace.

The molecule tracing algorithm is based on the work of Coltharp, *et al.* (2012).

### 7.2.2 Trace Molecules Plugin

The `TRACE MOLECULES` plugin allows temporal tracing to be performed on localisations loaded into memory. Illustration 15 shows the plugin dialogue.



*Illustration 15: Trace Molecules dialogue*

The following parameters can be configured:

| Parameter                | Description  |
|--------------------------|--|
| INPUT                    | Specify the localisations to use   |
| DISTANCE THRESHOLD (NM)  | Maximum distance (in nm) for two localisations to belong to the same trace   |
| DISTANCE EXCLUSION (NM)  | Exclusion distance (in nm) where no other localisations are allowed. Use this setting to be sure that a trace links together localisations that are not close to any other localisations.<br><br>Ignored if less than the distance threshold |
| TIME THRESHOLD (SECONDS) | Maximum distance in seconds for two localisations to belong to the same trace (should cover a minimum of 1 frame)  |
| TRACE MODE               | LATESTFORERUNNER - Search from the closest time distance in the past for earlier localisations of the same fluorophore. This is the best mode for moving molecules<br><br>EARLIESTFORERUNNER - Search from the maximum time distance in      |



|                 |  |
|-----------------|--|
|                 | the past for earlier localisations of the same fluorophore<br><br>SingleLinkage – Search all time-points in the past up to the maximum time distance to find the closest localisation. This is equivalent to single-linkage clustering. This mode is slower since the other modes will stop searching time-points when a localisation has been found within the distance threshold |
| PULSE INTERVAL  | Sets the pulse interval for traces (in frames). Set to zero to disable pulse analysis. See section 7.2.2.1: Pulse Analysis   |
| PULSE WINDOW    | Sets the pulse window for traces (in frames). Set to zero to disable pulse analysis. See section 7.2.2.1: Pulse Analysis   |
| SPLIT PULSES    | Enable this to split traces that span the a pulse boundary into separate traces. Use this setting if your imaging conditions use pulsed activation and you have imaged for long enough between pulses to be sure that all fluorophores have photo-bleached   |
| OPTIMISE        | If selected the plugin will provide a second dialogue that allows a range of distance and time thresholds to be enumerated (see section 7.2.3)   |
| SAVE TRACES     | When the tracing is complete, show a file selection dialogue to allow the traces to be saved   |
| SHOW HISTOGRAMS | Present a selection dialog that allows histograms to be output showing statistics on the traces, e.g. total signal, on time and off time   |
| SAVE TRACE DATA | Save all the histogram data to a results directory to allow further analysis. A folder selection dialog will be presented after the tracing has finished   |
| REFIT OPTION    | Provide the option to extract all the frames corresponding to a single trace from the source image into a combined image and perform PSF fitting   |

The plugin will trace the localisations and store the results in memory with a suffix 'Traced'. Two additional datasets are created: all single localisations which could not be joined are given a suffix 'Trace Singles'; all traces are given a suffix 'Traces'. The Trace Singles plus Traces datasets equal the Traced dataset.

A summary of the number of traces is shown on the ImageJ status bar. The results are accessible using the RESULTS MANAGER plugin.

### 7.2.2.1 Pulse Analysis

The options PULSE INTERVAL and PULSE WINDOW allow the user to specify a repeating period within the image time sequence. Only traces that originate within a frame defined by the pulse will be included in the output.

For example a pulse could be defined using `PULSE INTERVAL 30` and `PULSE WINDOW 3`. Only traces that have their first localisation in frames 1-3, 31-33, 61-63, etc. will be output in the final traces.

This option was added to allow analysis of images acquired using a pulsed activation laser. Consequently only localisations that could be traced back to a short period after the activation pulse are of interest. All other localisations are likely to be random background fluorescence events.

The options should be disabled when using a continuous activation laser by setting the parameters to zero.

When using a pulse activation it is possible to photo-bleach all fluorophores that were activated by the pulse before the next pulse. This requires a long pulse interval. If you are confident that all molecules have bleached then it does not make sense for a trace to span pulse intervals. Use the `SPLIT PULSES` option to break apart any traces that span a pulse interval boundary into separate traces.

### 7.2.3 Optimisation

It is possible to produce an estimate of the optimum distance and time thresholds if the blinking rate of the fluorophore is known. Note that the blinking rate can be estimated from the data using the `BLINK ESTIMATOR` plugin (see 7.7: Blink Estimator). Alternatively it can be measured by manual inspection of purified single fluorophores sufficiently spread on a slide to avoid two molecules in the same location. Care must be taken to ensure that the imaging condition are the same as those used for *in vivo* experiments.

The optimisation method is adapted from Coltharp, *et al.* (2012). Optimisation is based on the following equation:

$$\text{Expected molecules} = \frac{\text{Observed pulses}}{\text{Blinking rate}}$$

The observed pulses is the number of single pulse events that are observed in the data, i.e. continuous emission from the same fluorophore. Dividing this by the average blinking rate of a fluorophore should give the number of molecules.

The observed pulses can be found by tracing the localisations using a time threshold of 1 and a distance threshold that will allow a match of the same molecule position. This will only join localisations that are in adjacent frames into a single light pulse. The distance to use is obtained from the data using 2.5x the average localisation precision.

For example if 10,000 pulses have been identified by tracing at  $t=1$ ,  $d=2.5 \times \overline{\text{Precision}}$  and the blinking rate is known to be 2 then the expected number of molecules is 5,000.

A score metric can be computed for a given tracing result:

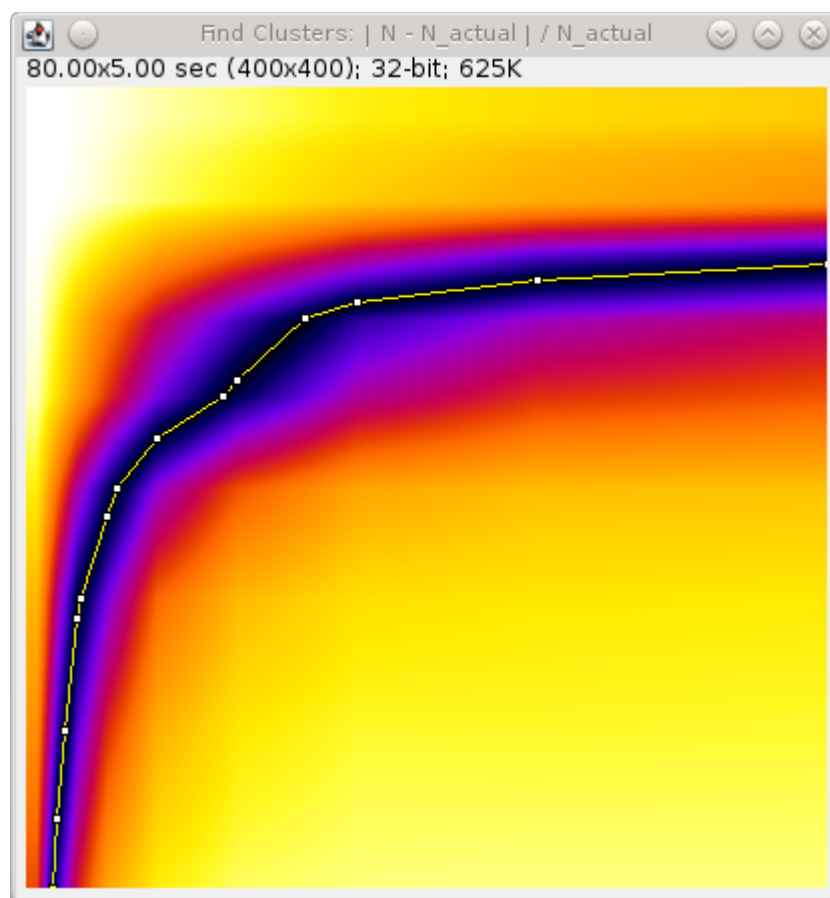
$$\text{Score} = \frac{\text{Traces} - \text{Expected molecules}}{\text{Expected molecules}}$$

The closer the score to zero, the more likely that the tracing parameters are correct. In addition a negative score indicates over-clustering, a positive score is under-clustering. Consequently a plot of the distance and time thresholds verses the score will indicate the parameters that are best suited to achieving a zero score. The zero score can be more easily seen on such a plot by using the absolute value of the score.

The following table describes the parameters used during the optimisation:

| Parameter                    | Description  |
|------------------------------|--|
| MIN DISTANCE THRESHOLD (PX)  | The minimum distance threshold   |
| MAX DISTANCE THRESHOLD (PX)  | The maximum distance threshold   |
| MIN TIME THRESHOLD (SECONDS) | The minimum time threshold   |
| MAX TIME THRESHOLD (SECONDS) | The maximum time threshold   |
| STEPS                        | The number of steps to use between the minimum and maximum. Steps intervals are chosen using a geometric (not linear) scale to bias sampling towards lower threshold values  |
| BLINKING RATE                | The average blinking rate of the fluorophore. Must be above 1 (otherwise no occurrences of repeat molecules are expected).<br>Note that the blinking rate is equal to the number of blinks + 1   |
| PLOT                         | Produce a plot of the score against the time/distance threshold: <ul style="list-style-type: none"> <li>• None: No plot</li> <li>• Nearest Neighbour: Output only the calculated values</li> <li>• Bilinear: Interpolate between calculated values to produce a smoother plot</li> </ul> |

When the optimisation option is selected it is preferable to choose a plot option. An ideal plot will show an inverted L-shape as shown in Illustration 16. The parameters that achieve a score close to zero are shown in black. The scale of the image has been calibrated to use the scale of the distance and time thresholds. Therefore hovering over a part of the image will show the time (X-axis) and distance (Y-axis) threshold required for the given score.



*Illustration 16: Output plot from the Trace Molecules optimisation algorithm. The plot shows the absolute score against the time (X-coordinate) and distance (Y-coordinate) thresholds.*

Note that at the end of optimisation the thresholds are automatically set using the zero score that is closest on the plot to the origin. This should be a compromise point between the two thresholds. The values used will be written to the ImageJ log window. The tracing algorithm then runs and the traces are stored in memory.

If the optimised thresholds are not suitable it is left to the user to interpret the plot of the scores and select the best values. For example this could be done by assuming the distance threshold calculated using  $2.5 \times \text{Precision}$  is correct and looking up the corresponding time threshold when the score is zero.

#### 7.2.4 Memory Output

The tracing algorithm assigns a unique ID to each trace. All the localisations that are a member of that trace are assigned the same ID. The results are then saved into memory. The results are named using the input results set name plus a suffix as follows:

| Suffix                | Description  |
|-----------------------|--|
| TRACED                | A full set of localisations with each assigned the corresponding trace ID  |
| TRACE CENTROIDS       | The localisations of each trace combined into a centroid.<br>Centroids have a signal equal to the sum of the localisations, the coordinates are set using the signal weighted centre-of-mass of the localisations. The background is averaged and the noise combined using the root of the sum-of-squares. The Gaussian standard deviation of the localisation is set using the average precision of the localisations, calculated using the Mortensen formula |
| TRACE SINGLES         | Contains the localisations that were not part of a trace, i.e. are a single localisation   |
| TRACE MULTI           | Contains the localisations that were part of a trace, i.e. are a multi-frame localisation  |
| TRACE CENTROIDS MULTI | The localisations of each trace combined into a centroid. Only traces with multiple localisations are included   |

It is possible to save these results to file using the RESULTS MANAGER plugin.

### 7.2.5 File output

If the SAVE TRACES option is selected then the plugin will show a file selection dialogue allowing the user to choose the location of the clusters results file. The results will use the same format as the plain-text file results option in the PEAK FIT and RESULTS MANAGER plugins. However all the localisations for each trace will be stored together under a Trace entry. The Trace entry will have the format:

```
#Trace x y (+/-sd) n=[n], b=[n], on=[f], off=[f], signal=[f]
```

where:

- x & y are the coordinates of the centroid
- sd is the standard deviation of distances to the centroid
- n is the size of the cluster
- b is the number of pulses (bursts of continuous time frames)
- on is the average on-time of each pulse
- off is the average off-time between each pulse
- signal is the total signal for all the localisations in the trace

The prefix # character allows the clusters to be ignored as comments, for example when the cluster file is loaded as a results file.

Note that the number of bursts is equal to the number of blinks + 1. It is equivalent to the blinking rate of the molecule.

### 7.2.6 Refit option

Provides the option to extract all the frames corresponding to a single trace from the source image into a combined image and perform PSF fitting.

The refit option is only available if the plugin can successfully load the original input source for the results. This may be an image open within ImageJ or could be the image file or series located on disk.

The plugin must also be able to recover the original fitting configuration used when generating the results. This should be stored by default as part of the results set. The fitting configuration is used to set the standard fitting options in the dialogue. The fitting options are the same as those available within the P<sub>EAK</sub> F<sub>IT</sub> plugin.

The plugin provides some parameters to control how the combined image is constructed and then fitted:

| Parameter               | Description   |
|-------------------------|---|
| FIT CLOSEST TO CENTROID | When enabled the candidate maxima is seeded using the centroid position of the trace; only one maxima should be fitted. If disabled the fitting algorithm is allowed to detect maxima in the combined image and fit them all. The closest maxima to the centroid is then chosen   |
| DISTANCE THRESHOLD      | Specify the maximum distance that a fitted maxima can deviate from the centroid before it is rejected   |
| EXPANSION FACTOR        | Fitting in P <sub>EAK</sub> F <sub>IT</sub> uses a region surrounding the candidate. This is defined using the expected peak standard deviation multiplied by the search width. This default region is enlarged by the expansion factor to allow a larger region to be input into the fitting routine. This allows inclusion of all the trace localisations and some surrounding pixels                       |
| DEBUG FAILURES          | The success of each fit is recorded. This option will show an image of the first few fits that fail for each different reason. The reason will be written to the ImageJ log and the stack of localisations will be displayed. The final image in the localisation stack is the average intensity projection of the stack. It should be a clear PSF spot if fitting succeeds and a poor spot if fitting failed |

The fitting routine will fit all the traces that contain more than one localisation. This can take a long time for large results sets since no heuristics are performed to rank traces and stop if fitting continuously fails.

A summary of the results of fitting each trace is shown in the ImageJ log, e.g.:

Trace fitting : 32 singles : 92 / 208 fitted : 880 separated

Note that singles are traces that contain one localisation. These are not fitted. The separated score shows the number of localisations that were part of traces that could not be fitted. These are judged to be from poor tracing through non identical molecules and are thus split apart.

Fitting results are stored in memory. The results set is named using the input results set name plus the suffix 'Trace Fit'. Note that the results will be a composite of the successful fits and the remaining localisations. All the traces that were a single molecule are included; the positions of successfully fitted traces are included; and all the localisations that were traced but not successfully fitted are included as single localisations.

### 7.2.6.1 Use case for Trace Fitting

When using Peak Fit all frames are processed individually. This excludes the possibility of using adjacent frames to improve the signal-to-noise ratio of spots. It is possible to run an off-line analysis of an image by:

- Identification of candidates using `SPOT FINDER`
- Tracing neighbouring candidates using `TRACE MOLECULES` with `t=1` and a distance threshold of 1 (or more). Note the candidates generated by spot finder have coordinate accuracy to integer pixel values so a threshold of at least 1 pixel is recommend
- Fitting the traces of neighbours using the refit option of `TRACE MOLECULES`
- Fitting the remaining candidates using the `FIT MAXIMA` plugin to select the combined 'Trace Fit' results. `FIT MAXIMA` ignores any localisations which span multiple frames and only fits the remaining single frame candidates

The resulting dataset will be the output of fitting the neighbouring localisations across time frames and all the other candidate maxima per time frame.

## 7.3 Cluster Molecules

Cluster localisations into clusters using distance and optionally time thresholds. When using a time threshold each cluster can only have one localisation per time frame. With the correct parameters a cluster should represent all the localisations of a single fluorophore molecule including blinking events.

This plugin is very similar to the `TRACE MOLECULES` plugin and many of the options are the same. The following options are available:

| Parameter                | Description   |
|--------------------------|---|
| INPUT                    | Specify the localisations to use  |
| DISTANCE THRESHOLD (NM)  | Maximum distance (in nm) for two localisations to belong to the same cluster  |
| TIME THRESHOLD (SECONDS) | Maximum distance in seconds for two localisations to belong to the same cluster (should cover a minimum of 1 frame).<br><br>This is only used for some algorithms |

|                      |   |
|----------------------|---|
| CLUSTERING ALGORITHM | Set the clustering algorithm. See below for details   |
| PULSE INTERVAL       | Sets the pulse interval for clusters (in frames). Clusters will only contain localisations from within the same pulse.<br><br>Set to zero to disable pulse analysis. See section 7.2.2.1: Pulse Analysis  |
| SPLIT PULSES         | Enable this to ensure clusters contain only localisations within a single pulse (defined by the PULSE INTERVAL).<br><br>Use this setting if your imaging conditions use pulsed activation and you have imaged for long enough between pulses to be sure that all fluorophores have photo-bleached |
| SAVE CLUSTERS        | When the clustering is complete, show a file selection dialogue to allow the clusters to be saved   |
| SHOW HISTOGRAMS      | Present a selection dialog that allows histograms to be output showing statistics on the clusters, e.g. total signal, on time and off time  |
| SAVE CLUSTER DATA    | Save all the histogram data to a results directory to allow further analysis. A folder selection dialog will be presented after the clustering has finished   |
| REFIT OPTION         | Provide the option to extract all the frames corresponding to a single cluster from the source image into a combined image and perform PSF fitting  |

The plugin will cluster the localisations and store the results in memory with a suffix ' Clustered'. Two additional datasets are created: all single localisations which could not be joined are given a suffix ' Cluster Singles'; all clusters are given a suffix ' Clusters'. The Cluster Singles plus Clusters datasets equal the Clustered dataset.

### 7.3.1 Clustering Algorithms

The following table lists the available clustering algorithms:

| Algorithm                 | Description  |
|---------------------------|--|
| PARTICLE SINGLE-LINKAGE   | Joins the closest pair of particles, one of which must not be in a cluster. Clusters are not joined and can only grow when particles are added.  |
| CENTROID-LINKAGE          | Hierarchical centroid-linkage clustering by joining the closest pair of clusters iteratively   |
| PARTICLE CENTROID-LINKAGE | Hierarchical centroid-linkage clustering by joining the closest pair of any single particle and another single or cluster. Clusters are not joined and can only grow when particles are added. |



|  |   |
|--|---|
| <b>PAIRWISE</b>                                      | Join the current set of closest pairs in a greedy algorithm. This method computes the pairwise distances and joins the closest pairs without updating the centroid of each cluster, and the distances, after every join (centroids and distances are updated after each pass over the data). This can lead to errors over true hierarchical centroid-linkage clustering where centroid are computed after each link step. For example if A joins B and C joins D in a single step but the new centroid of AB is closer to C than D. |
| <b>PAIRWISE WITHOUT NEIGHBOURS</b>                   | A variant of <b>PAIRWISE</b> <b>WITHOUT NEIGHBOURS</b> is to join the closest pairs only if the number of neighbours for each is 1. In the event that no pairs has only a single neighbour then only the closest pair is joined.<br><br>This algorithm should return the same results as the <b>CLOSEST</b> algorithm (but with different run-time performance)   |
| <b>CENTROID-LINKAGE (DISTANCE PRIORITY)</b>          | Hierarchical centroid-linkage clustering by joining the closest pair of clusters iteratively. Clusters are compared using time and distance thresholds with priority on the closest time gap (within the distance threshold).   |
| <b>CENTROID-LINKAGE (TIME PRIORITY)</b>              | Hierarchical centroid-linkage clustering by joining the closest pair of clusters iteratively. Clusters are compared using time and distance thresholds with priority on the closest distance gap (within the time threshold).   |
| <b>PARTICLE CENTROID-LINKAGE (DISTANCE PRIORITY)</b> | Hierarchical centroid-linkage clustering by joining the closest pair of any single particle and another single or cluster. Clusters are not joined and can only grow when particles are added.<br><br>Clusters are compared using time and distance thresholds with priority on the closest time gap (within the distance threshold).   |
| <b>PARTICLE CENTROID-LINKAGE (TIME PRIORITY)</b>     | Hierarchical centroid-linkage clustering by joining the closest pair of any single particle and another single or cluster. Clusters are not joined and can only grow when particles are added.<br><br>Clusters are compared using time and distance thresholds with priority on the closest distance gap (within the time threshold).   |

Only the **(DISTANCE PRIORITY)** AND **(TIME PRIORITY)** methods use the time information. All the other algorithms will ignore the **TIME THRESHOLD** and optional **PULSE INTERVAL** parameters.

All the clustering algorithms (except **PAIRWISE**) are multi-threaded for at least part of the algorithm. The number of threads to use is the ImageJ default set in **EDIT > OPTIONS > MEMORY & THREADS...**

The **PAIRWISE** algorithm is not suitable for multi-threaded operation but is the fastest algorithm by an order of magnitude over the others. All other algorithms have a similar run-time performance except the **PAIRWISE WITHOUT NEIGHBOURS** algorithm which doesn't just

search for the closest clusters but also tracks the number of neighbours. The algorithm should return the same results as the `CLOSEST` algorithm but the analysis of neighbours has run-time implications. At very low densities this algorithm is faster since all pairs without neighbours can be joined in one step. However at most normal and high densities tracking neighbours is costly and the algorithm is approximately 3x slower than the next algorithm.

### 7.4 Draw Clusters

Draws collections of localisations with the same ID on an image, for example the output from `TRACE MOLECULES`, `CLUSTER MOLECULES` or `TRACE DIFFUSION`.

The `DRAW CLUSTERS` plugin collects localisations together with the same ID into clusters. If localisations have no ID they are ignored. The localisations in each cluster are then sorted by start frame to create an ordered trace. Each cluster is then drawn on a selected image, or a new empty image using an ImageJ overlay. The size of the empty image can be specified but the plugin will zoom a small image until the display window is 500 pixels on the long edge. The overlay can be removed using '`IMAGE > OVERLAY > REMOVE OVERLAY`'.

The user has the option to draw the cluster as a series of points (a cluster) or as a connected line (a trace). Plotting points is appropriate when the clusters contain multiple localisations that do not represent a single molecule. Plotting a line is appropriate when the cluster represents a single molecule's position through time.

If an input image is selected then the number of time frames in the image is compared with the maximum start frame in all the localisations. If the image stack is large enough the plugin can draw each cluster on the specific frame containing the localisation. In this case the entire cluster is drawn and the point contained in that particular frame is highlighted using a cross.

An example is show below:



The following options are available:

| Parameter         | Description   |
|-------------------|---|
| INPUT             | Select the input results to draw. The list only contains results where all the localisations have an ID   |
| IMAGE             | Specify the output image to draw on. If [None] is selected then the output will be to an image named Draw Clusters.   |
| IMAGE SIZE        | The size of the default output image on the long edge. Small images will be zoomed  |
| EXPAND TO SINGLES | <p>Expand any localisation with a different start and end frame into a series of singles (with the same coordinates).</p> <p>This option is useful for drawing multi-frame localisations, for example centroid representations of clusters. The expansion is performed before the size filtering.</p> |

| Parameter          | Description   |
|--------------------|---|
| MIN SIZE           | The minimum size of clusters. All clusters below this will be ignored   |
| MAX SIZE           | The maximum size of clusters. All clusters above this will be ignored. Set below the MIN SIZE to disable  |
| TRACES             | Select this option to assume the localisations are connected as a time-series trace. The output will draw lines connecting the points.<br>If not selected the output will draw each point individually  |
| SORT               | Specify how the clusters will be sorted before drawing. The sort order determines the colour taken from the look-up table   |
| SPLINE FIT         | If selected the line will be drawn as a spline fit. This is only valid when TRACES is selected  |
| USE STACK POSITION | If selected the plugin will draw each cluster on the specific frame containing the localisation.<br>If an output image is selected then the stack must contain enough frames to plot all the localisations.<br>If no output image is selected then the default output image will be created as a stack of the required size |
| LUT                | Specify the look-up table (i.e. the colour) used to plot the clusters. Each cluster is a single colour. The colour is varied according to the order the clusters are processed.<br>Note: Single colour LUTs vary the intensity of the colour from 50% to 100% to provide identification of the order                        |

### 7.4.1 Drawing cluster centroids

Note that the DRAW CLUSTERS plugin will draw all the members of a cluster on the image. If you wish to draw only the centroids then you should either:

- Load the centroids from a pre-processed file as a single localisation with a unique ID
- Run a clustering algorithm (TRACE MOLECULES / CLUSTER MOLECULES) and then select an appropriate centroids dataset that is stored in memory

If you wish to draw the cluster centroid on each frame that contains a member of the cluster then you can use the EXPAND TO SINGLES and USE STACK POSITION options.

## 7.5 Density Image

Analyses the local density around each localisation and outputs an image using the

density score. Can optionally filter localisations based on the density score into a new results set.

The DENSITY IMAGE plugin counts the number of localisations in the neighbourhood of each localisation. The score is then used to create an image of the density of localisations. The following options are available:

| Parameter                | Description  |
|--------------------------|--|
| INPUT                    | Select the input results to analyse  |
| RADIUS                   | Specify the radius of the local region around each localisation  |
| USE ROI                  | <p>If selected the user will be presented with a dialogue allowing the selection of an image with an area ROI drawn on. If only one image has an area ROI it will be chosen automatically. The ROI will be scaled to the dimensions of the input results and only localisations within this region analysed.</p> <p>This option allows the user to construct density images using only part of the input image results</p> |
| ADJUST FOR BORDER        | Localisations that are close to the edge of the analysis region will have a local area that extends outside the region. This will result in under-counting of neighbours. This can be compensated for by scaling the count of neighbours using the proportion of area sampled, e.g. if 40% of the area is sampled (60% missing) the count is multiplied by 100/40  |
| IMAGE SCALE              | Specify the scale of the output image relative to the input results  |
| CUMULATIVE IMAGE         | The output image may contain several localisations on the same pixel. If CUMULATIVE IMAGE is enabled each pixel will be the combined sum of all the localisation scores at that point. If disabled each pixel will be highest score of any localisation score at that point  |
| USE SQUARE APPROXIMATION | <p>If enabled use a square neighbour region with edges of 2 x radius. If disabled use a circular neighbour region with the specified radius.</p> <p>Using the square approximation avoids computation of pairwise distances by assigning localisations to a grid and summing the counts within the grid neighbourhood. It is very fast when using extremely large datasets</p>   |
| SQUARE RESOLUTION        | Define how many grid points will be used per pixel. A larger resolution provides more accurate counting results at the cost of computational speed and memory size   |

| Parameter               | Description  |
|-------------------------|--|
| SCORE                   | Define the score that will be output in the density image. See 7.5.1: Available Density Score functions  |
| FILTER LOCALISATIONS    | Remove all localisations that have a score below the specified value. The remaining localisations will be stored in a new results set in memory. The results will be name using the source results name plus the suffix 'Density Filter' |
| FILTER THRESHOLD        | Define the filter threshold. Most score functions will be either zero or 1 if the density matches that expected for randomly distributed particles   |
| COMPUTE RIPLEY'S L PLOT | Analyse the data using a range of distance thresholds and compute the Ripley's L-score for each. Show a plot of the L-score against the distance threshold. See 7.5.2: Ripley's L-plot   |

### 7.5.1 Available Density Score functions

The plugin computes the number of localisations in a region surrounding each localisation. This value is biased by the total number of localisations and the total area sampled. The number can be normalised using the sample size and sample area to produce different scores. The scores are based on the Ripley's K and L functions. Ripley's K-function describes the fraction of points within a distance threshold  $r$  normalised by the average density of points:

$$K(r) = \lambda^{-1} \sum_{i \neq j} I(d_{ij} < r) / n$$

where  $I$  is the indicator function,  $d_{ij}$  is the Euclidean distance between the  $i^{\text{th}}$  and  $j^{\text{th}}$  points in a data set of  $n$  points, and  $\lambda$  is the average density of points, generally estimated as  $n/A$ , where  $A$  is the area of the region containing all points. Note that the indicator function has a value of 1 for all elements ( $d_{ij}$ ) that are a member of a set (in this case  $d_{ij} < r$ ), otherwise it is zero. If the points are randomly distributed then  $K(r)$  should be equal to  $\pi r^2$ .

The K function can be variance normalised to the L function:

$$L(r) = \sqrt{(K(r) - \pi r^2) / \pi}$$

The L function should be equal to  $r$  if the points are randomly distributed and its variance should be approximately constant in  $r$ . A plot of  $L(r) - r$  should follow the horizontal zero-axis if the data are dispersed using a homogeneous Poisson process. A plot of  $(L(r) - r) / r$  will also be zero but should have equal variance at any  $r$  thus the magnitude of deviations from zero can be directly compared for any  $r$ .

Note that the K and L functions apply to an entire dataset. However it is possible to produce a per localisation K or L score as follows.

$$K_i(r) = \lambda^{-1} \sum_{j=1}^j I(d_{ij} < r)$$

Since the count of localisations is expected to be the area of the local region multiplied by

the average sample density ( $\lambda$ ) then the  $K_i(r)$  score is expected to be the area of the local region.

The  $K_i(r)$  score can be normalised by the local region area and should be equal to 1. As before the  $K_i(r)$  score can be variance normalised to an L score:

$$L_i(r) = \sqrt{(K_i(r)/\pi)}$$

In this case the L-score is normalised by  $\pi$ . If using a square approximation then the normalisation factor for the area of a square is 4.

The  $L_i(r)$  score should be equal to  $r$ . The following scores are supported in the plugin:

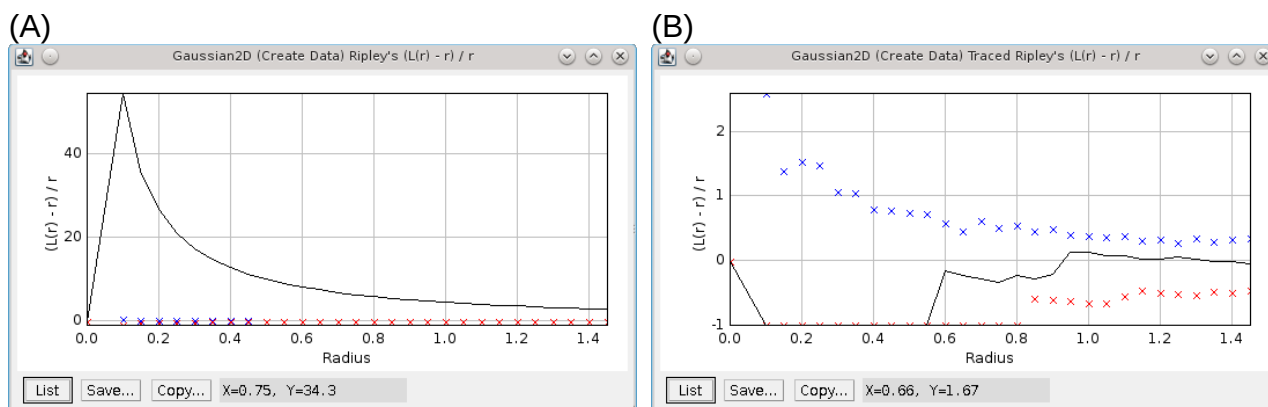
| Score            | Description  |
|------------------|--|
| DENSITY          | The count of the number of localisations in the region around a localisation |
| $K_i(r)$         | K score. Should be equal to the local region area                            |
| $K_i(r)/Area$    | Should be equal to 1   |
| $L_i(r)$         | Should be equal to the radius  |
| $L_i(r) - r$     | Should be equal to zero  |
| $L_i(r)/r$       | Should be equal to one. Scores should be comparable across different radii   |
| $(L_i(r) - r)/r$ | Should be equal to zero. Scores should be comparable across different radii  |

## 7.5.2 Ripley's L-plot

If the COMPUTE L PLOT option is selected then the plugin will compute the L-score for a range of radii. A dialogue is presented allowing the user to select the MIN RADIUS and MAX RADIUS and the INCREMENT used to move between them. A plot is produced showing the  $(L(r) - r)/r$  score for each radius. The score should be equal to zero for homogeneous data and is comparable across radii.

If the CONFIDENCE INTERVALS parameter is enabled the plugin will create 99 random simulations of the given sample size in the same sample area. The  $(L(r) - r)/r$  score is computed for each simulation and the upper (blue) and lower (red) bounds of the confidence intervals are shown on the chart. If the actual data values lies outside the confidence intervals then it is a non-random distribution with 99% confidence.

An example of Ripley's L-plot is shown in Illustration 47. The figure shows the L-plot for simulated localisations of fluorophores. The raw data show significant deviation from a random distribution (47A). Tracing of the localisations with a suitable time and distance threshold collates the individual localisations into single molecule traces. These do not show any deviation from a random distribution (47B).



**Illustration 17:** Ripley's L-plot shown for two datasets with 99% confidence intervals, upper limit (blue) and lower limit (red). (A) Data showing a significantly non-random distribution. Data was generated by simulating randomly placed fixed position fluorophores with average  $t$ -on times of 3 frames. Localisations are expected to be non-random. (B) The same data following tracing with  $t=10$  frames,  $d=0.3$  pixels. Localisations have been collated into single molecules and the traced data do not show deviation from a random distribution.

## 7.6 Dark Time Analysis

The **DARK TIME ANALYSIS** plugin computes a dark time histogram for blinking fluorophores and then outputs the time threshold required to capture a specified percentage of the blinks.

Fluorophores can be inactive (dark) for a variable amount of time between fluorescent bursts. If tracing is to be used to connect all separate bursts from the same fluorophore into a single molecule then the tracing must be done using the maximum dark time expected from the fluorophore. This can be estimated using the **DARK TIME ANALYSIS** plugin, ideally on an image sample of fluorophores captured under the same imaging conditions as will be used for *in vivo* experiments. The most success will be obtained using fixed fluorophore samples.

The plugin performs tracing or clustering on the localisations using a specified search distance and the maximum number of frames in the results for the time threshold. This allows the algorithm to connect a localisation in the first frame to the last frame. All the algorithms only allow localisations to be joined to the closest localisation in a different frame (i.e. not the same frame). However only those closer than the distance threshold are joined.

Illustration 18 shows the plugin dialogue.



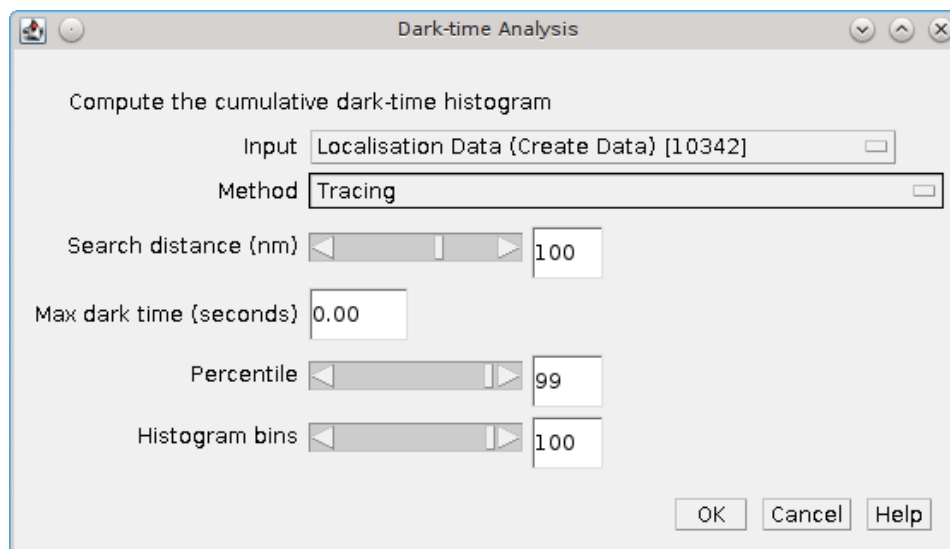


Illustration 18: Dark Time Analysis plugin dialogue

The plugin supports the following parameters:

| Parameter               | Description   |
|-------------------------|---|
| INPUT                   | Specify the results to use for analysis   |
| METHOD                  | Specify the tracing or clustering method. Details of the tracing and clustering algorithms can be found in the sections describing the Trace Molecules and Cluster Molecules plugins. |
| SEARCH DISTANCE (NM)    | Specify the maximum distance between localisations to be part of the same molecule  |
| MAX DARK TIME (SECONDS) | Specify the maximum allowed dark time between localisations to analyse. Any dark time distances above this are ignored. Set to zero to allow any dark time.                           |
| PERCENTILE              | Specify the percentile limit used to report the maximum dark-time   |
| HISTOGRAM BINS          | Specify the number of bins to use for the histograms  |

When started the progress of the algorithm is shown on the ImageJ task bar. When tracing/clustering is complete the plugin produces a histogram and cumulative histogram of the dark time for all traces. The dark-time corresponding to the specified percentile is then reported to the ImageJ log window.

## 7.7 Blink Estimator

Estimate the blinking rate of fluorophores in a results set.

The **BLINK ESTIMATOR** uses the method of Annibale *et al* (2011) to estimate the blinking rate of a fluorophore. The method is based on performing trace analysis of the localisations using different time thresholds. The distance threshold used for the trace analysis is set using a factor of the average precision of all the localisations in the sample. When the time

threshold is 1 then the number of traces will be equal to the total number of fluorescent bursts in the sample. As the time threshold is increased the number of traces should asymptote at the number of molecules in the sample, assuming that no missed localisation counts occur. The curve of traces against time threshold can be fitted using the following formula:

$$N(t_d) = N \cdot \left( 1 + n_{\text{blink}} \cdot e^{\left( \frac{1-t_d}{t_{\text{off}}} \right)} \right)$$

where  $t_d$  is the dark time of the fluorophore (time-threshold),  $N(t_d)$  is the number of traces at  $t_d$ ,  $n_{\text{blink}}$  is the average number of blinks per fluorophore,  $t_{\text{off}}$  is the average off time for fluorophores. To avoid the problems associated with missed counts Annibale *et al* only fit this formula to the first five data points of the curve.

The **BLINK ESTIMATOR** computes the  $N(t_d)$  curve and fits the curve to the specified formula. By default the plugin fits the curve and shows a plot of the  $N(t_d)$  curve with the fitted line (see Illustration 19). The fit parameters are reported to the ImageJ log window.

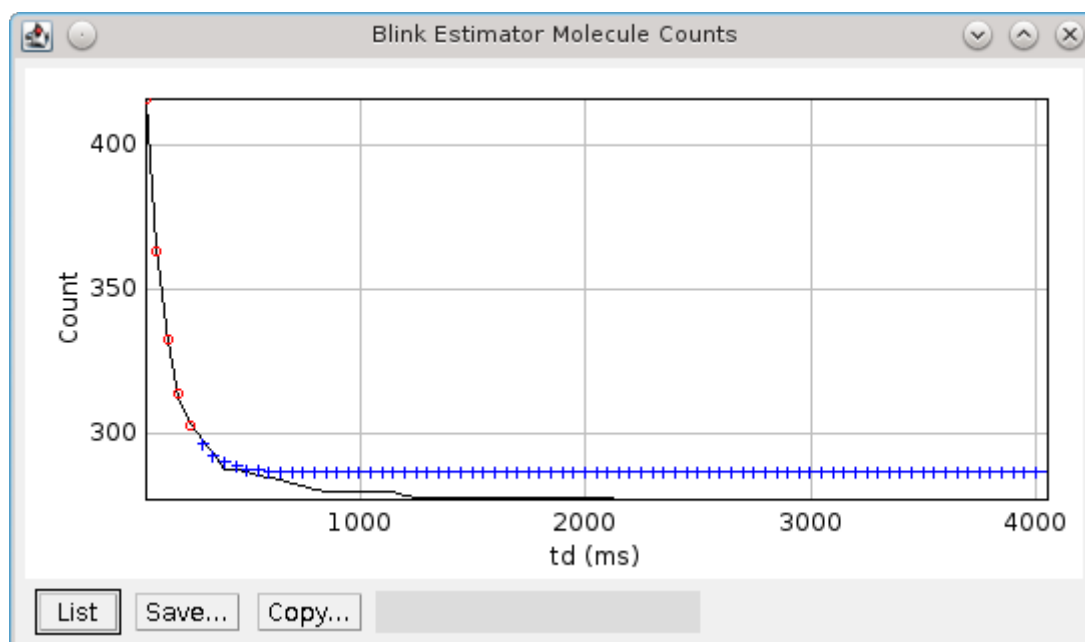
Alternatively the plugin can fit the curve multiple times using a different number of points for each fit. In this case the output is a set of plots showing the values of each of the fitted parameters against the number of fitted points.

Note that tracing localisations through time also requires a distance threshold to determine if the two coordinates are the same molecule. The distance threshold used for the analysis is computed using the average precision of the localisations multiplied by a search factor. The average precision is determined by fitting a skewed Gaussian to a histogram of the localisation precision. This fit can optionally be displayed.

The following parameters can be set within the plugin to control the output:

| Parameters             | Description   |
|------------------------|---|
| INPUT                  | Select the input results set  |
| MAX DARK TIME          | The maximum dark time to use for the $N(t_d)$ plot  |
| HISTOGRAM BINS         | The number of histogram bins  |
| SHOW HISTOGRAM         | Show the histogram of the precision along with the fitted skewed Gaussian curve   |
| SEARCH DISTANCE        | Define the distance threshold for trace analysis  |
| RELATIVE DISTANCE      | If selected the <b>SEARCH DISTANCE</b> is relative to the average precision of the localisations in the dataset.<br>If unselected the <b>SEARCH DISTANCE</b> is absolute (in pixels). |
| FITTED POINTS          | The number of points to fit on the $N(t_d)$ curve. Must be 4 or more  |
| RANGE OF FITTED POINTS | The maximum number of additional points to fit for variation analysis. Fit are computed from [FITTED POINTS] to [FITTED POINTS + RANGE]. Set to zero to show the $N(t_d)$ curve       |

| Parameters          | Description  |
|---------------------|--|
| TIME AT LOWER BOUND | <p>When constructing the dark time curve the number of frames used for tracing is converted to time in milliseconds.</p> <p>Set to true to output time as <math>nFrames \times msPerFrame</math>.</p> <p>Set to false to output time as <math>(nFrames+1) \times msPerFrame</math>.</p> <p>This setting will shift the time axis of the curve and will produce different curve fitting results. Simulations using the <b>CREATE DATA</b> plugin suggest that this should be TRUE.</p> <p>A value of false is stating that localisations in consecutive frames may have blinked into a dark state of less than 1 frame.</p> |



*Illustration 19:  $N(t_d)$  curve output from the **BLINK ESTIMATOR**. The curve data is plotted in black. The red circles show the fitted points. The blue crosses indicate where the fitted line will asymptote.*

## 7.8 Neighbour Analysis

Saves all localisations paired with their neighbour (if present) to file.

The **NEIGHBOUR ANALYSIS** plugin searches for the neighbour within the given time and distance thresholds with the highest signal. These are then saved to file as a series of clusters. Each cluster has either one or two entries. The first contains the details of the localisation, the optional second entry contains the details of the neighbour.

This plugin was written to allow off-line analysis of the neighbours of localisations, for example investigation of how average distance, signal strength and frequency of neighbours vary when localisations are categorised using their own signal strength.

## 7.9 Filter Analysis

Performs filtering on a set of categorised localisation results and computes match statistics for each filter.

The `FILTER ANALYSIS` plugin is used to benchmark different methods for filtering the results of fitting a 2D Gaussian to an image. When a Gaussian peak is fitted to an image it has many parameters that are changed in order to optimise the fit of the function to the data. For example the centre of the spot, the width of the spot, etc. It can be difficult to know if the parameters have been adjusted too far and the Gaussian function is no longer a fair representation of the image. This plugin requires the user to prepare input data with fitting results identified as correct or incorrect. The plugin will load the results and then filter them with many different settings. The filtered results are compared with the classifications and various comparison metrics are generated allowing the best filter to be identified.

### 7.9.1 Input Data

The plugin requires a folder containing the results of running `PEAK FIT` on one or more images. The results will contain the X and Y coordinates, X and Y standard deviations and amplitude of the Gaussian function. The results will also contain the original pixel value of the peak candidate location. This column must be updated to contain a zero for any result that is incorrect and a non-zero value for any result that is correct. The classification of spots can be performed using any method.

One successful data preparation method employed by the authors was to manually inspect a set of super-resolution images and extract entire stacks from the image containing suitable spots. These stacks should contain both on and off frames for the fluorophore and examples of good, medium and poor spots that could still be identified manually. The example spots were then manually scored by multiple researchers and a jury system used to score if the frame contained a spot or not. The images were fitted using `PEAK FIT` and the results files updated using the jury classification to mark results as correct/incorrect.

When the plugin is run the user must select a directory. The plugin will attempt to read all the files in the directory as `PeakResults` files. The results are cached in memory. If the same directory is selected the user can opt to re-use the results. Click 'No' to re-read the results, for example if the files have been modified with new classifications.

### 7.9.2 Available Filters

When the results have been loaded the user is presented with a dialogue as shown in Illustration 20. There are several filters available with parameters to control them. Enable the filter by using the checkbox for the filter name. Each filter typically has a set of parameters that control the filter. The plugin allows the parameters to be enumerated so that the performance of the filter can be tested. The plugin dialogue will allow the minimum and maximum value for the filter to be set along with an increment used to move between them. The filter will be run for each combination of parameters that is to be enumerated to produce a filter set.



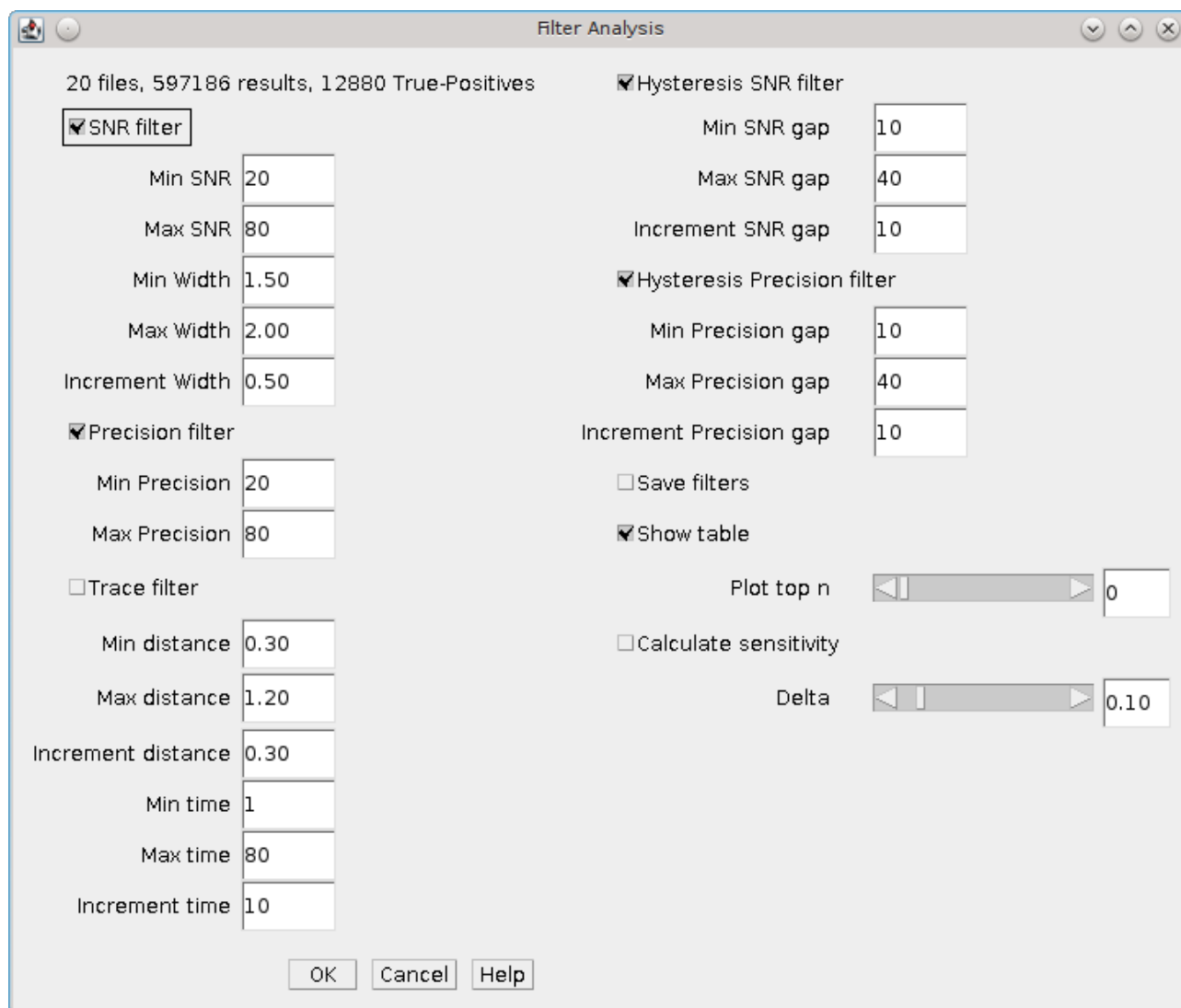


Illustration 20: Filter Analysis plugin dialogue

The following filters are available:

| Filter           | Description  |
|------------------|--|
| SNR FILTER       | Remove all results with a Signal-to-Noise Ratio (SNR) below a set value. Additionally remove peaks where the width has expanded above a set ratio from the estimated PSF width.<br>The SNR ratio used is from MIN SNR to MAX SNR in integer increments. The width used is from MIN WIDTH to MAX WIDTH in increments of INCREMENT WIDTH |
| PRECISION FILTER | Remove all results with a precision below a set value.<br>The precision used is from MIN PRECISION to MAX PRECISION in integer increments  |

| Filter                      | Description   |
|-----------------------------|---|
| TRACE FILTER                | <p>Remove all results that do not form part of a molecule trace using the specified time and distance threshold. See TRACE MOLECULES for details of the tracing algorithm.</p> <p>The minimum and maximum time and distance threshold can be configured along with the increments used to move between the min and max</p>  |
| HYSTERESIS SNR FILTER       | <p>Mark all results with a Signal-to-Noise Ratio (SNR) above an upper level as valid. Remove all results with a Signal-to-Noise Ratio (SNR) below a lower level. Any result with a SNR between the upper and lower levels is a candidate result. Candidates are retained if they can be traced directly or via other candidates to a valid result.</p> <p>Uses the same SNR and width parameters from the SNR FILTER iterating over them as before. The SNR is used to set the lower SNR limit. Contains additional parameters to set the gap to the upper limit. The user can configure the min and max gap and the increment to move between them</p> |
| HYSTERESIS PRECISION FILTER | <p>Mark all results with a precision below a lower level as valid. Remove all results with a precision above an upper level. Any result with a precision between the upper and lower levels is a candidate result. Candidates are retained if they can be traced directly or via other candidates to a valid result</p> <p>Uses the same precision parameters from the PRECISION FILTER to set the lower limit. Contains additional parameters to set the gap to the upper limit. The user can configure the min and max gap along with the increment to move between them</p>  |

Note that the hysteresis filters perform tracing using a time threshold of 1 and a distance threshold of 2x the average precision of the results.

More flexibility over the filters to use in the analysis is provided by using the CREATE FILTERS and FILTER ANALYSIS (FILE) plugins (see the following sections).

### 7.9.3 Additional parameters

After the filters have been configured there are additional parameters controlling the analysis:

| Parameter    | Description   |
|--------------|---|
| SAVE FILTERS | Save the filters to file. The filters can be read using the FILTER ANALYSIS (FILE) plugin |
| SHOW TABLE   | Show a table of the analysis results  |

| Parameter             | Description  |
|-----------------------|--|
| PLOT TOP N            | Produce a plot of the top N filter sets showing their Jaccard score verses a named parameter value. If a filter set contains filters with different named parameters then the filters are plotted in sequential order. |
| CALCULATE SENSITIVITY | Calculate how sensitive the scoring metric is to a change in each filter parameter. Each filter parameter is adjusted by the delta and the gradient computed at the point of the optimum filter score                  |
| DELTA                 | The delta value used to adjust parameters to calculate the sensitivity   |

## 7.9.4 Output

At least one output method must be chosen otherwise the plugin will show an error message.

### 7.9.4.1 Table output

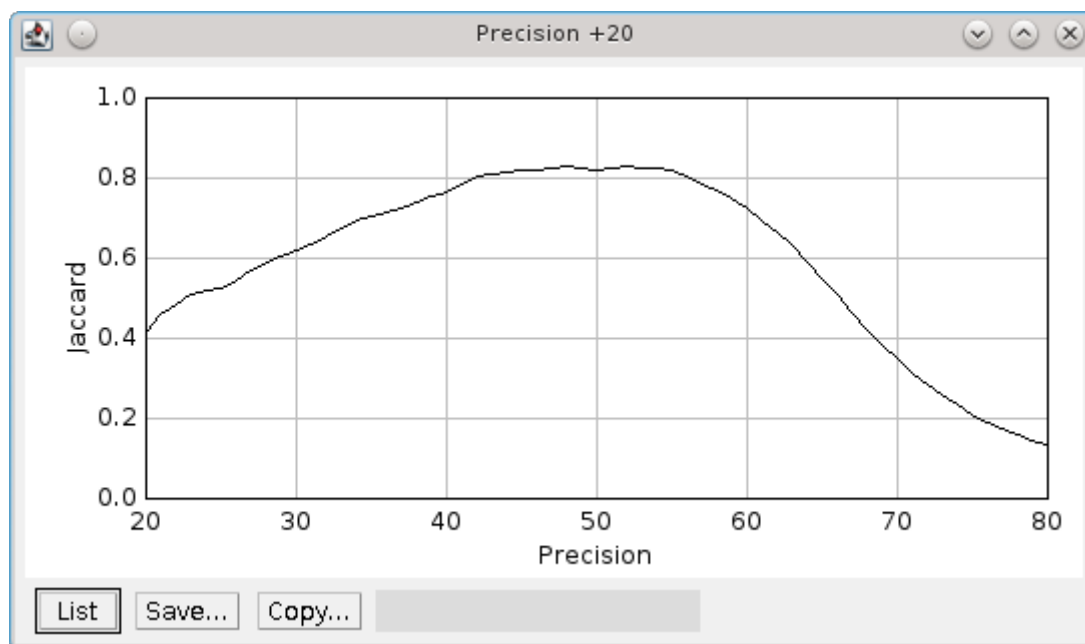
If `SHOW TABLE` is selected the plugin produces a table of the match statistics for the filtered results compared to the categorised inputs. The results for all the filters in a filter set are shown together. An empty row is used to separate filter sets.

The available scores are Precision, Recall, F-score and Jaccard. Details of the match statistics are given in Appendix B : Comparison Metrics.

### 7.9.4.2 Plot output

If a value for `PLOT TOP N` is provided the plugin produces a chart of the Jaccard score against one independent filter variable for each of the top N performing filter sets. An example of the `HYSTERESIS PRECISION` filter chart is shown in Illustration 47. The chart shows a clear improvement and then fall-off in performance as the lower threshold is increased.





*Illustration 21: Plot of the Jaccard score versus lower precision limit for the `HYSTERESIS PRECISION` filter. The upper limit for hysteresis thresholding is 20nm higher than the lower limit (+20).*

### 7.9.4.3 Sensitivity analysis

If `CALCULATE SENSITIVITY` is selected the plugin produces a table of the sensitivity of each parameter for each unique filter (identified by name). The sensitivity is calculated for the filter parameters that produce the highest Jaccard score. The sensitivity is the partial gradient of the score with respect to each parameter.

If the sensitivity is low then the filter will perform well using a range of parameter values. If the sensitivity is high then the filter performance will rapidly fall-off as the optimum parameter value is changed. Ideally a good filter will have good performance metrics and low sensitivity across all the parameters. It may be better to choose a filter that does not have the highest Jaccard score but has the lowest sensitivity. This filter should perform better on a range of input data.

The sensitivity is calculated by altering each parameter by a relative amount. The delta parameter controls the update magnitude. Note that integer parameters will be correctly adjusted to the next integer. The score is recomputed using the new parameter value following a positive or negative change. The average rate of change of the score is then computed. The sensitivity is output for the Jaccard, Precision and Recall scores. Two scores are output: (a) the raw average change, titled `SENSITIVITY (DELTA)`; or (b) the average change divided by the change in the parameter value, titled `SENSITIVITY (UNIT)`.

## 7.10 Create Filters

The `CREATE FILTERS` plugin can be used to prepare a large set of filters for use in the `FILTER ANALYSIS (FILE)` plugin (see section 7.11). The purpose is to create a set of filters that can be applied to a testing dataset to identify the best filter.

The plugin dialog shows a text area where a filter template can be specified as shown in

Illustration 22. If the `SHOW DEMO FILTERS` checkbox is selected then several example filters will be recorded in the ImageJ log.

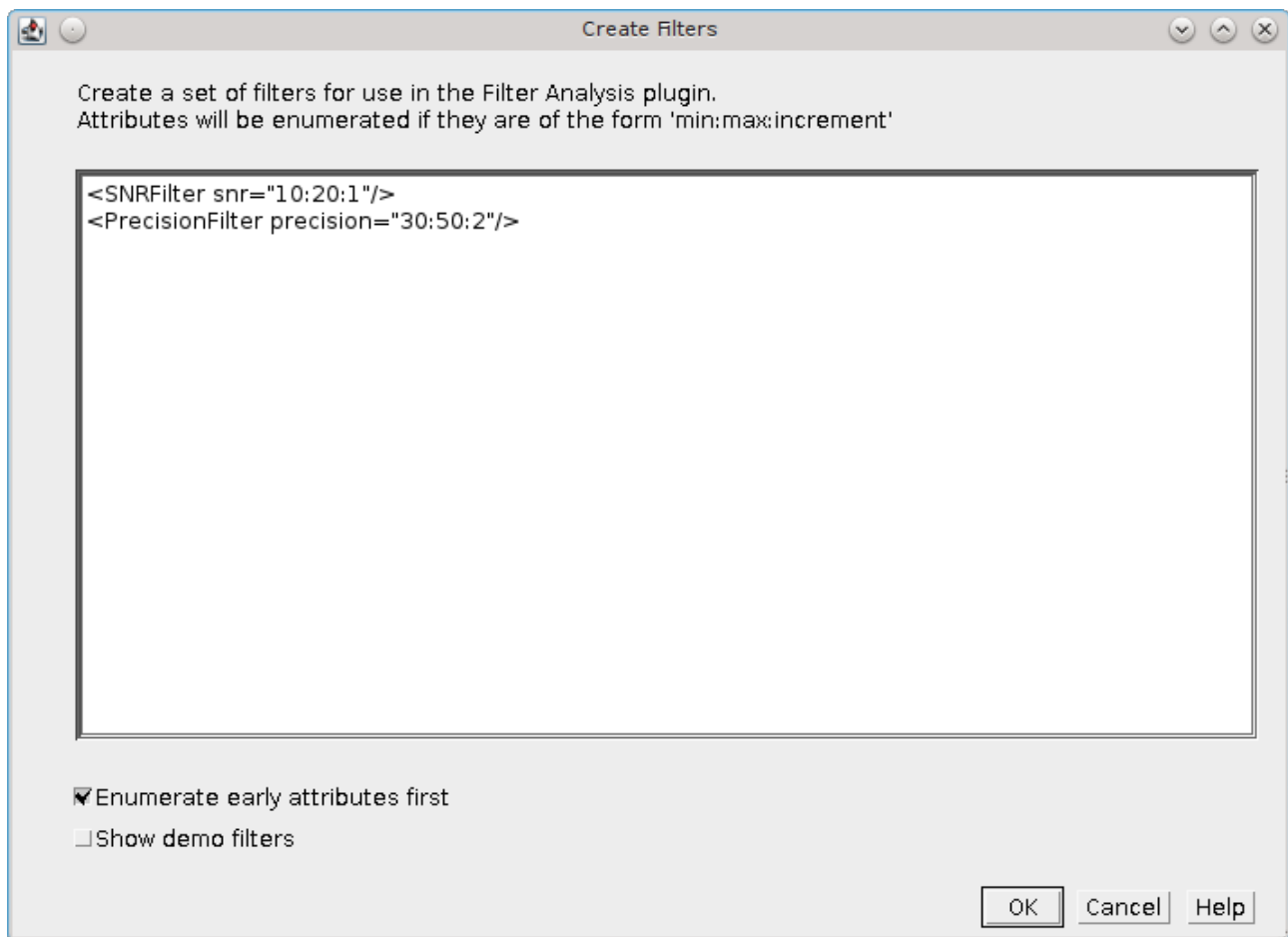


Illustration 22: Create Filters plugin dialogue

The filter template must be valid XML elements. Each element should take the format of valid XML filters that can be passed to the `FREE FILTER RESULTS` plugin. An example set of all the valid filters can be output to the ImageJ log window using the `FREE FILTER RESULTS` plugin (see section 6.10).

Each XML filter template is processed into a filter set. The attributes of the filter are scanned for the pattern “min:max:increment”. If this is found a filter is created with the attribute set at each value from the minimum to the maximum using the specified increment. If not then the filter value is left untouched.

If a filter has multiple attributes then all combinations will be enumerated. Use the `ENUMERATE EARLY ATTRIBUTES FIRST` checkbox to enumerate the attributes in the order they appear in the template. Otherwise they will be enumerated in reverse order.

The output filter set will be named using the name of the filter. For example:

```
<SNRFilter snr="10:20:2"/>
```

Becomes:

```
<FilterSet name="SNR">
  <filters class="linked-list">
    <SNRFilter snr="10"/>
```

```
<SNRFilter snr="12"/>
<SNRFilter snr="14"/>
<SNRFilter snr="16"/>
<SNRFilter snr="18"/>
<SNRFilter snr="20"/>
</filters>
</FilterSet>
```

When the plugin has generated all the filters it will present a file selection dialogue allowing the user to specify the output filter file. If the file already exists it will be overwritten.

### 7.11 Filter Analysis (File)

Performs filtering on a set of categorised localisation results and computes match statistics for each filter.

The **FILTER ANALYSIS (FILE)** plugin is similar to the **FILTER ANALYSIS** plugin (see section 7.9). The difference is that the plugin will not present options to create filters in the main dialogue. Instead the filters are created from an input file that is selected following the main dialogue. All other functionality is identical.

The filter file can specify thousands of different filters which are described using XML. This allows combinations of filters to be specified using And and Or filters; this is something that is not possible with the **FILTER ANALYSIS** plugin. A filter file can be generated from a filter template by enumerating the filter parameters using the **CREATE FILTERS** plugin (see section 7.10).

### 7.12 Spot Analysis

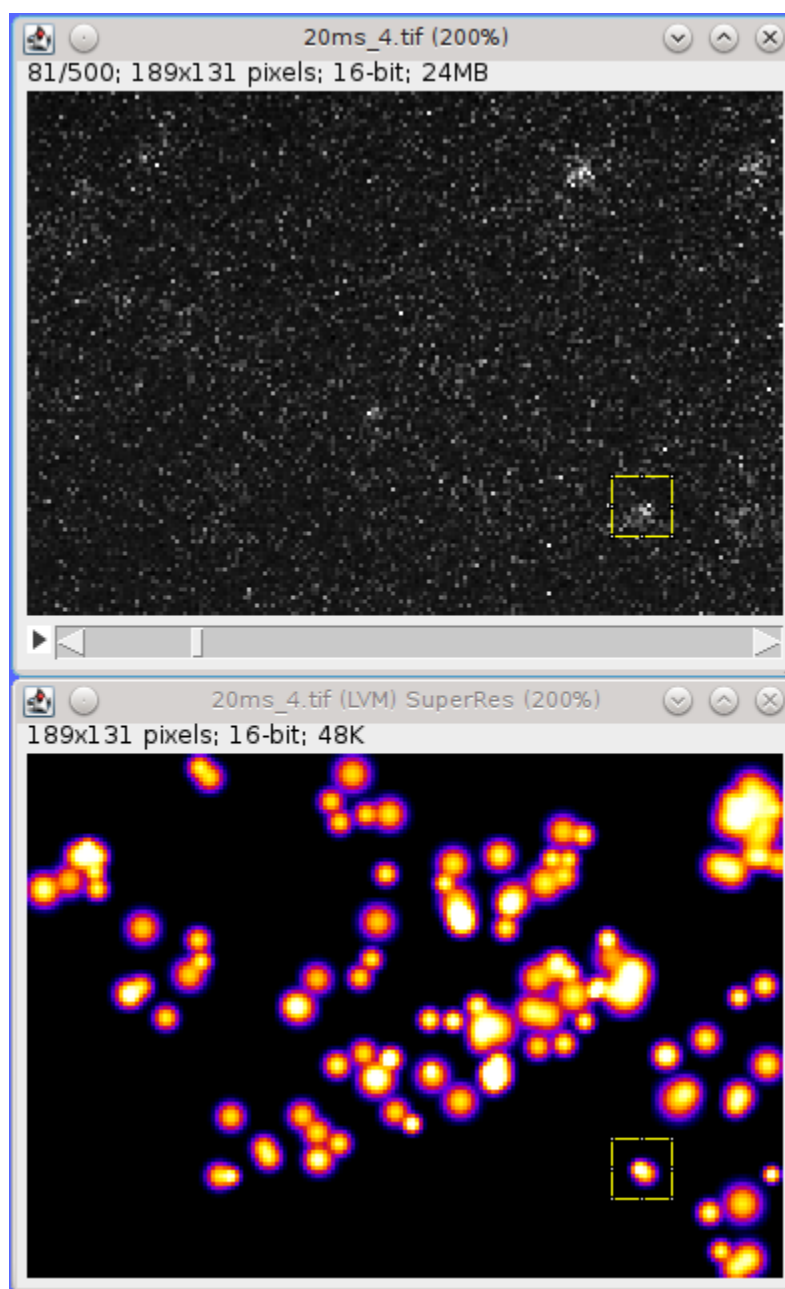
The **SPOT ANALYSIS** plugin is designed to allow analysis of fluorophore blinking kinetics using calibration images.

The plugin allows the user to extract an ROI outlining a fluorophore position from an image and manually mark the frames where a spot is visible. A trace of the image intensity helps the user choose the brightest frames where a spot may be present. The plugin records the signal, on-time and off-time for the fluorophore to files for later analysis.

The plugin has been used to generate distributions for the signal-per-frame, on-times and off-times for mEOS3 fluorophores captured at a frame rate of 50 frames a second (20ms exposure time). At this rate the fluorophore emissions are very faint but frames can be manually labelled as on (or off). By analysing hundreds of spots it is possible to extract distribution parameters that allow the fluorophores to be modelled using the **CREATE DATA** plugin.

#### 7.12.1 Input Images

The purpose of the plugin is to mark the fluorescent bursts from a single fluorophore. Consequently the input images should ideally be of single fluorophores spread evenly on the image and fixed in position. Overlapping fluorophores can be excluded from analysis by only selecting spots that appear distinct. An example input image with a simulated PSF image generated using the **PEAK FIT** plugin is shown in Illustration 23. A single spot has been selected with an ROI on the PSF image. The selection has been applied to the original image to select the region for analysis.



*Illustration 23: Example Spot Analysis input image of mEOS3 expressed in *S. pombe*, fixed and activated at a low power to produce non-overlapping fluorophore bursts. The super-resolution image is a simulated PSF image of localisations with a SNR above 100. A single fluorophore is selected for analysis.*

### 7.12.2 Plugin Interface

The Spot Analysis plugin displays a window as shown in Illustration 24. The window contains a drop down selection list allowing the input image to be selected and various parameters for the analysis can be set. The centre of the window shows the analysis of the current frame from the spot regions. The window then has several buttons allowing different actions. The final region of the window is a list of the currently selected on-frames

with the calculated signal (in photons).

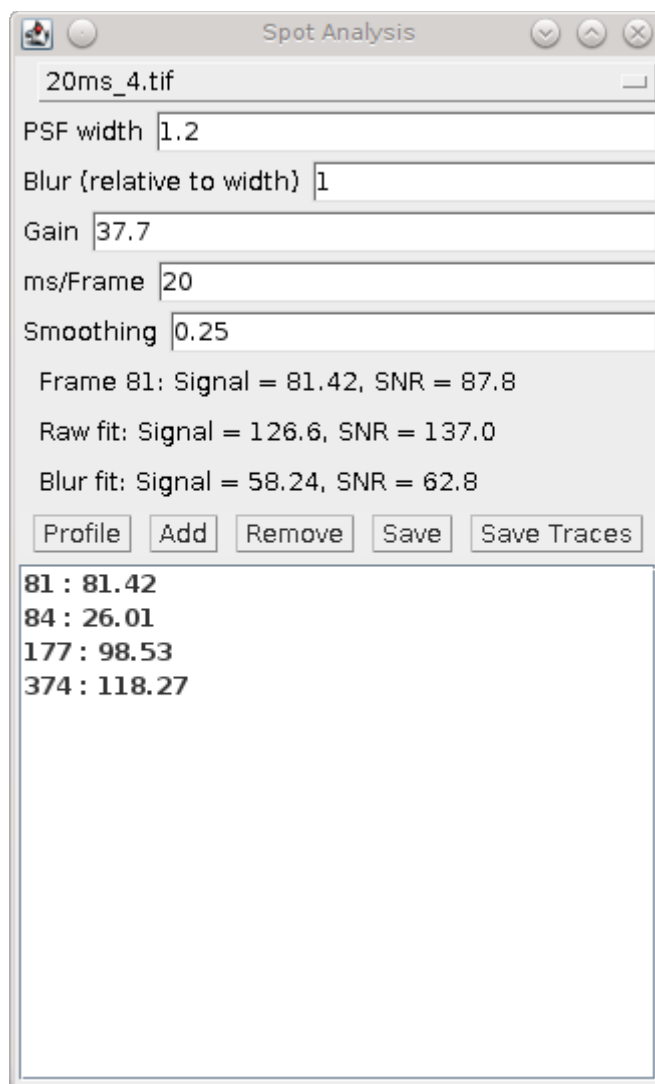


Illustration 24: Spot Analysis plugin window

The following parameters can be set:

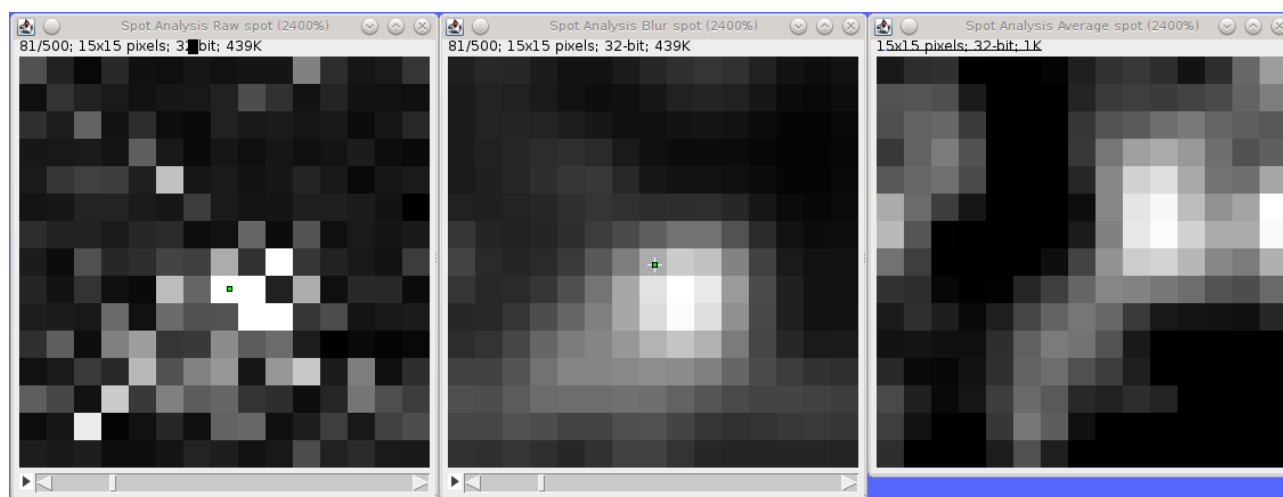
| Parameters            | Description   |
|-----------------------|---|
| PSF <sub>WIDTH</sub>  | The estimated peak width of the spots (in pixels). This is used to fit the spot region using Peak Fit |
| BLUR                  | The Gaussian blur to apply to create the blurred spot image. The width is relative to the PSF width   |
| GAIN                  | The total gain of the system (ADUs/photon). This is used to convert the counts to photons             |
| MS/F <sub>FRAME</sub> | The exposure time. Used to calculate the on/off times in milliseconds                                 |

| Parameters | Description   |
|------------|---|
| SMOOTHING  | The LOESS smoothing parameter used to create the smoothed average for the profile plots |

### 7.12.2.1 Profiling a spot region

The first action is to select a spot on the input image using a rectangular ROI. It is difficult to see spots on the input image which has many frames and very few fluorescent bursts. Consequently it is recommended that an overview image is created to allow the spots to be identified. This could be a maximum or average intensity projection of the input image. However it is best to identify spots using the **FIND PEAKS** plugin and then reconstruct an image using the fitted PSFs. If the image scale is set to 1 then the reconstructed image is the same size as the input image. This allows spots to be outlined on the PSF image and then the ROI reapplied to the original image. An example of this is shown in Illustration 23.

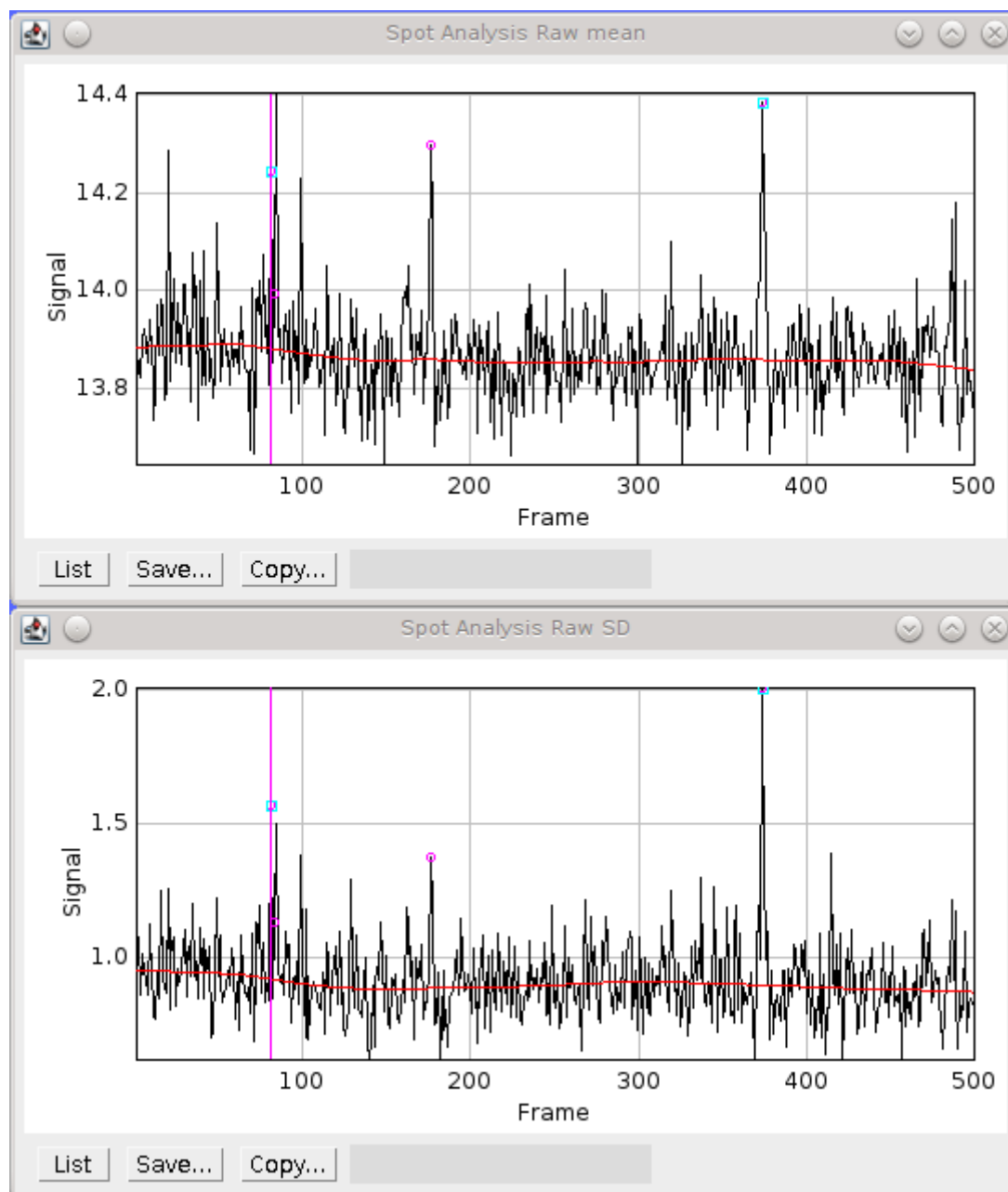
When a suitable spot has been outlined the region can be analysed using the **PROFILE** button. The analysis extracts the region into a new image named **SPOT ANALYSIS RAW SPOT**. This is magnified to allow easier viewing. The same region is subjected to a Gaussian blur and extracted to a new image name **SPOT ANALYSIS BLUR SPOT**. The blurred image can make it easier to see peaks in very noisy images. In addition an average intensity projection is created for the spot region and named **SPOT ANALYSIS AVERAGE SPOT**. The average image can indicate if the spot is fixed and central, moves around the image or overlaps with other signal. An example of the raw, blur and average images are shown in Illustration 25. The average spot image shows a blurred spot biased to the upper-right corner but the signal is distinct so the spot is suitable for analysis. The **FIND PEAKS** fitting algorithm is applied to the current frame of the raw and blurred spot (using the configured PSF width). If a spot is detected within 50% of the distance to the centre of the frame it is marked on the image using an cross overlay.



*Illustration 25: Spot Analysis raw spot, blurred spot and average spot images. The centre of the fitted peak for the raw and blurred image for the current frame is shown using a cross overlay.*

The plugin then analyses the spot region and creates profile plots of the mean and the signal standard deviation per frame as shown in Illustration 26. The currently selected

frame in the spot images is shown using a vertical magenta line. The profile can be very noisy and so a LOESS smoothing is applied to the raw plot data to produce a background level (shown in red). Any selected on frames are labelled using magenta circles. In addition the plugin searches the FIND PEAKS results that are held in memory for any results that were created from the input image. If so then these are potential candidates for on-frames and are shown as cyan squares.



*Illustration 26: Spot Analysis mean and standard deviation profiles of the spot region. The current frame is shown as a vertical pink line. Candidate on-frames from Peak Fit results are shown as cyan box. Selected on-frames are shown as magenta circles. The red line is the LOESS smoothing of the plot data.*

### 7.12.2.2 Adding on-frames

Any fluorophore on-frame should have a higher mean than the frames around it. These can be easily viewed as spikes on the plot profile and the relevant frames in the spot image inspected.

When the spot image is moved to a new frame the plugin will update the signal and SNR figures for the frame. The signal is calculated as the sum of the raw spot minus the LOESS smoothing fit of the mean profile, i.e. the background. The noise is calculated using the LOESS smoothing fit of the standard deviation profile.

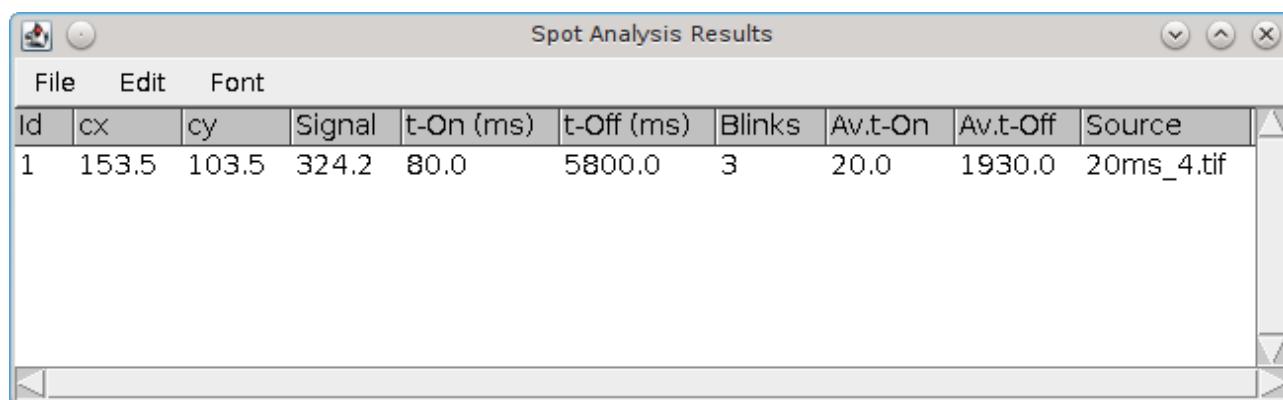
The plugin also runs the **PEAK FIT** algorithm on the raw and blurred spot image. Identified localisations will be marked on the image with an overlay. The signal and SNR for the raw fit and blur fit are shown in the plugin window.

Combining inspection of the spot images with consideration of the signal and SNR of the raw data and fitted data should allow the user to decide if a fluorophore is on in the selected frame. The frame can be added to the on-frames list using the **ADD** button. This can also be performed using a shortcut key mapped to the **SPOT ANALYSIS (ADD)** plugin (see section 7.13).

The frame is added to the list in the **SPOT ANALYSIS** window along with the current estimate of the signal for that frame. Note that LOESS smoothing only uses frames that have not been selected as on-frames. Consequently the background and the noise are updated as more on-frames are added to the list.

### 7.12.2.3 Saving a fluorophore sequence

When all the on-frames have been marked the fluorophore sequence must be saved using the **Save** button. This will record the sequence to a summary window as shown in Illustration 27. The window shows the fluorophore signal, on and off times and number of blinks.



The screenshot shows a window titled "Spot Analysis Results" with a menu bar (File, Edit, Font) and a table. The table has 10 columns: Id, cx, cy, Signal, t-On (ms), t-Off (ms), Blinks, Av.t-On, Av.t-Off, and Source. There is one row of data with the following values: 1, 153.5, 103.5, 324.2, 80.0, 5800.0, 3, 20.0, 1930.0, and 20ms\_4.tif.

| Id | cx    | cy    | Signal | t-On (ms) | t-Off (ms) | Blinks | Av.t-On | Av.t-Off | Source     |
|----|-------|-------|--------|-----------|------------|--------|---------|----------|------------|
| 1  | 153.5 | 103.5 | 324.2  | 80.0      | 5800.0     | 3      | 20.0    | 1930.0   | 20ms_4.tif |

*Illustration 27: Spot Analysis summary window*

The **SAVE** button can only be pressed once to prevent duplicates in the window. If the sequence list is updated by adding or removing on-frames then the listed can be re-saved. Note that the old sequence will remain in the summary window and the new sequence will have a new ID. The old sequence should be removed using the summary tables **EDIT > CLEAR** command.



### 7.12.3 Analysis Workflow

The following workflow is recommended when using the SPOT ANALYSIS plugin.

1. Open the next input image.  
Magnify the image if necessary (IMAGE > ZOOM > IN [+])
2. Run the PEAK FIT plugin to identify peaks with a SNR above 100 (or other criteria to show clear spots).  
Create a PSF image output with a scale of 1
3. Select a spot on the PSF image using a new rectangle ROI (or moving the previous one). It is recommended to maintain the same ROI size throughout the analysis.  
Choose a spot that does not overlap any other spots
4. Select the input image and press CTRL+SHIFT+E to restore selection (apply the same ROI) to the input image. This will outline the spot location on the input image
5. Press the PROFILE button. This will produce a profile of the image intensity and standard deviation per frame for the spot. Any localisation results from the PEAK FIT plugin are marked on the profile. The PROFILE command will also extract the ROI to a new detail image and zoom in for easier viewing. A blurred version of the image will also be shown using a Gaussian blur to make the spots more uniform. Contrast may have to be adjusted in these images.  
The images will be set to the first frame containing a localisation fitting result (or frame 1 if none are available)
6. Scroll left and right through the image using the left/right keys to locate spots. (The original and blurred images will be synchronised.)
7. Click the ADD button when a spot is visible in the frame. Alternatively use the keyboard shortcut that has been mapped to the SPOT ANALYSIS (ADD) command. The frame will be added to the list with the signal estimate for the spot
8. Remove bad frames from the list by highlighting them and using the REMOVE button. Double-clicking a frame in the list will select that frame on the detail images
9. Save the frames listed in the window to the results summary by clicking the SAVE button. The summary shows the fluorophore signal, on and off times and number of blinks. The list can only be saved once. It must be updated (ADD/REMOVE) before it can be saved again
10. Repeat steps 3-9 for different candidate spots
11. Save all the results to file using the SAVE TRACES button.  
Only results that are present in the table are saved. Duplicate or unwanted results can be removed using the table EDIT > CLEAR command to remove selected rows

### 7.12.4 Results Files

Fluorophore sequences in the summary table can be saved to file. The plugin saves only the sequences within the table. Any sequence that is removed from the table using the EDIT > CLEAR command will be discarded. Click the SAVE TRACES button to open a folder selection dialogue. Following selection the plugin will write several files to the directory. Any old files will be over-written. The following files are saved:

| File       | Description  |
|------------|--|
| TRACES.TXT | Contains a summary of each fluorophore. Each record shows the ID, centre X and centre Y of the ROI outlining the spot, the total signal, the number of blinks and then a chain of paired start and stop frames for each pulse.<br><br>This file contains all the information of the other files except the signal-per-frame (see SIGNAL.TXT) |
| TOn.TXT    | Contains the on-times for all the results  |
| TOff.TXT   | Contains the off-times for all the results   |
| SIGNAL.TXT | Contains the signal for each frame. The frame and signal are shown separated by a space.<br><br>A header line is added before the signal for each fluorophore showing the ID, number of blinks and the total signal  |
| BLINKS.TXT | Contains the number of blinks for all the results. Blinks are the number of dark times recorded for a fluorophore. It is one less than the number of pulses (on-time events)   |

### 7.13 Spot Analysis (Add)

This plugin provides a named plugin command for the **ADD** button of the **SPOT ANALYSIS** plugin.

Any named plugin command can be mapped to a hot key using ImageJ's **PLUGINS > SHORTCUTS > CREATE SHORTCUT...** command. Thus the **ADD** button can be mapped by ImageJ to a keyboard shortcut. This allows the user to scroll through an image generated by the **SPOT ANALYSIS** plugin using the standard left and right arrow keys to move between frames. When a spot is present the user adds the spot to the list by clicking the **ADD** button on the **SPOT ANALYSIS** window. If the **ADD** command is mapped to a shortcut the user can perform the same action using the hotkey. This allows faster analysis of images using only simple keyboard commands.

### 7.14 Fourier Image Resolution

Calculate the resolution of an image using Fourier Ring Correlation (FRC).

This plugin is based on the **FIRE** (Fourier Image RESolution) plugin produced as part of the paper by Niewenhuizen, *et al* (2013). Changes to the original functionality include: additional options for computation of the FRC curves; improved determination of FRC line crossing points; speed optimisations; and support for multi-threaded random repeats. Additional functionality for correction of spurious correlations has been added based on Matlab code kindly provided by Prof. Bernd Reiger (corresponding author of the **FIRE** paper).

Fourier Ring Correlation computes the correlation between the features of two images at a certain spatial frequency. A 2D image is transformed into the frequency domain using a

Fourier transform. The frequency image represents the amplitude and phase of the original data at different frequencies. These can be visualised as a 2D image, the centre of the image represents the lowest frequencies moving to the highest frequencies at the edge of the image. Points taken from a ring drawn around the centre will represent all the amplitude and phases present at a single frequency. Two images can be compared at a given frequency by measuring the correlation of the data sampled from the same ring on both images. This is the Fourier Ring Correlation (FRC).

To compute the image resolution of a set of localisations requires the following steps:

- Produce two super-resolution images using the localisation data
- Transform the two images to the frequency domain
- Compute the FRC between the two images at all the available frequencies
- Plot the FRC against the frequency. This will gradually fall to zero as the frequency decreases (see Illustration 28)
- Smooth the raw FRC curve using robust local regression (LOESS)
- Use the smoothed FRC curve to look-up the spatial frequency corresponding to a correlation threshold (cut-off limit)

The spatial frequency at the correlation limit is the Fourier Image Resolution (FIRE). This can be interpreted as stating that there is no meaningful similarity information at any frequency higher than the resolution, i.e. if the images were reconstructed using only frequencies at this level and above then they would not be recognised as matching.

### 7.14.1 Input Data

Note that the FRC analysis requires two input images; these are constructed from localisations as super-resolution images. This means that the FRC can be computed using a single input dataset split into two halves to compute the resolution of the data, or using two input datasets to compute the resolution of one dataset against the other. This may be useful if two different super-resolution datasets have been obtained for the same structure for example by using a two-channel experiment with different fluorophores.

In the classic case of a single input dataset the localisations must be split into halves. The plugin provides a framework to do this where all localisations are sorted by time frame. Consecutive localisations are then allocated to a block of set size. The resulting blocks can then be split alternating or randomly between the two output images. Using blocks of localisations sorted by time should reduce the number of occurrences when the same molecule that appears in consecutive frames is split into different images. This artificially increases the resolution as the same molecule is present in both images.

Note that if non-random splitting is performed then the output results will be the same when repeated. If data is split randomly then the output resolution will vary since the super-resolution images are different between repeats. Repeating the analysis multiple times with different splits is used to produce an average resolution and a confidence interval.

When two datasets are used then no splitting is required.

### 7.14.2 Threshold methods

The final resolution is dependent on the threshold used to assess when the correlation of the image at a given spatial frequency is not significant. Changing the threshold method will change the calculated resolution. The FRC plugin can compute the correlation threshold using the following methods:

#### 7.14.2.1 Fixed

Use the fixed value  $1/7 = 0.142857$ . This is the threshold value preferred by Niewenhuizen, *et al* (2013) as it provides output that closely matches the apparent resolution of images under manual inspection.

#### 7.14.2.2 *n*-Bit information content (Fourier-space signal-to-noise-ratio)

This method sets the threshold using a level expected for a given information content (Signal-to-Noise Ratio, SNR) in each pixel of the super-resolution image.

The threshold is computed at each frequency ( $T_i$ ) for the predefined Fourier-space signal-to-noise ratio ( $SNR_T$ ) from the number of data points ( $N_i$ ) used to compute the FRC with the following formula:

$$T_i = \frac{SNR_T/2 + 2/\sqrt{(N_i)} \cdot \sqrt{(SNR_T/2)} + 1/\sqrt{(N_i)}}{SNR_T/2 + 2/\sqrt{(N_i)} \cdot \sqrt{r_i} + 1}$$

$$T_i = \frac{SNR_T/2 + (2\sqrt{(SNR_T/2)} + 1)/\sqrt{(N_i)}}{1 + SNR_T/2 + 2\sqrt{(SNR_T/2)}/\sqrt{(N_i)}}$$

(see Heel & Schatz, (2005), equation 13). Note that the desired  $SNR_T$  is divided by two since we need half the SNR each each of the half datasets (the two images). For a smoother curve the number of points ( $N_i$ ) is computed as  $2 \cdot \pi \cdot r_i$  and not the actual number of samples computed during the Fourier ring analysis.

The SNR threshold is set using the desired number of bits of information required per pixel using:

$$n - bits = \log_2(1 + SNR_T)$$

Note the use of the binary logarithm. Classically this can be done using  $n - bits = 1/2, 1$  or  $2$  bits. For example for a  $1/2$  bit curve  $SNR_T = 0.4142$  and the formula reduces to:

$$T_i = \frac{0.2071 + 1.9102/\sqrt{N_i}}{1.2701 + 0.9102/\sqrt{N_i}}$$

#### 7.14.2.3 Sigma

Compute the threshold at each frequency ( $T_i$ ) from the number of data points ( $N_i$ ) used to compute the FRC with the following formula:

$$T_i = \frac{\sigma}{\sqrt{N_i/2}}$$

with  $\sigma$  the sigma-factor (see Heel & Schatz, (2005), equation 2). For example sigma can be 1, 2, 3, etc.

### 7.14.3 Spurious correlation (Q) correction

When analysing a single dataset, if the same emitter is localised more than once then this can lead to correlations being observed at all spatial frequencies if localisations from the same emitter are split into each half-dataset (because a non-zero pixel will be present in the same position in both images). These spurious correlations can be corrected by subtracting a term from the numerator of the FRC and adding it to the denominator. The FRC at a given frequency  $q$  is given by the sum of the conjugate multiple of the Fourier transform of two sub-images divided by the product of the sums of the squared absolute values of each transformed sub-image:

$$FRC(q) = \frac{\sum_{\vec{q} \in C} \hat{f}_1(\vec{q}) * \hat{f}_2(\vec{q})}{\sqrt{\sum_{\vec{q} \in C} |\hat{f}_1(\vec{q})|^2} \sqrt{\sum_{\vec{q} \in C} |\hat{f}_2(\vec{q})|^2}}$$

The sum is computed over all pixels at the specified spatial frequency  $q = |\vec{q}|$  where  $C$  denotes the pixels of the perimeter of a circle of constant spatial frequency and  $q$  is valued as  $0/L, 1/L, 2/L, \dots$  with  $L$  the pixel size of the input images ( $1/L$  is the pixel size in Fourier space). The ring sum can be scaled to a ring mean by dividing by the number of samples in the circle ( $N_C$ ):

$$FRC(q) = \frac{\frac{1}{N_C} \sum_{\vec{q} \in C} \hat{f}_1(\vec{q}) * \hat{f}_2(\vec{q})}{\frac{1}{N_C} \sqrt{\sum_{\vec{q} \in C} |\hat{f}_1(\vec{q})|^2} \sqrt{\sum_{\vec{q} \in C} |\hat{f}_2(\vec{q})|^2}}$$

The correction term is computed as:

$$Corr(q) = \frac{Q}{Q_{norm}} \cdot H(q) \text{sinc}(\pi \cdot q)^2$$

where  $Q$  is the average number of activation cycles per emitter (i.e. repeat localisations),  $Q_{norm}$  is a normalisation using the mean pixel value in the input images prior to Fourier transform:

$$Q_{norm} = \frac{1}{mean_1} + \frac{1}{mean_2}$$

and  $H(q)$  is the factor in the correlation averages related to the localisation uncertainties which is modelled as a Gaussian distribution with mean  $\sigma_m$  and width  $\Delta\sigma$  (in units of super-resolution pixels):

$$H(q) = \frac{1}{\sqrt{1 + 8\pi^2 \Delta \sigma^2 q^2}} \exp\left(-\frac{4\pi^2 \sigma_m^2 q^2}{1 + 8\pi^2 \Delta \sigma^2 q^2}\right)$$

Note that the given equations have been updated from those provided in Niewenhuizen, *et al* (2013) to those used in the code. If  $Q=0$  then no correction is applied. A plugin for estimating  $Q$  (the average number of activation cycles per emitter) is provided (see 7.15 FIRE Q Estimation).

The corrected FRC is thus:

$$FRC(q) = \frac{\frac{1}{N_C} \sum_{\vec{q} \in C} \hat{f}_1(\vec{q}) * \hat{f}_2(\vec{q}) - Corr(q)}{\frac{1}{N_C} \sqrt{\sum_{\vec{q} \in C} |\hat{f}_1(\vec{q})|^2} \sqrt{\sum_{\vec{q} \in C} |\hat{f}_2(\vec{q})|^2} + Corr(q)}$$

#### 7.14.4 Data Selection

The FRC plugin provides the following options to select the localisation data:

| Parameters         | Description  |
|--------------------|--|
| INPUT              | The results set to analyse   |
| INPUT2             | The optional second results set to analyse   |
| USE ROI/CHOOSE ROI | <p>If any open images current have an area ROI then the option to choose an ROI is provided. This option is not shown when no ROIs are available</p> <p>Use ROI is shown when only 1 ROI is available.</p> <p>Choose ROI is shown when multiple ROIs are available. A second dialog will be shown to select the image with the desired ROI</p> |

If an ROI is selected then the ROI will be scaled from the source image to the bounds of the input data and only localisations within the bounds selected. An efficient way to produce an appropriate ROI is to construct a super resolution image of the input dataset using the RESULTS MANAGER plugin. An ROI can be drawn on the super resolution image and only that region will be used for the FIRE computation.

#### 7.14.5 Parameters

The FRC plugin provides options for each stage of the process: Image construction; Fourier transform; FIRE analysis; and additional options for single dataset processing:

##### 7.14.5.1 Image construction

The full 2D bounds of the data are used to define an image scale so the output is a suitable width/height for Fourier transform. If the output is very large then Fourier transform will be slow and if too large will fail due to lack of memory, so options are provided to scale the image automatically to a suitable size for Fourier transform. Localisations are rendered onto the image using bilinear weighting onto the 4 neighbour pixels surrounding the coordinates to reduce pixelation artefacts in the super-resolution image.

| Parameters  | Description  |
|-------------|--|
| IMAGE SCALE | Specify the image enlargement scale to create the super-resolution image from the localisation data. Auto will create an image that fits the data into the Auto image size |

| Parameters      | Description   |
|-----------------|---|
| AUTO IMAGE SIZE | The size for images created with <code>AUTO</code> scaling.<br><br>Note that <code>AUTO</code> is the preferred method since the images must be a power of 2 on the long edge for efficient Fourier transform and an optimal image will use all of the available pixels |
| MAX PER BIN     | Set above zero to limit the maximum number of localisations that are drawn in each super-resolution pixel.<br><br>Note: This is set to 5 in Nieuwenhuizen, <i>et al</i> (2013) to limit the effect of bright spots from anomalous fluorescent molecules                 |

#### 7.14.5.2 Fourier options

| Parameters      | Description  |
|-----------------|--|
| FOURIER METHOD  | Set the method to compute the Fourier transform. Using the <code>JTRANSFORMS</code> library has marginal performance improvements. The default Fast Hartley Transform (FHT) will produce identical results   |
| SAMPLING METHOD | Choose the method to sample pixels on the Fourier ring.<br><br><code>RADIAL SUM</code> bins all the pixels between radius <code>n</code> to <code>n+1</code> (exclusive) into ring <code>n</code> .<br><br><code>INTERPOLATED CIRCLE</code> draws a circle on the Fourier image and samples the perimeter at intervals by interpolating the pixels surrounding the perimeter coordinates |
| SAMPLING FACTOR | The FRC is calculated by sampling the circle at <code>N</code> points where <code>N</code> is equal to $\pi r \times \text{SAMPLING FACTOR}$ . Increasing this value will slow down the algorithm but increase resolution of the FRC curve.<br><br>Note that samples are taken in the interval $0-\pi$ radians since the Fourier image is two-fold radially symmetric                    |

#### 7.14.5.3 FIRE options

| Parameters       | Description  |
|------------------|--|
| THRESHOLD METHOD | The method for determining the correlation cut-off (see 7.14.2 Threshold methods)                  |
| SHOW FRC CURVE   | Display the FRC curve. The curve shows the raw data, the smoothed data and the correlation cut-off |

#### 7.14.5.4 Single dataset processing

A single data must be split into two halves to construct the images for analysis. If this split is random then multiple repeats can be performed to provide a mean resolution with a



confidence interval. Additionally spurious correlations can be corrected using a Q correction factor (see 7.14.3 Spurious correlation (Q) correction).

| Parameters                      | Description  |
|---------------------------------|--|
| BLOCK SIZE                      | The number of sequential localisations to assign to a block.<br>Using blocks reduces correlation artefacts from the same molecule occurring in consecutive frames being split into the two images  |
| RANDOM SPLIT                    | If true then the block data is split randomly into two halves. Otherwise the data is split by alternating blocks   |
| REPEATS                         | If using a RANDOM SPLIT then repeating the analysis will produce in a different result. Use this option to perform multiple repeats to provide a mean resolution with a confidence interval.<br>Repeats use multi-threaded processing              |
| SHOW FRC CURVE REPEATS          | When using repeats only the smoothed curve is added to a combined FRC curve output.<br>Select this option to draw an FRC curve plot with raw and smoothed data for each repeat   |
| SHOW FRC TIME EVOLUTION         | Sort the results by time and compute the FIRE number using 10 step increments up to the max time. This shows how the resolution has changed at the time increases.<br>This option is computationally intensive and is not currently multi-threaded |
| SPURIOUS CORRELATION CORRECTION | Select this to perform spurious correlation correction   |
| Q-VALUE                         | The Q correction value. This is the average number of times a molecule is repeat localised. Zero means each molecule is localised only once and no correction is performed   |
| PRECISION MEAN                  | The mean of the Gaussian distribution of the localisation precision  |
| PRECISION SIGMA                 | The standard deviation (width) of the Gaussian distribution of the localisation precision  |

### 7.14.6 Output

The plugin will display progress of the computation on the ImageJ status bar. When the plugin has finished the FIRE number will be reported to the ImageJ log window:

Fit Spot Data : FIRE number = 31.98 nm (Fourier scale = 33.0)

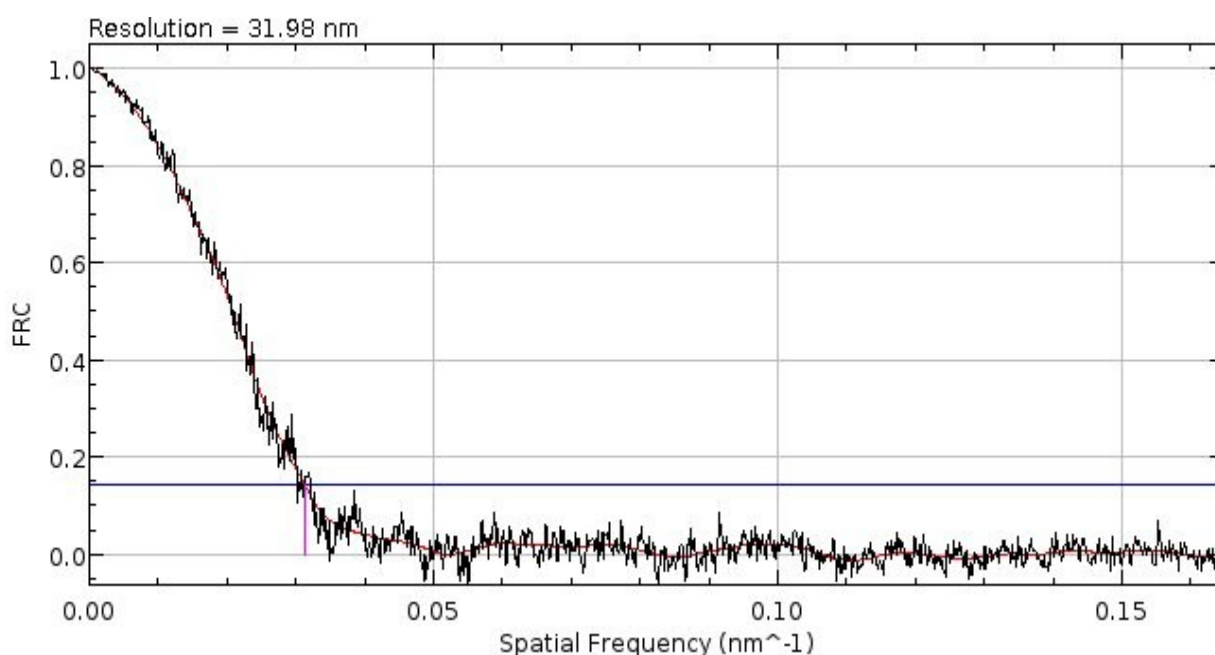


If repeats were performed then the mean resolution is reported with a confidence interval:

Fit Spot Data : FIRE number = 32.95 +/- 0.4573 nm [95% CI, n=10]  
(Fourier scale = 33.0)

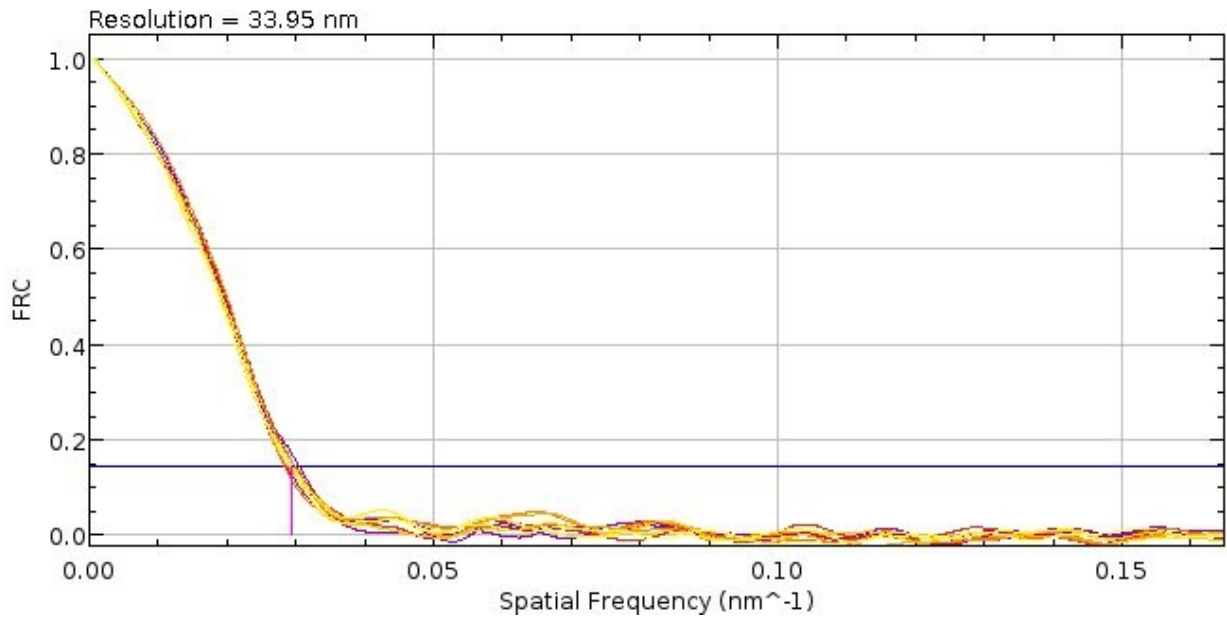
### 7.14.6.1 FRC Curve

The FRC curve data can be very noisy. The data is smoothed using a LOESS smoothing procedure and the crossing points with the correlation cut-off curve are computed. Illustration 28 shows an example of an FRC curve. The smoothed FRC line crosses the cut-off curve at a frequency of approximately  $0.031 \text{ nm}^{-1}$ . The resolution is therefore approximately 32nm.



*Illustration 28: Example FRC curve from the Fourier Image Resolution plugin. The raw FRC data are in black. Smoothed FRC data are in red. The resolution cut-off threshold is in blue at a fixed value of 1/7.*

If multiple repeats are performed with randomly split data then a combined curve is produced with all the smoothed curves and the mean resolution (see Illustration 29).



*Illustration 29: Example combined FRC Curve from 10 repeats of Fourier Image Resolution analysis*

## 7.15 FIRE Q Estimation

Estimate the spurious correction factor  $Q$  for use in Fourier Image Resolution (FIRE) analysis.  $Q$  is the average number of activation cycles per emitter (i.e. repeat localisations).

This plugin is based on the Matlab code kindly provided by Prof. Bernd Reiger (corresponding author of the FIRE paper (Niewenhuizen, *et al* (2013))). The method described here is an update to the original method described in the Online Methods of the FIRE paper.

Fourier Ring Correlation (FRC) computes the correlation between the features of two images at a certain spatial frequency. This correlation is high at low spatial frequencies and progressively reduces. When the correlation falls below a set threshold there is no meaningful similarity information at any further frequency. The spatial frequency at the correlation limit is the Fourier Image Resolution (FIRE).

When analysing a single dataset, if the same emitter is localised more than once then this can lead to correlations being observed at all spatial frequencies if the localisations from the same emitter are split into each half-dataset. These spurious correlations can be corrected by subtracting a term from the numerator of the FRC and adding it to the denominator. The correction term is computed using a factor  $Q$  which is the average number of activation cycles per emitter (i.e. repeat localisations), and by assuming the localisations have an uncertainty (precision) that is distributed as a Gaussian with mean  $\sigma_m$  and width  $\Delta\sigma$ . This allows the correction to the correlation to be modelled as:

$$Corr(q) = \frac{Q}{Q_{norm}} \cdot H(q) \text{sinc}(\pi \cdot q)^2$$

where  $Q_{norm}$  is a normalisation using the mean pixel value in the input images prior to

Fourier transform:

$$Q_{norm} = \frac{1}{mean_1} + \frac{1}{mean_2}$$

and  $H(q)$  is the factor in the correlation averages related to the localisation uncertainties:

$$H(q) = \frac{1}{\sqrt{1+8\pi^2\Delta\sigma^2q^2}} \exp\left(-\frac{4\pi^2\sigma_m^2q^2}{1+8\pi^2\Delta\sigma^2q^2}\right)$$

Note that  $H(q)$  is typically computed assuming the theoretical distribution of localisation uncertainties. However it can be constructed using a sample of  $N$  points from the actual localisation uncertainties if they are available:

$$H(q) = \frac{1}{N} \sum_{i=0}^{i=N} \exp(-4\pi^2\sigma_i^2q^2)$$

### 7.15.1 Estimating Q

The true correlation term (due to underlying structure in the data) will decline towards zero at high spatial frequencies. This leaves the spurious correlation term as the main contributor to the FRC numerator at high spatial frequencies. A method for estimating the term involves fitting the model of the correction term to the FRC numerator using a range of spatial frequencies where there is no actual correlation.

There are three terms to estimate:  $Q$ ,  $\sigma_m$  and  $\Delta\sigma$ . If the uncertainty is not available in the dataset then it must be entered as fixed value and only  $Q$  will be estimated. If the localisation uncertainty is available for each localisation then it can be estimated using two methods. The first method fits a Gaussian to a histogram of the localisation uncertainty. Alternatively the  $H(q)$  term can be initially constructed using a random sample of the actual localisation uncertainty and the mean estimated from the initial quadratic fit (see below).

The FRC numerator at a given frequency  $q$  is given by the sum of the conjugate multiple of the Fourier transform of two sub-images:

$$FRC\ numerator(q) = \sum_{\vec{q} \in C} \hat{f}_1(\vec{q}) * \hat{f}_2(\vec{q})$$

The sum is computed over all pixels at the specified spatial frequency  $q = |\vec{q}|$  where  $C$  denotes the pixels of the perimeter of a circle of constant spatial frequency and  $q$  is valued as  $0/L$ ,  $1/L$ ,  $2/L$ , ... with  $L$  the pixel size of the input images ( $1/L$  is the pixel size in Fourier space). This term is divided by the number of pixels in the Fourier circle to produce function  $v(q)$ :

$$v(q) = \frac{1}{2\pi qL} \sum_{\vec{q} \in C} \hat{f}_1(\vec{q}) * \hat{f}_2(\vec{q})$$

Note that the term  $2\pi qL$  is the circumference of the Fourier circle and is adjusted during computation to be the actual number of samples taken from the Fourier circle.

If the correlation is mainly composed of the spurious correlation then the following relation holds:

$$1 = \frac{v(q)}{\frac{Q}{Q_{norm}} \cdot H(q) \text{sinc}(\pi \cdot q)^2}$$

Q can thus be estimated using:

$$Q = \frac{Q_{norm} \cdot v(q)}{H(q) \text{sinc}(\pi \cdot q)^2}$$

This term varies widely in magnitude with q and so the following function is used:

$$f(q) = \log(|Q_{norm} v(q) / H(q) / \text{sinc}(\pi q)^2|)$$

This function is described below equation (7) of the Online Methods of Niewenhuizen, *et al* (2013). The function  $f(q)$  typically is a function that initially decreases, then levels off to a constant plateau value and finally increases again. The function can be well fit using a quadratic (  $f(q) = a + bq^2$  ) if the fit is performed over a suitable range of  $q$  where the plateau occurs. To reduce noise the plugin allows smoothing using a median window before application of the absolute and logarithm functions or it can be smoothed afterwards using LOESS smoothing. The Y-intercept is used as an initial estimate of Q:

$$Q = \exp(a)$$

Note that  $H(q)$  is typically computed assuming the theoretical distribution of localisation uncertainties. However it can be constructed using a random sample of  $N$  points from the actual localisation uncertainties. In this case the parameter  $\sigma_m$  can also be estimated from the fit using:

$$\sigma_m = \sqrt{\left(\frac{1}{N} \sum \sigma_i^2 - b/4\pi^2\right)}$$

and the other parameter  $\Delta\sigma$  set using the standard deviation of the localisation uncertainty values.

The initial estimate used a quadratic fit of the function  $f(q)$ . This estimate is then refined by optimising the parameters so that  $v(q)$  most closely matches  $Corr(q)$ . This is done by optimising the following cost function for all  $q$ :

$$\begin{aligned} \text{cost}(q) &= 1 - \exp(-\text{diff}^2 / \text{noise}^2); \\ \text{diff} &= \frac{v(q)}{Corr(q)} - 1 \end{aligned}$$

with  $\text{noise}$  set to a defined level to scale the difference value appropriately. The cost function has a value between 0 and 1 for all observations; a low value is better. Note that  $Corr(q)$  is constructed using Q,  $\sigma_m$  and  $\Delta\sigma$  and this second optimisation can be done by fitting Q alone and fixing  $\sigma_m$  and  $\Delta\sigma$ , or fitting the 3 parameters together.

## 7.15.2 Parameters

The FRC plugin provides options for each stage of the process: Data selection; Image construction; Fourier transform; and Estimation options:

### 7.15.2.1 Data Selection

These are the same as the FIRE plugin (see 7.14.4: Data Selection) without the option for a second dataset. A subset of the data can be selected using an ROI selected from an open image.

**7.15.2.2 Image construction**

These are the same as the FIRE plugin for single datasets. Note that localisations are drawn using bilinear weighting onto the 4 neighbour pixels surrounding the coordinates to reduce pixelation artefacts in the super-resolution image.

| Parameters      | Description   |
|-----------------|---|
| IMAGE SCALE     | Specify the image enlargement scale to create the super-resolution image from the localisation data. Auto will create an image that fits the data into the Auto image size  |
| AUTO IMAGE SIZE | The size for images created with AUTO scaling.<br>Note that AUTO is the preferred method since the images must be a power of 2 on the long edge for efficient Fourier transform   |
| BLOCK SIZE      | The number of sequential localisations to assign to a block.<br>Using blocks reduces correlation artefacts from the same molecule occurring in consecutive frames being split into the two images   |
| RANDOM SPLIT    | If true then the block data is split randomly into two halves. Otherwise the data is split by alternating blocks  |
| MAX PER BIN     | Set above zero to limit the maximum number of localisations that are drawn in each super-resolution pixel.<br>Note: This is set to 5 in Nieuwenhuizen, <i>et al</i> (2013) to limit the effect of bright spots from anomalous fluorescent molecules |

**7.15.2.3 Fourier options**

These are the same as the FIRE plugin (see 7.14.5.2: Fourier options).

**7.15.2.4 Estimation options**

| Parameters       | Description   |
|------------------|---|
| THRESHOLD METHOD | The method for determining the correlation cut-off (see 7.14.2 Threshold methods) |

| Parameters       | Description   |
|------------------|---|
| PRECISION METHOD | <p>Specify the method to use to determine the parameters for the distribution of the localisation precision (assumed to be Gaussian).</p> <p>FIXED: Use a fixed value for the precision distribution mean and standard deviation.</p> <p>STORED: Use the precision value that is stored in the results, e.g. from loaded data. These are fit with a Gaussian to estimate <math>\sigma_m</math> and <math>\Delta\sigma</math>.</p> <p>CALCULATE: Calculate the precision of each localisation using the formula of Mortensen. These are fit with a Gaussian to estimate <math>\sigma_m</math> and <math>\Delta\sigma</math>. This option requires that the localisations have a signal and width value and the results are calibrated with a gain and pixel size (in nm)</p> |
| PRECISION MEAN   | The mean of the localisation uncertainty (precision) for use in FIXED estimation ( $\sigma_m$ )   |
| PRECISION SIGMA  | The standard deviation of the localisation uncertainty (precision) for use in FIXED estimation ( $\Delta\sigma$ )   |
| SAMPLE DECAY     | <p>Set this to true to sample the localisation precision values to generate <math>H(q)</math> for the initial Q-estimation. <math>\sigma_m</math> is then also estimated from the initial fit.</p> <p>The default generates <math>H(q)</math> using <math>\sigma_m</math> and <math>\Delta\sigma</math> from the FIXED input or generated from fitting the precision histogram with a Gaussian</p>  |
| LOESS SMOOTHING  | Smooth the curve $f(q)$ using LOESS smoothing. The default smooths the data using a median window before applying the absolute and logarithm functions  |
| FIT PRECISION    | Set to true to optimise $\sigma_m$ and $\Delta\sigma$ when fitting $v(q)$ to $Corr(q)$ . The default only optimises the Q-value   |
| MINQ             | The minimum spatial frequency $q$ value to use for fitting*   |
| MAXQ             | The maximum spatial frequency $q$ value to use for fitting*   |

\* Note:  $q$  is specified in terms of the Fourier image width so has a range from 0 to 0.5. Choose a range for  $q$  where the correlation has fallen off from the initial high value to approach 0, i.e. the range where the correlation is mainly due to spurious correlations between repeats of the same molecule.

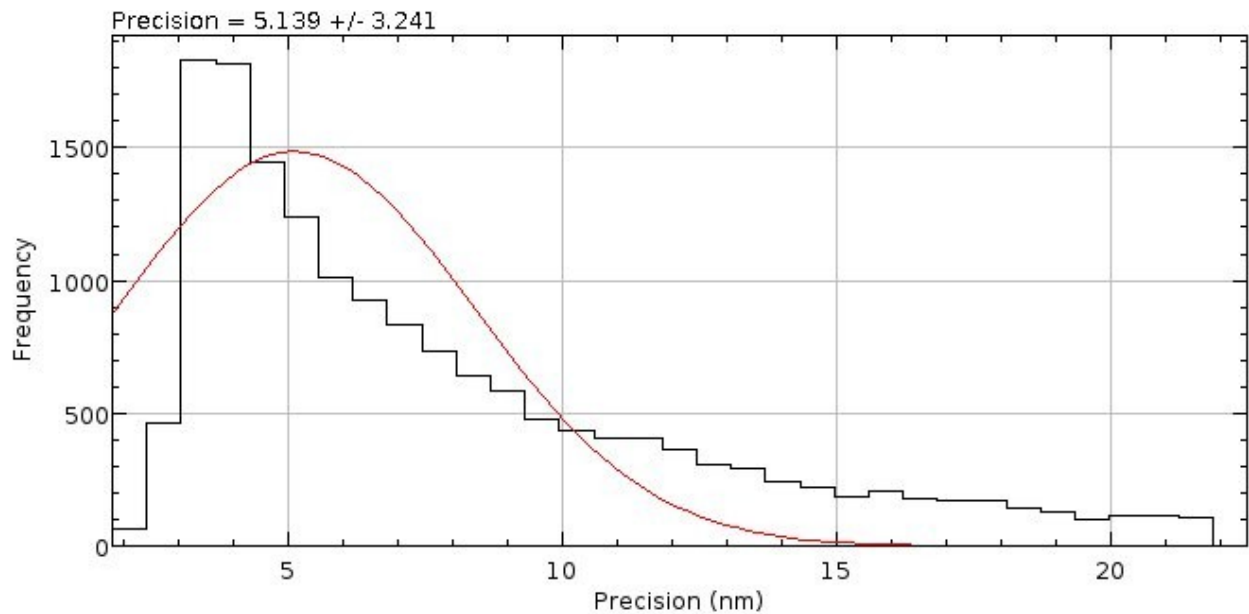
### 7.15.3 Output

The plugin will display progress of the computation on the ImageJ status bar. When the plugin has finished the results are presented graphically. A non-blocking dialog is used to

allow the user to adjust the estimated values and the displayed results are dynamically updated.

### 7.15.3.1 Precision Histogram

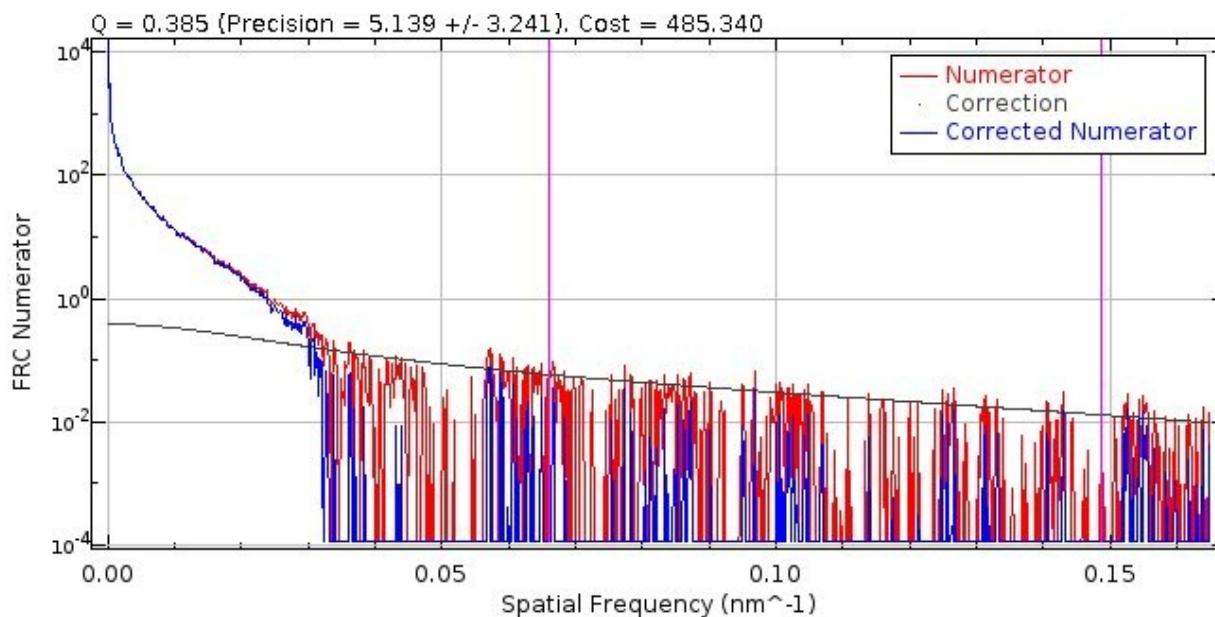
The histogram of the localisation uncertainty (precision) is displayed with an overlay of the current mean and standard deviation. If no precision is available then only the overlay is shown.



### 7.15.3.2 FRC Numerator Curve

The FRC numerator function,  $v(q)$ , is plotted against the correction,  $Corr(q)$ . The corrected numerator is also plotted to show the effect of correction.

Note that the purpose of the plugin is to maximise the fit between the numerator and the correction within the selected fitting range of  $q$ . The range is shown using magenta vertical lines. Ideally the range should exclude the low frequencies where the numerator is significantly different from the correction due to actual correlation within the dataset.



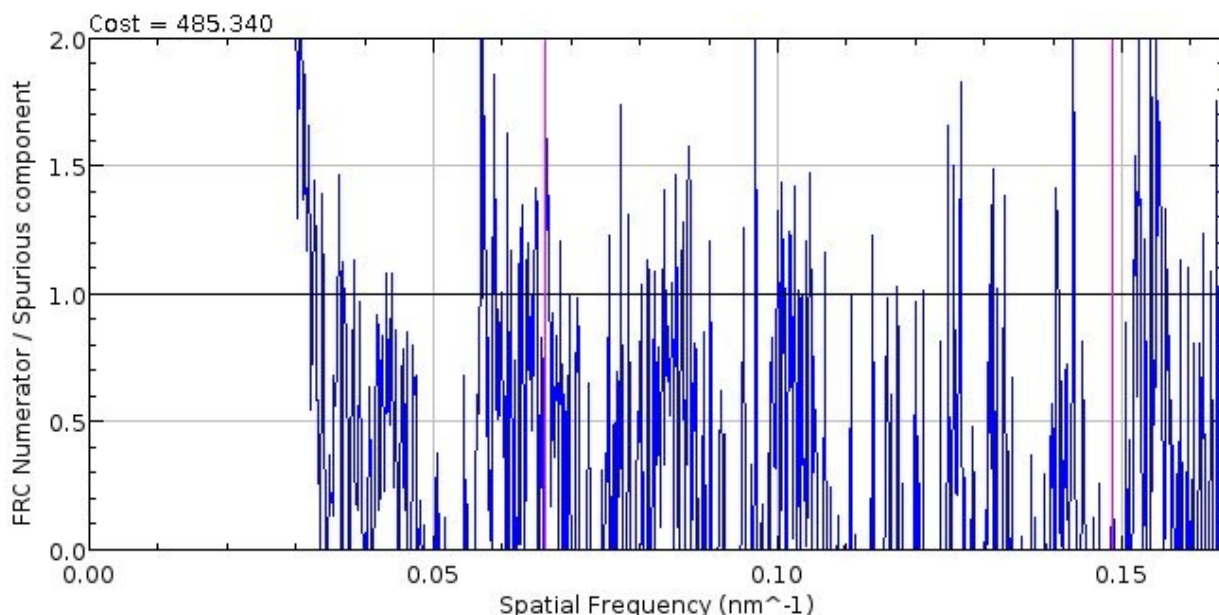
### 7.15.3.3 FRC Numerator/Correction Ratio

The cost function aims to minimise:

$$\sum g\left(\frac{v(q_i)}{Corr(q_i)} - 1\right)$$

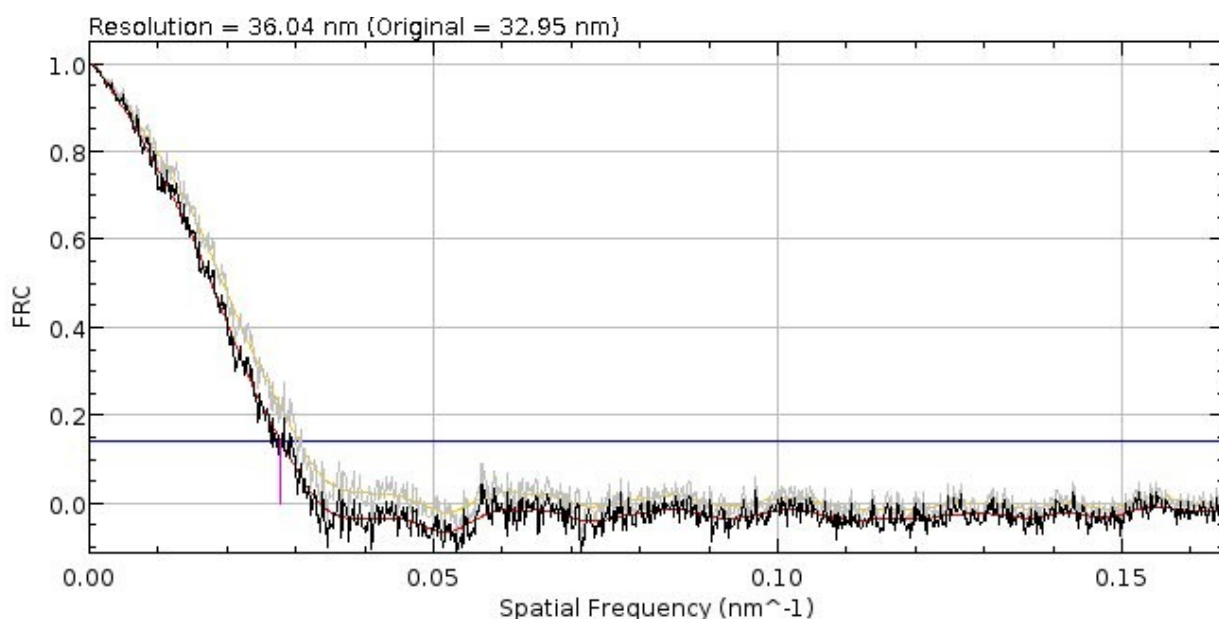


over a selected range of  $q$ , where  $g(x)$  is a mapping function with value 0 when  $x$  is zero and a maximum value of 1. Thus the ideal ratio  $v(q)/\text{Corr}(q)$  is 1. The Numerator/Correction Ratio plot shows  $v(q)/\text{Corr}(q)$ . The range is set to 0-2 and a line drawn at 1 for the ideal value. The fit range for  $q$  is shown using magenta vertical lines. This plot is often very noisy indicating that optimising the fit is non-trivial and sub-optimal results may occur (i.e. estimation has failed).



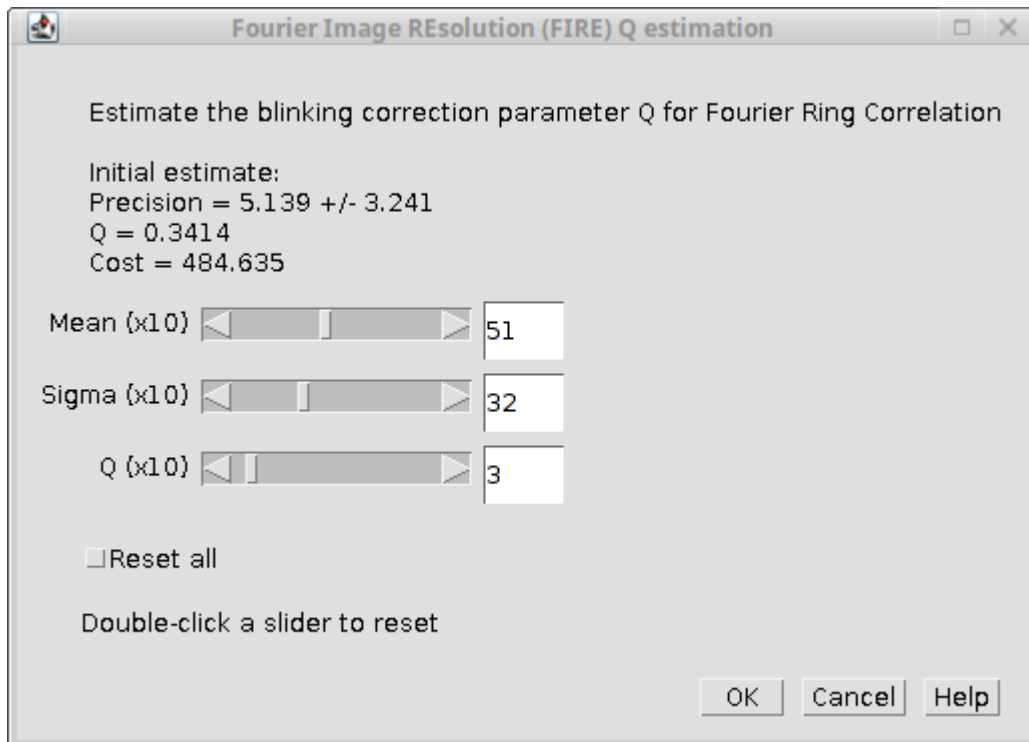
#### 7.15.3.4 FRC Curve

The FRC curve is similar to that shown by the FIRE plugin (see section 7.14.6.1). The raw data of the correlation is shown in black and the smoothed line in red. The resolution is computed using the crossing point with the threshold curve (blue). If the Q-value is above zero then the original data is shown with 50% opacity. The resolution before and after Q correction are shown in the label.



### 7.15.3.5 Interactive Q-Estimation

Once the plots have been displayed the plugin shows a dialog containing the initial estimates for  $Q$ ,  $\sigma_m$  and  $\Delta\sigma$  and the value of the cost function. This is a non-blocking dialog and so allows the user to move other ImageJ windows to optimise the layout.



The user can adjust any of the current parameter values and see the effects on the plots dynamically. The cost function value is updated and shown so that the effects of manual estimation can be compared to the initial auto-estimation. Adjustments can be made to the slider or by editing the text field value. Note that the estimate values are magnified 10-fold to allow finer adjustment of the estimates. This is due to the limited functionality of sliders within ImageJ dialogs.

The **RESET** checkbox can be used to set the values back to the initial estimates, for example if adjustments have significantly changed the cost function. Alternatively a double-click on a single slider will reset only that slider.

Clicking the **OK** button will close the dialog and the current estimates are saved to memory. These values will be used in the FIRE plugin for the fields for spurious correlation correction.

Clicking the **CANCEL** button will close the dialog without saving the current estimates.

## 8 PC PALM Plugins

The following plugins are used to analyse the auto-correlation of a set of localisations using Pair Correlation (PC) analysis. This can provide information on whether the localisations are randomly distributed or clustered ([Sengupta *et al*, 2011], [Sengupta, *et al*, 2013], [Puchnar, *et al*, 2013]).

The analysis uses a set of localisations assumed to represent single molecules or single fluorescent bursts. The data is compared to itself and a curve is computed as a function of the distance from the centre of the localisation. A flat curve is indicative that the distribution of localisations is no different from a random distribution. Shaped curves can be fit using models that apply to various distributions of localisations (random; fluctuations; or an emulsion).

The plugins are described in the following sections using the order presented on the PLUGINS > GDSC SMLM > PC PALM menu.

### Note:

**The PC-PALM plugins are under development and should be considered experimental. The plugins are subject to change and so no documentation has yet been produced to describe them. They are included in this package to assess interest from the community. If you wish to use the plugins please contact [a.herbert@sussex.ac.uk](mailto:a.herbert@sussex.ac.uk) for more information.**

### 8.1 PC-PALM Molecules

Prepare results held in memory for analysis using pair correlation methods. This plugin either simulates results or filter results from a results set to a set of coordinates with time and photon signal information. The localisations are drawn on a binary image to allow regions of the data to be selected for analysis.

### 8.2 PC-PALM Analysis

Perform pair-correlation analysis in the frequency domain as per the paper by [Sengupta *et al*, 2011], [Sengupta, *et al*, 2013] to produce a  $g(r)$  correlation curve.

### 8.3 PC-PALM Spatial Analysis

Perform pair-correlation spatial analysis as per the paper by [Puchnar, *et al*, 2013]. This methods plots the molecule density around each localisation as a function of distance from the localisation.

### 8.4 PC-PALM Save Results

Saves all the PC-PALM results held in memory to a results folder.

### 8.5 PC-PALM Load Results

Load all the PC-PALM results from a results folder to memory.

## **8.6 PC-PALM Fitting**

Combines multiple correlation curves calculated by PC-PALM Analysis into an average curve and fits the curve using various models.

## **8.7 PC-PALM Clustering**

Clusters localisations using a distance threshold and produces a histogram of cluster size. This can be fit using a zero-truncated negative Binomial distribution (with parameters  $n, p$ ) to calculate the size of the clusters ( $n$ ) and the probability of seeing a fluorophore ( $p$ ).

## 9 Model Plugins

The following plugins allow single-molecule images to be simulated. This can be used for benchmarking fitting parameters by comparing the results of fitting the simulation to the actual known positions. Note that comparison of two sets of localisations can be done using the **RESULTS MATCH CALCULATOR** plugin (see section 6.12).

### 9.1.1 Benchmarking

The benchmarking system provides the means to optimise the SMLM tools to produce the best results on data that is typical of a microscope setup.

- Create an ideal PSF of the microscope (**PSF CREATOR** and **PSF DRIFT**)
- Draw spots on an image using a defined camera noise model at configured XY positions and z-depths (**CREATE SPOT DATA**)
- Identify candidates for fitting (**FILTER SPOT DATA**)
- Fit candidate spots (**FIT SPOT DATA**)
- Filter the fitting results, i.e accept or reject fitting results (**BENCHMARK FILTER ANALYSIS**)

The benchmarking workflow will provide statistics on the recall and precision that can be achieved when fitting localisations of a given photon signal strength throughout the depth-of-field of the point spread function.

The plugins are described in the following sections using the order presented on the **PLUGINS > GDSC SMLM > MODEL** menu.

### 9.2 PSF Creator

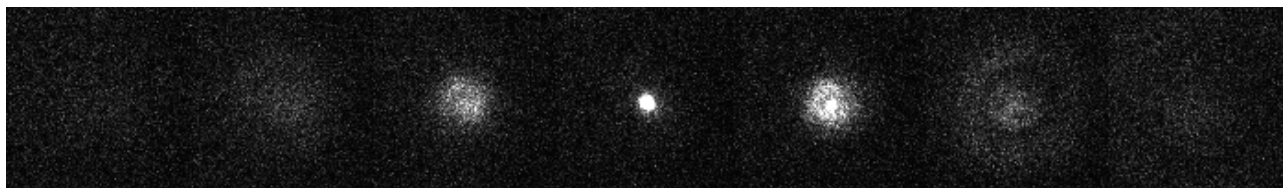
Produces an average PSF image using selected diffraction limited spots from a sample image.

The **PSF CREATOR** plugin can be used to create a Point Spread Function (PSF) image for a microscope. The PSF represents how a single point source of light passes through the microscope optics to be captured on the camera. Due to physical limits the wavelength of light cannot be focussed perfectly and will appear as a blurred spot. The spot will change size with z-depth as the light will be captured as it is converging to or diverging from the focal point. Additionally since the spot is actually composed of a series of waves it may appear as a ring due to diffraction. More details on the PSF can be found in section 2.1: Diffraction limit of light microscopy.

The exact shape of a PSF can be calculated using various models that account for the diffraction of various immersion media (water, oil, etc.) used to image samples. However individual microscope optics are unique and the PSF may vary from one set-up to another even if the hardware is duplicated. The **PSF CREATOR** allows an image model of the PSF to be created that can be used in simulations to draw diffraction limited spots that appear the same as those taken on the microscope. These simulations can be used to optimise localisation analysis.

### 9.2.1 Input image

The input image must be a z-stack of diffraction limited spots, for example quantum dots or fluorescent beads. The spot must be imaged through a large z range in small increments from out-of-focus through focus to out-of-focus. This will allow the entire PSF to be captured. The first and last frames are used to set a background level for the image intensity so ideally the spot should not be visible at all. An example spot imaged at 1000nm intervals is shown in Illustration 30. It can be seen that the spot disappears when 3 $\mu$ m out of focus. Ideal input images should cover a similar range but using a smaller step size, for example 20nm.



*Illustration 30: Fluorescent bead imaged at 1000nm intervals. The central frame is in focus. Contrast levels have been set to show the PSF when out-of-focus.*

When preparing a calibration image not all the spots are ideal due to problems with sample preparation. The spots should be inspected and only those that show a small in-focus spot and a smooth transition to out-of-focus should be selected for analysis. The spots may have their focal point in different z slices. In addition there should be no surrounding spots that will contribute overlapping PSFs to the image.

The spots should be marked using the IMAGEJ Point ROI tool. Right-clicking on the toolbar button will allow the tool to be changed to multiple-point mode. Clicking the image will add a point. Points can be dragged using the mouse and a point can be removed by holding the ALT key down while clicking the point marker.

### 9.2.2 Analysis

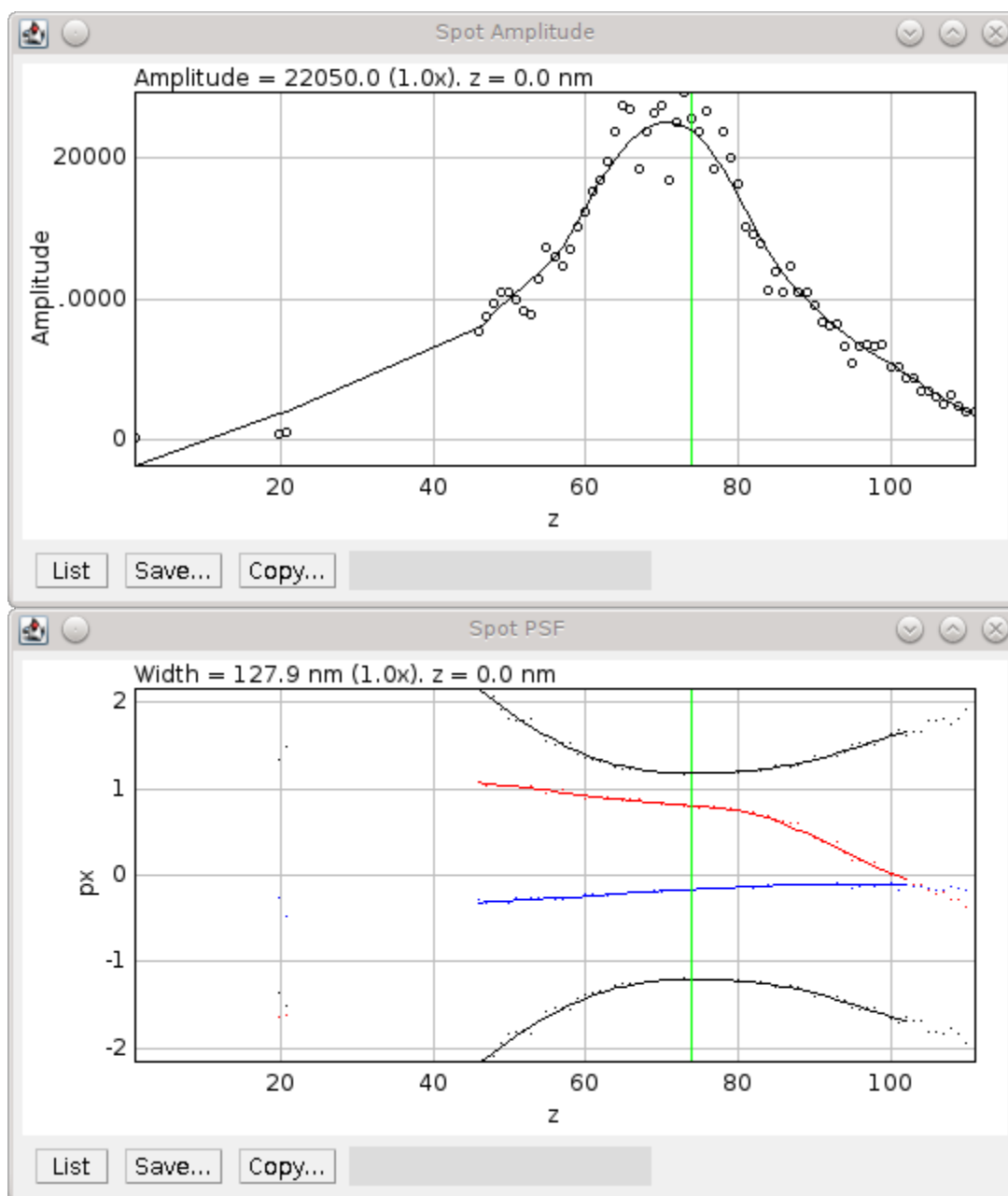
Each marked spot will be analysed in turn. Spots will only be used when there are no other spots within a specified distance to ensure a clean signal is extracted, i.e. no overlapping PSFs.

For each frame the plugin will run the P<sub>EAK</sub> F<sub>IT</sub> algorithm to fit the amplitude, centre and width of the peak. Fitting will begin to fail when the peak is very out-of-focus as the PSF may not resemble a 2D Gaussian.

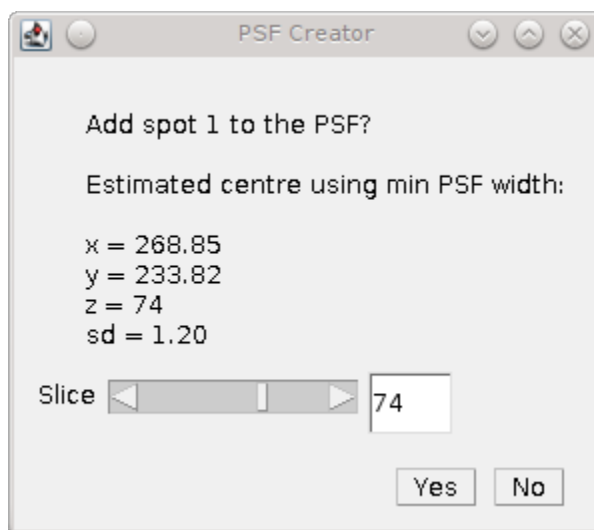
The amplitude is smoothed using a LOESS smoothing algorithm and plotted against the z-position. The amplitude should be highest when the peak is in focus. This point from the smoothed data is taken as the initial centre slice. The range of the in focus spot is marked by moving in either direction from the center slice until the smoothed amplitude is below a set fraction of the highest point.

The width and centre X and Y positions are then extracted for the in-focus range and smoothed using the LOESS algorithm. Since the amplitude is not a very consistent marker the centre slice is moved to the point with the lowest width. The spot centre is then recorded for the centre slice using the smoothed centre X and Y data.

The identification of the spot centre can be run automatically using configured parameters. Alternatively the plugin can run in interactive mode. In this instance the plugin will produce plots of the raw and smoothed data as shown in Illustration 31. The calculated centre is shown as a green line and the user is asked if the analysis result should be accepted or rejected (see Illustration 32). The user is able to adjust the centre of the spot using a slider if the plugin appears to have miscalculated.



*Illustration 31: Amplitude and PSF plots generated by the PSF Creator plugin. Amplitude plot shows raw data (circles) and smoothed data (black line). PSF plot shows raw data as spots and smoothed data from the in-focus region as a line. Width (black), X centre (blue) and Y centre (red). The centre z-slice is marked with a green line.*

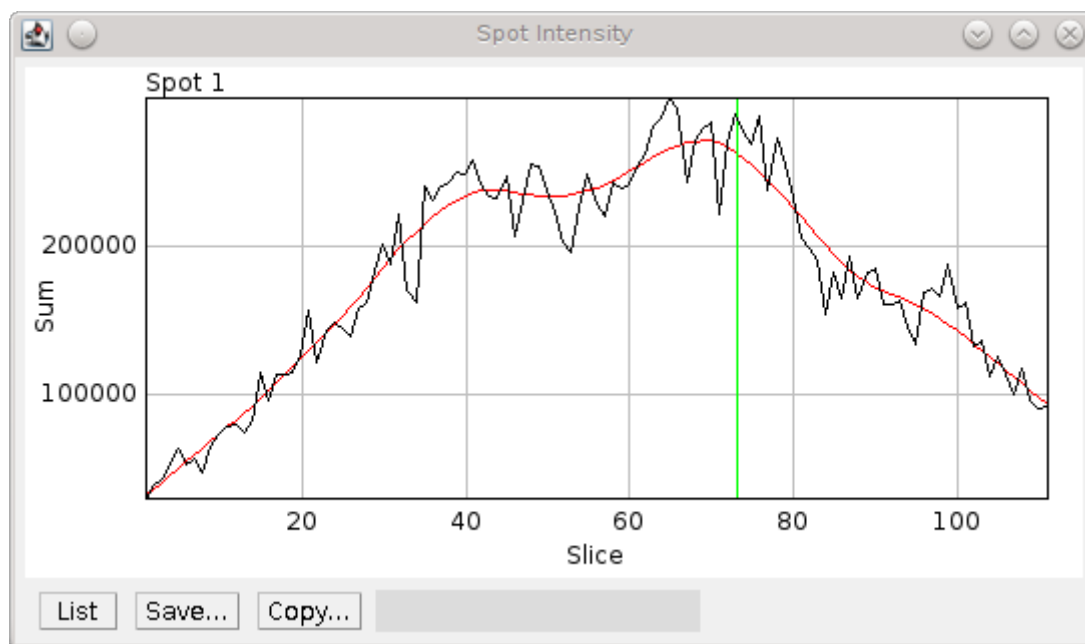


*Illustration 32: PSF Creator Yes/No dialogue show in interactive mode.*

When all the spot centres have been identified the plugin will generate a combined PSF image. Each spot is extracted into a stack and enlarged using the configured settings. The background is calculated for the spot using the N initial and M final frames and subtracted from the image. A Tukey window is then applied to the spot so that the edge pixels approach zero.

If using interactive mode the user has a second chance to view the spot data and accept it. A plot is produced of the total intensity within half of the region surrounding the spot against the z-position (see Illustration 33). At this stage the centre cannot be adjusted. However it is possible to view when the profile does not smoothly fall away in intensity from the centre, i.e. the spot is gradually defocussed from the centre, and reject the spot.





*Illustration 33: Spot intensity within half the region surrounding the spot. The profiles is produced after the image has been scaled, background normalised and windowed. The centre is marked using a green line.*

For all spots that are accepted, the spots are then overlaid using their X, Y and Z centres into an average PSF image. It is assumed that the in-focus spot can be modelled by a 2D Gaussian. All the pixels with 3 standard deviations of the centre are summed as foreground pixels. The image is then normalised across all frames so that the sum of the foreground is 1.

### 9.2.3 Parameters

| Parameters               | Description   |
|--------------------------|---|
| UPDATE FIT CONFIGURATION | Run the FIT CONFIGURATION plugin to allow the fitting parameters to be updated.<br><br>The WIDTH FACTOR should be set to allow very wide (out-of-focus) spots (e.g. 5) and the SIGNAL STRENGTH should allow poor spots (e.g. 30). |
| NM PER SLICE             | The z-slice step size used when acquiring the calibration image   |
| RADIUS                   | The square radius around each marked point to use for analysis. Any spot within 2 x radius will eliminate the spot from analysis  |
| AMPLITUDE FRACTION       | The fraction of the peak amplitude to use to mark the in-focus spot   |
| START BACKGROUND FRAMES  | The number of initial frames to use to calculate the background   |

| Parameters            | Description   |
|-----------------------|---|
| END BACKGROUND FRAMES | The number of final frames to use to calculate the background   |
| MAGNIFICATION         | The magnification to use when enlarging the final PSF image   |
| SMOOTHING             | The LOESS smoothing parameter   |
| CENTRE EACH SLICE     | Set the centre of each slice the to the centre-of-mass.<br>Note that using this option may cause consecutive frames to shift erratically. A better approach is to disable this and compute a drift curve using the PSF DRIFT plugin |
| CoM CUT OFF           | The amplitude cut-off for pixels to be included in the centre-of-mass calculation. Any pixels below this fraction of the maximum pixel intensity are ignored as noise   |
| INTERACTIVE MODE      | Set to true to manually accept/reject each spot analysis result. This allows the parameters to be fine tuned until successful and then they can be applied in batch analysis  |
| INTERPOLATION         | Set the interpolation mode to use when enlarging images   |

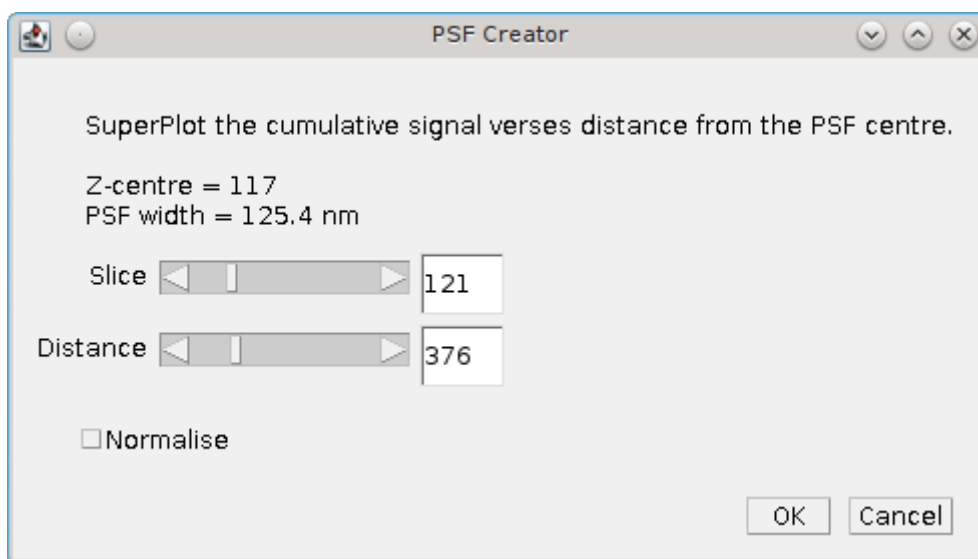
### 9.2.4 Output

The plugin will log details of each spot analysed to the `IMAGEJ` log window (e.g. centre and width). When complete the plugin will record the z-centre, scale and standard deviation of the final PSF image to the log. The plugin also fits a 2D Gaussian to the combined PSF image and records the fitted standard deviation at the z-centre as a measure of the PSF width.

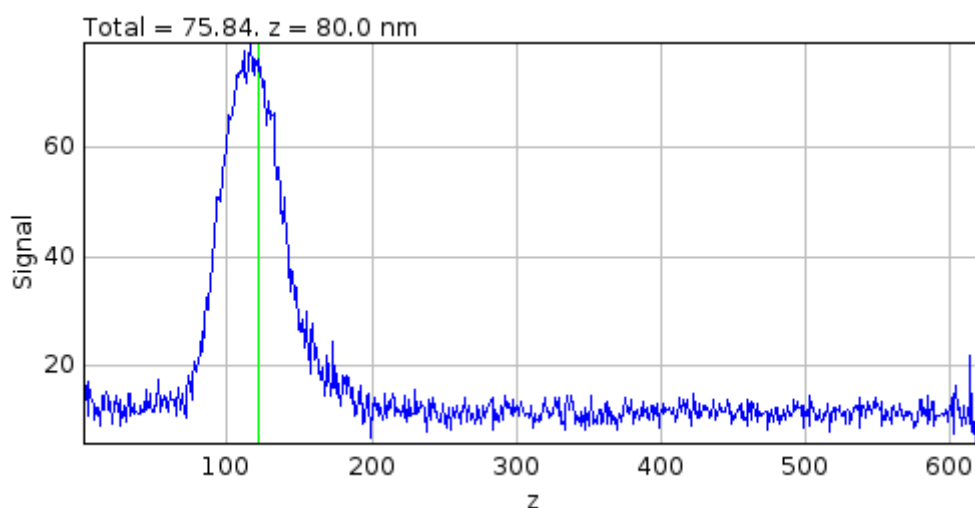
The final PSF image is shown as a new image. The z-centre is selected as the active slice. The PSF image has an XML tag added to the image info property containing the z-centre, image scale, number of input images used and the PSF width. This will be saved and reloaded when using the TIFF file format in `IMAGEJ`. This information is used in the PSF DRIFT, PSF COMBINER and CREATE DATA plugins. The information can be viewed using the `IMAGE > SHOW INFO...` command, e.g.

```
<gdsc.smlm.ij.settings.PSFSettings>
  <zCentre>453</zCentre>
  <nmPerPixel>11.428571428571429</nmPerPixel>
  <nmPerSlice>20.0</nmPerSlice>
  <nImages>1</nImages>
</gdsc.smlm.ij.settings.PSFSettings>
```

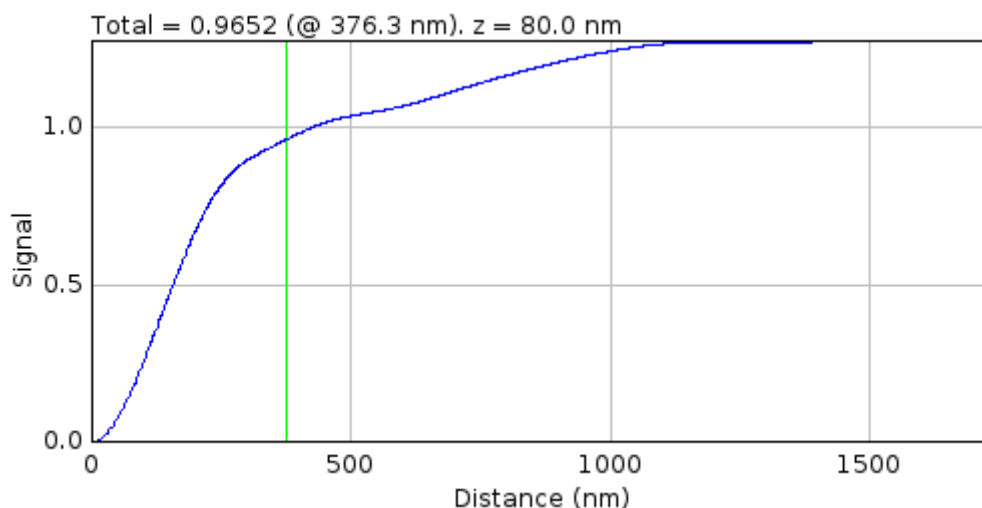
When the final PSF image has been constructed the plugin will show the Amplitude and PSF plots for the final PSF image. A dialogue is then presented allowing analysis of the PSF to be done interactively:



The `SLICE` parameter controls the current slice from the PSF image that will be analysed. The `DISTANCE` parameter controls the distance used for the cumulative signal analysis. Two additional plots are displayed and updated interactively when the `SLICE` and `DISTANCE` parameters change. One shows the percentage of the PSF signal at different z-depths that is within 3 times the standard deviation of the fitted PSF SD for the z-centre. This shows that as the spot moves out of focus less of the signal is captured within the same area. The green line shows the currently active slice:



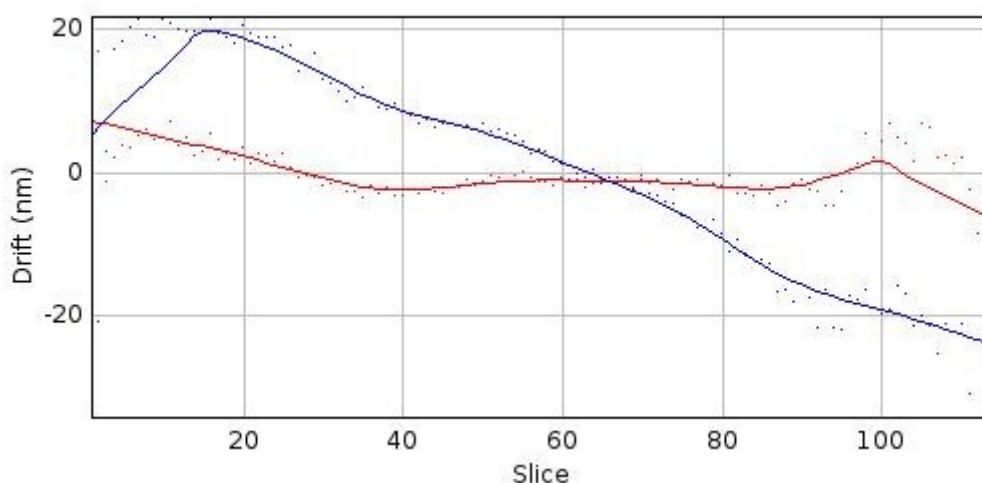
A plot is also shown of the cumulative signal as the distance from the centre of the PSF increases. This plot is drawn using data for the currently active slice in the PSF:



The green line shows the current distance selected and the total is shown in the plot label. If the NORMALISE parameter is selected then the cumulative signal up to the distance is normalised to 1 on the chart (but the label is unchanged). This plot visualises how much of the PSF signal is missed at a given distance and how the focal depth changes how the signal is distributed.

The interactive dialogue is a blocking window. It must be closed before the plots can be saved.

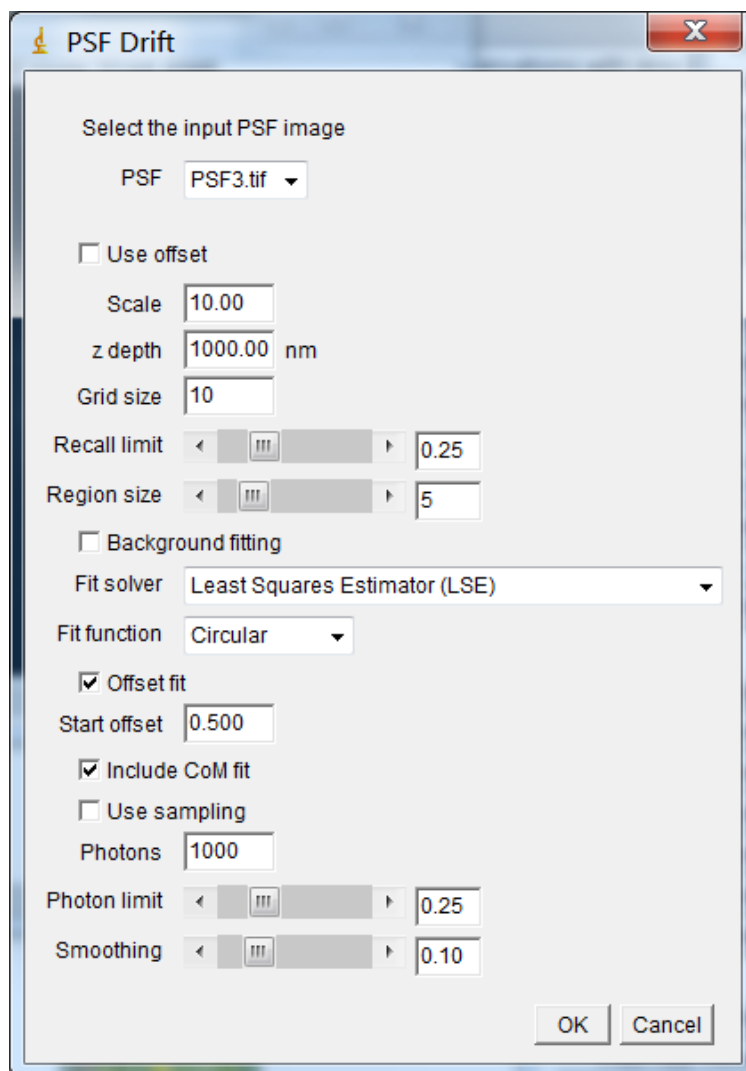
Finally the Centre-of-Mass (CoM) of the PSF is computed and shown on a plot. The CoM is computed using all pixels within a fraction of the maximum pixel intensity of the image. The default is 5%. This should avoid including noise in the CoM calculation. If the PSF is symmetric about the fitted centre then the CoM drift should be low. In the example shown below the red line (X-drift) is approximately flat but the blue line (Y-drift) shows that the PSF is skewed in the Y direction as the CoM moves past the centre determined by the fitting algorithm.



### 9.3 PSF Drift

The PSF DRIFT plugin computes the drift in a PSF image. The drift can be stored as a correction factor applied to the PSF centre when reconstructing images using the PSF.

When the plugin is run it searches all the open images for valid PSF images. These will be tagged in the image info property with PSFSettings XML containing details of the PSF. The plugin then presents a dialogue where the user can configure how to compute the drift curve:



### 9.3.1 Drift Calculation

The drift curve represents the centre of the PSF for each image in the PSF stack. This is computed by drawing the PSF into an image at a specified scale and then fitting the image with a 2D Gaussian (as per the `PEAK FIT` plugin). Note that the PSF rendering uses bilinear interpolation to scale the PSF before insertion. The integral of the scaled PSF over each output pixel is then used to set the image pixel value. Alternatively the PSF can be treated as a 2D probability distribution. The coordinates from random sampling of this distribution are then mapped to the output pixels to generate the counts for each pixel.

The PSF is drawn multiple times to reduce bias. The PSF is inserted into the image centre pixel at each point on an  $N \times N$  grid, so reducing bias from the fitting due to the location the PSF was inserted. For example a grid of 10 would insert the PSF at 100 locations spaced at 0.1 pixel intervals starting from 0 in each dimension. 100 fits would be computed and the recall (number of successful fits) recorded.

Fits are accepted if the fitting algorithm successfully converged and the fitted signal is close to the actual signal (defined by a limit of 2x higher or a user configured lower fraction).

### 9.3.2 Parameters

| Parameters         | Description  |
|--------------------|--|
| PSF                | The PSF used to compute the drift  |
| USE OFFSET         | Use an existing drift curve stored in the PSF to offset the insert location.<br><br>Note that this can be used to check that the existing drift curve is correct for the given image reconstruction and fitting settings   |
| SCALE              | The reduction scale for the PSF  |
| Z DEPTH            | The range of the PSF stack to compute the drift. z-positions outside this range will not be processed. Use this option to speed up processing when the depth-of-field of the PSF is known  |
| GRID SIZE          | The number of intervals to use to construct the NxN grid for inserting the PSF into the centre pixel   |
| RECALL LIMIT       | The fraction of fits that must be successful for a valid drift calculation   |
| REGION SIZE        | Defines the size of the image to insert the PSF into. The actual size is $2N + 1$  |
| BACKGROUND FITTING | Select this to allow the algorithm to fit the background. Note that the background should be zero as no data is inserted into the image apart from the PSF. This can be used to more closely match the fitting performed on real data  |
| FIT SOLVER         | The solver used to fit the data.<br><br>Note that a second dialog will be presented for the selected solver to be configured. The values are initially set to the defaults which should work in most cases. See the Peak Fit plugin for details of how to configure the solver |
| FIT FUNCTION       | The function used to fit the data  |
| OFFSET FIT         | Fit each image with the initial guess for the centre shifted by an offset. The guess is shifted in each of the 4 diagonal directions from the true centre  |
| START OFFSET       | The offset to use with the OFFSET FIT option   |

| Parameters      | Description  |
|-----------------|--|
| INCLUDE CoM FIT | Fit each image with the initial guess for the centre as the centre-of-mass of the pixels   |
| USE SAMPLING    | Draw the PSF by sampling it as a 2D probability distribution. The alternative is to draw it exactly using bilinear interpolation to scale the PSF        |
| PHOTONS         | The signal to draw in photons  |
| PHOTON LIMIT    | The lower fraction of the actual photons where fits will be rejected. Fits are always rejected when the photons are 2-fold higher than the actual value. |
| SMOOTHING       | The smoothing parameter used to smooth the fit curve using the LOESS smoothing algorithm   |

### 9.3.3 Output

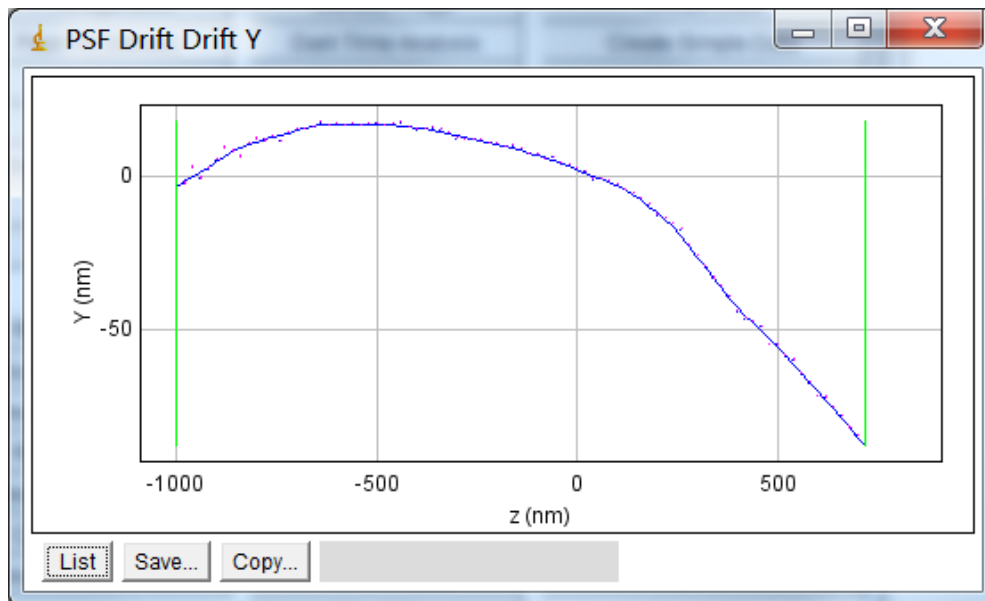
The drift for each frame is computed as the mean of all the fit centres. The curve represents the average centre of the PSF following idealised fitting of the data with the chosen FIT SOLVER and FIT FUNCTION (i.e. no noise other than Poisson noise if USE SAMPLING is enabled).

The drift curves for each dimension (X&Y) are then plotted along with the recall against the z-depth. The z-axis is limited to the input z-depth or the available depth of the PSF, whichever is lower.

### 9.3.3.1 Drift Curve

The drift curves show the original data points in blue with magenta vertical bars for the standard error of the mean. A high standard error would indicate that the curve is not a good estimate at the given point. The smoothed curve is shown as a blue line. Green vertical lines mark the points where the recall falls below the configured limit.

The following example shows a curve computed for an equivalent pixel pitch of 107nm (scaling a 10.7nm PSF 10-fold). The drift is minimal when the PSF is in focus however the fit Y-centre drifts nearly a full pixel as the PSF moves out of focus. This is due to an alignment error with the microscope optics.

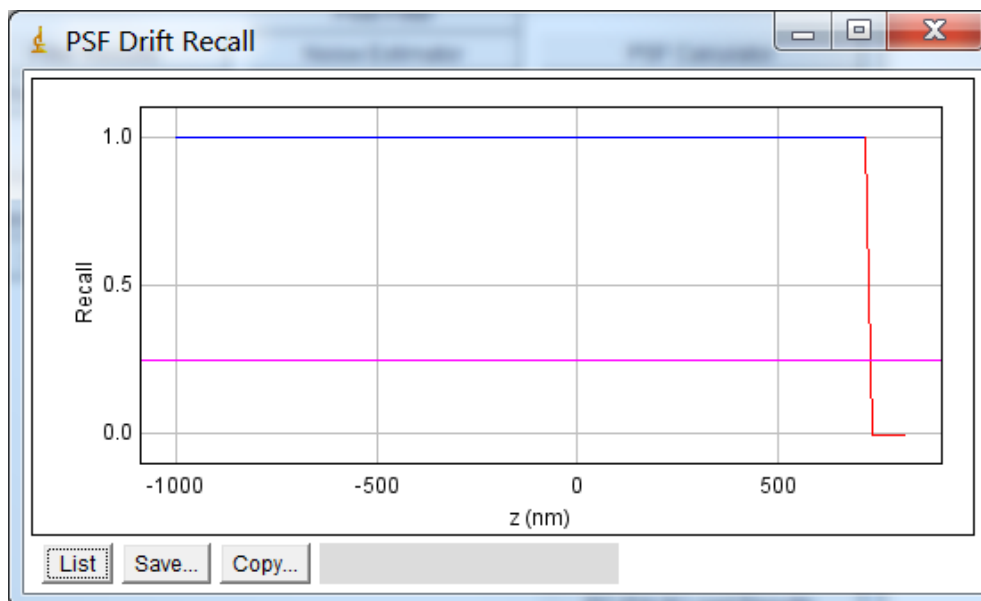


### 9.3.3.2 Recall Curve

The recall curve shows the recall above the recall limit using a blue line and below the limit using a red line. The limit is shown using a magenta line.

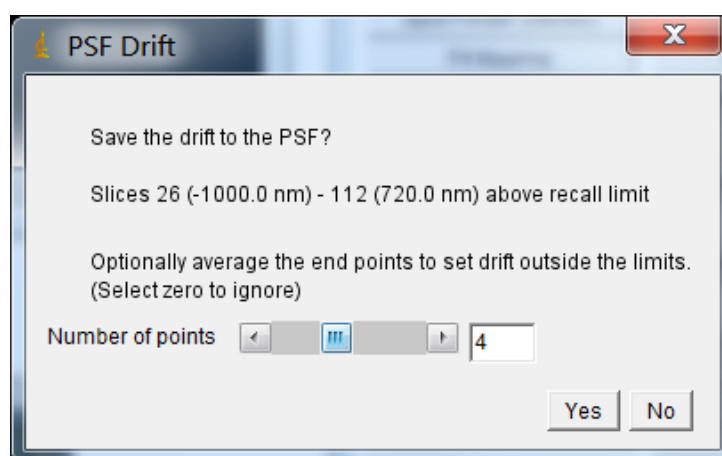
The following example shows that fitting is successful until 720 nm out-of-focus. The z-depth used for analysis could be extended as the recall is still 1 at the maximum negative depth (-1000nm).





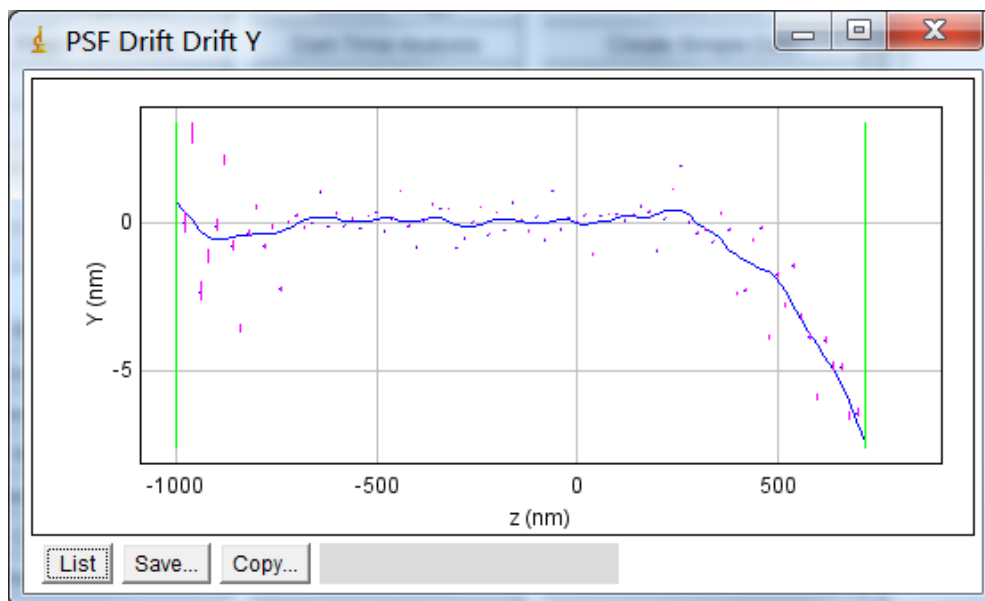
### 9.3.4 Saving the Drift

When the calculation is complete the user is presented with the option to save the curve to the PSF image. The curve is added to the PSFSettings XML stored in the Image Info data field. If the entire stack is not covered by the calculated drift curve then the plugin provides the user with the option to average the last n frames of the drift curve in each direction and store this average drift for the terminal frames:



Click YES to save the curve, or No to discard the results.

The saved drift can be used to offset the centre of each frame of the PSF when reconstructing images. This can be done when running the PSF DRIFT plugin to check the curve is correct. The following example shows a re-run of the plugin using the recently computed drift curve. Note that the maximum drift has been reduced from -87nm to -7.4nm and most of the drift is below 0.5nm.

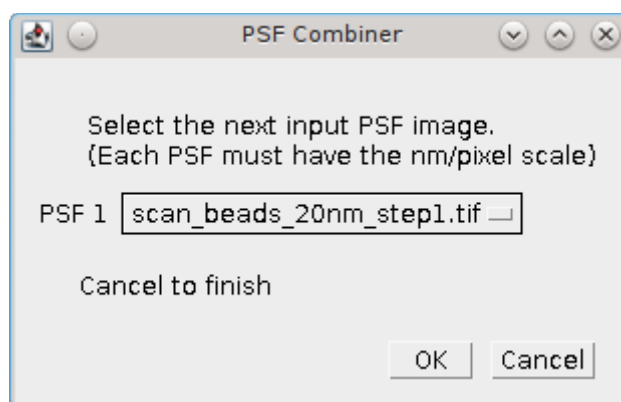


The saved drift curve is used by default in the `CREATE DATA` plugin when reconstructing images. This allows benchmarking data to be constructed by placing the localisation data at the average centre that would be found by idealised fitting of that PSF.

## 9.4 PSF Combiner

The `PSF COMBINER` plugin produces an average PSF image from multiple PSF images. PSF images can be created using the `PSF CREATOR` plugin (see section 9.2).

When the plugin is run it searches all the open images for valid PSF images. These will be tagged in the image info property with the z-centre, image scale and number of input images used to create the PSF. The plugin then presents a dialogue where the user can select the images to combine (Illustration 34). The dialogue is presented iteratively to allowing only one image to be selected from the available images each time. Select the first image from the dialogue and click `OK` to include the image. The list of available images is then updated and the dialogue reshown. Click `CANCEL` to stop adding images. Note that the iterative addition of images allows the plugin to be fully supported by the ImageJ macro recorder.



*Illustration 34: PSF Combiner image selection dialogue*

When the input images have been selected the plugin checks that each PSF has the same image scale. Note that input PSFs can have different X, Y and Z dimensions. If the scales are not the same then the images cannot be combined and an error is shown. Otherwise the plugin then presents a dialogue where the z-depth of the combined PSF can be selected. This allows the size of the output PSF to be limited to N frames above and below the z-centre.

The combined PSF is created by overlaying the x,y,z-centres and summing the individual PSF images. Each PSF is weighted using the number of images used to create the PSF divided by the total number of images:

$$weight_i = \frac{n_i}{\sum n_i}$$

The combined PSF image is shown as a new image. The z-centre is selected as the active slice. The PSF image has an XML tag added to the image info property containing the z-centre, image scale and number of input images used. This information is used in the CREATE DATA plugin. The information can be viewed using the IMAGE > SHOW INFO... command.

### 9.5 Create Data

Creates an image by simulating single molecule localisations using a model of photoactivated diffusing fluorophore complexes. The simulation is partly based on the work of Colthorpe, *et al* (2012).

#### 9.5.1 Simulation

Fluorophores initialise in an inactive state where they do not fluoresce. The switch to an active state is caused by subjecting the fluorophore to an activation laser. Once in an active state the molecule can fluoresce when subjected to a readout laser. The amount of fluorescence is proportional to the intensity of the readout laser. The active molecule can reversibly switch into a dark state where it does not emit fluorescence. Switching on and off causes the molecule to blink. Eventually the molecule will irreversibly bleach to a state where it no longer fluoresces.

Molecules are randomly positioned in a 3D volume. These are then subjected to photoactivation laser illumination and readout laser illumination. The illumination is not constant across the image but uses a radial fall-off function to simulate the darkening towards the edges of a wide-field microscope image. The light fall-off is 50% at the field edge. Illumination light and background fluorescence are subject to Poisson noise.

The read-out laser is a continuous light source. The activation laser can be continuous or pulsed. When pulsed mode is used all readout frames have a low level of activation light. This is interspersed with pulses of the activation laser at set intervals. The pulse is deemed to be a zero time event. The ratio between the amount of energy a fluorophore can receive during the pulse and between pulses can be controlled. This allows the simulation to vary the level of background activation, i.e. molecules that activate in frames not directly following an activation pulse.

The amount of photons required for photo-activation of each molecule is defined by sampling from a random exponential distribution. The average for this distribution is set using the cumulative number of photons in the centre of the field at 30% of the simulation

length. Thus approximately 50% of the molecules should have activated by 1/3 of the simulation.

The simulation allows for a single dark state or dual dark state model. For the single dark state model the fluorophore can be either on or off (dark state). The number of time the fluorophore enters the dark state is selected from a probability distribution. For the dual dark state model the fluorophore can be on or in either dark state 1 or dark state 2. The dark state can only transition between the on state. There is no transition from dark state 1 to 2 or the reverse. The number of time the fluorophore enters the 2<sup>nd</sup> dark state is selected from a probability distribution. For each time the molecule is in the on state the number of time the fluorophore enters the 1st dark state is selected from a probability distribution. The dual dark state model can be used to simulate flickering of the fluorophore at a fast rate (i.e. moving between the on state and dark state 1) broken by longer period of off time (i.e. moving between the on state and dark state 2).

The number of blinks is sampled from a Poisson or Geometric distribution and the length of time in the on-state and off-state(s) are sampled from exponential distributions. The average for these distributions are set as simulation parameters.

Analysis of the signal of mEOS3 fluorophores in yeast show that the total signal and signal-per-frame can be modelled using a Gamma distribution. Analysis of the signal-per-frame verses the time for the lifetime of the fluorophore shows that the signal is approximately constant, i.e. the signal does not get weaker over time. Consequently the signal is modelled by sampling from a Gamma distribution with an average photon emission rate per second. Alternatively a custom distribution can be used, for example inputting a set of photon budgets extracted from real data.

An additional observation from real data is that the signal-per-frame of a molecule is negatively correlated with the total on-time, i.e. molecules that are on for a shorter amount of time have a brighter signal. This may be because the release of more photons per second causes the molecule to expend the total photon budget and then photo bleach in a shorter time. Thus the simulation allows the total on-time of the fluorophore to be correlated with the average photon emission rate.

Molecules can move using diffusion. The diffusion is modelled using a random walk as described in the `DIFFUSION RATE TEST` plugin (see section 10.7). The diffusion can be random or confined to a specified volume. The diffusion can be limited to a fraction of the molecules by fixing a random sample of the molecules.

The simulation runs for a specified duration at a given time interval per simulation step. At each step the simulation calculates the new position, if diffusing, and fluorescence of the molecules. These are then drawn on an image at a specified exposure rate. The simulation interval does not have to match the exposure time of the output image. Using a shorter simulation step than the exposure time is useful when simulating diffusion molecules. The appearance of the fluorophore is modelled using a configurable point spread function (PSF).

When the molecules have been simulated the results can be filtered to remove low signal spots. This allows the `CREATE DATA` plugin to generate images at a certain signal-to-noise ratio for benchmarking experiments.

The simulation creates an ImageJ image stack and the underlying data can be saved in various formats. The raw localisations per frame are also written to a results set in memory

allowing the results of fitting the simulated image to be compared to the actual underlying data.

The simulation computes the fluorophores using a single worker thread. The time intensive rendering of the localisations as an image is multi-threaded. The number of threads uses the ImageJ setting under **EDIT > OPTIONS > MEMORY & THREADS...**

### 9.5.2 Point Spread Function

The appearance of the fluorophore is modelled using a configurable point spread function (PSF). The number of photons in the fluorophore is used to create a Poisson random variable of the number photons,  $N$ , that are actually observed. The PSF is then sampled randomly  $N$  times and each sample is mapped from the PSF coordinates on to the correct pixel in the image.

#### 9.5.2.1 Gaussian PSF

The Gaussian PSF uses a 2D Gaussian function. The width of the Gaussian is obtained from the microscope parameters (wavelength and Numerical Aperture) using the same approximation formula as the PSF CALCULATOR plugin. Alternatively the width can be specified explicitly in the plugin. The width changes using a z-defocussed exponential model. The width is scaled using the following formula:

$$Scale = 1 / (\exp(-|z| \cdot \ln(2) / zDepth))$$

where  $z$  is the  $z$  position relative to the focal plane ( $z=0$ ) and  $zDepth$  is the depth at which the width should be double. The  $zDepth$  constant is set at 500nm. This approximately matches the width spread of the PSF observed using quantum dots on the wide field super-resolution microscope at the GDSC.

PSF sampling is done by drawing a Gaussian random variable for the  $X$  and  $Y$  coordinates and then adding this location to the image.

#### 9.5.2.2 Airy PSF

The Airy PSF uses the Airy pattern to describe the PSF. The width of the Airy pattern is obtained from the microscope parameters using the same formula as the PSF CALCULATOR plugin. The Airy PSF is valid for a  $z$ -depth of zero. However the software does not implement an advanced defocussed PSF model for the Airy pattern. When defocussed the width changes using a  $z$ -defocussed exponential model as per the Gaussian PSF.

PSF sampling is done by constructing a cumulative Airy pattern (i.e. power of the Airy pattern) for all distances up to the 4<sup>th</sup> zero ring. This is approximately 95.2% of the entire Airy pattern power. Note however that the pattern diminishes gradually to infinity so sampling beyond this ring is not practical. A random sample from 0 to 1 is taken for the total Airy power. If outside the 4<sup>th</sup> zero ring it is ignored. Otherwise the radius for the power is interpolated and the radius used with a randomly orientated vector to generate the  $X$  and  $Y$  coordinates. The location is added to the image.

#### 9.5.2.3 Image PSF

The PSF image can be created using the PSF CREATOR and PSF COMBINER plugins (see sections 9.2 and 9.4).

When the plugin is run it will check all open images for the PSFSettings XML tag in the image info property. This contains details of the image pixel width and depth scales and the location of the z-centre in the image stack. If no valid images are found then the Image PSF option is not available.

The PSF image pixel scale may not match the simulation; ideally the PSF image should have a smaller pixel scale than the output image so that many pixels from the PSF cover one pixel in the output image. The resolution of the output, i.e. the accuracy of the centre of the spot, will be determined by the ratio between the two image scales. For example a PSF image of 15nm/pixel and an output width of 100nm/pixel will have a resolution of  $15/100 = 0.15$  pixels.

During initialisation the PSF image is normalised so the z-centre has a sum of 1 (and all the other slices are scaled appropriately). A cumulative image is then calculated for each slice. No cumulative image is allowed a total above 1.

PSF sampling is performed by selecting the appropriate slice from the image using the z-depth. The z-centre is specified using the middle of the slice so if the slice depth is 30nm then both -10 and 10 will be sampled from the centre slice. A random sample from 0 to 1 is taken and used to look up the appropriate pixel within the cumulative image for that slice. This sampled PSF pixel is then mapped to the output and the location added to the image. Note that if the cumulative total for the slice is below 1 then the sample may be ignored. This is allowed since the Image PSF has a limited size. Missed samples are unlikely to effect the output image as the pixels are very far from the PSF centre.

### 9.5.3 Noise model

The simulation aims to match the data produced by the pixel array of an EM-CCD camera. Photons are generated in a random process modelled by the Poisson distribution. Photons are captured on the sensor and converted to electrons. The conversion is subject to the quantum efficiency of the camera sensor modelled as a binomial distribution. The electrons are amplified through an Electron Multiplying device to increase the number. This process is subject to gamma noise. The electrons are read from the camera and digitised to Analogue-to-Digital Units (ADUs). Reading the electrons is subject to Gaussian read noise.

The simulation models the camera CCD array as a set of photo cells that will be read into pixels. The photons emitted by fluorophores are spread onto the photo cells using a point spread function. Background photons are also captured. The photons are amplified and then read into an image.

Each frame starts with an empty image. A background level of photons is sampled from a Poisson distribution and added to each pixel to simulate a background fluorescence image. Alternatively the background can be specified using an input image subject to Poisson noise. Note that at this stage the quantum efficiency is not factored in to save computational time. It is assumed all photons are captured. The camera read noise for each pixel is simulated using a Gaussian distribution. This is computed as a separate read-noise image.

Then all the localisations are processed. For each active fluorophore the total on-time is computed. If a correlation between on-time and photon emission rate is modelled a second set of on-times (tCorr) are created with a specified correlation to the actual on-times.

These are used to specify the average emission rate for each fluorophore using a proportion of the input emission rate:

$$rate_i = rate * \frac{tCorr_i}{\sum_N tCorr_i / N}$$

If no correlation is used then the emission rate is the sampled from the configured distribution (either a Gamma or custom distribution) with the mean set to the input emission rate.

The emission rate for each fluorophore is constant. The mean number of photons emitted for each simulation step is calculated using the photon emission rate multiplied by the fraction of the step that the fluorophore was active. The number of photons is then sampled from the Poisson distribution with the given mean for the step. This models the photon shot noise at a per simulation step basis. The photons are then sampled onto the photo cells using a point spread function.

When the localisation is drawn on the image the variance of all the background pixels in the affected area is computed to be used to compute the localisation noise. The variance of the background image is combined with the variance of the read-noise image to produce the total variance. The background image variance is in photons, the read noise is in electrons. The units are converted to ADUs using the appropriate gain factors. The square root of the sum of the variances is the fluorophore noise. Note that the noise value calculated is the noise that would be in the image with no fluorophore present. This is the true background noise and it is the noise that is estimated by the `PEAK FIT` plugin during fitting. This noise therefore IGNORES THE PHOTON SHOT NOISE of the fluorophore signal.

Once all the localisations have been processed the captured photons are converted to electrons by sampling from a binomial distribution with the probability set to the quantum efficiency. The electrons are amplified using the Tubbs model (Tubbs, 2004) for EM noise where the number of addition electrons generated in the EM amplification are sampled from a gamma distribution with the shape parameter equal to the input electrons and the scale parameter as:

$$scale = (EMgain - 1) + \frac{1}{electrons}$$

The amplified electrons are converted to ADUs using the camera gain. The read-noise calculated earlier (for use in the per localisation noise calculation) is then added to the image along with the camera bias. The bias offset (above zero) ensures that the final output image using unsigned integers can record negative noise values.

Note: Accurate values for the read noise, gain and EM-gain for a camera can be obtained using the `MEAN-VARIANCE TEST` plugin (see section 10.3) or the `EM-GAIN ANALYSIS` plugin (see section 10.4.4).

### 9.5.4 Particle distribution

The simulation can distribute the particle using the following methods:

| Method         | Description  |
|----------------|--|
| UNIFORM RNG    | The particle are randomly positions in the 3D volume defined by the SIZE and DEPTH parameters. The coordinates are drawn using a random number generator   |
| UNIFORM HALTON | The particle are randomly positions in the 3D volume defined by the SIZE and DEPTH parameters. The coordinates are drawn using a Halton sequence that very uniformly distributes the particles   |
| UNIFORM SOBOL  | The particle are randomly positions in the 3D volume defined by the SIZE and DEPTH parameters. The coordinates are drawn using a Sobol sequence that very uniformly distributes the particles  |
| MASK           | <p>The plugin will ask the user to select a mask image. The image must be square but width/height dimensions are scaled to match the simulation. Any stack image must have the z-depth of each slice defined so the plugin asks for the slice depth (in nm). If the slice depth is set to zero then it will be auto configured to fit the entire stack into the depth of the simulation.</p> <p>The particles will be placed randomly within a non-zero pixel selected from the mask. If a single slice is used then the z-depth uses the DEPTH parameter.</p> <p>If a stack is used then the mask slice is chosen with a frequency proportional to the number of non-zero pixels in the slice compared to the total non-zero pixels. The particle is then placed randomly in a non-zero pixel in the mask and the z-coordinate randomly selected from the slice z-depth</p> |
| GRID           | <p>Particles are placed on a grid. The plugin will ask the user to specify the grid parameters.</p> <p>The image area is divided into square cells of CELL SIZE dimensions (in pixels). A particle is placed randomly in the middle 50% of the cell. A second particle can be placed in the cell with the specified probability (P-BINARY). The second particle is randomly located from the first using a minimum and maximum distance (in nm).</p> <p>The grid distribution simulates an exact proportion of monomer/dimer localisations. The distribution can be used for benchmarking techniques for identification of single/double localisations.</p>  |

Once the particles are distributed within the volume they can move using a diffusion coefficient. To prevent the particles moving too far the diffusion can be confined using the following methods:



| Method | Description   |
|--------|---|
| MASK   | Confine the movement to a mask defined using an input image.<br>The plugin will ask the user to select a mask image. The image must be square but width/height dimensions are scaled to match the simulation. The boundaries are specified by non-zero pixels. If a single slice is used then the z-depth uses the DEPTH parameter. Any stack image must have the z-depth of each slice defined so the plugin asks for the slice depth (in nm). |
| SPHERE | Confine the movement to a sphere with the origin at the diffusion start location. The plugin will ask for the sphere radius (in nm).  |

### 9.5.5 Parameters

The following parameters can be used to control the simulation:

| Parameter       | Description  |
|-----------------|--|
| PIXEL PITCH     | The simulated size of the pixel in the image (in nm)   |
| SIZE            | The width and height of the image in pixels  |
| DEPTH           | The depth of the simulation (in nm). Molecules will only be sampled within this volume. Note that the output image is only 2D. Set to zero to have no depth simulation                                   |
| FIXED DEPTH     | Select this to use the DEPTH parameter as a fixed z-coordinate. This allows simulating out-of-focus spots  |
| SECONDS         | The duration of the simulation   |
| EXPOSURE TIME   | The exposure time for the output image   |
| STEP PER SECOND | The number of simulation steps per second  |
| ILLUMINATION    | The type of illumination for the image (i.e. specify the intensity of light across the image as uniform or radial falloff)   |
| PULSE INTERVAL  | The interval at which the activation laser is used. Set to zero to disable   |
| PULSE RATIO     | The strength of the activation laser relative to the background activation level, e.g. 100 means 100 times more activation photons in a pulse than all the frames between pulses. Set to zero to disable |

| Parameter          | Description   |
|--------------------|---|
| BACKGROUND IMAGE   | <p>Only presented if there are suitable background images. Images must be grayscale and square.</p> <p>Select the image that will form the background photon level. The image will be scaled so that the maximum value is the level defined by the BACKGROUND parameter. Each frame will use this image as the background with each pixel subject to Poisson noise.</p>   |
| BACKGROUND         | The background level in photons. This is subject to Poisson noise. Convert to actual ADU value by multiplying by the product of the camera gain, EM-gain and quantum efficiency   |
| EM GAIN            | The EM-gain of the simulated camera   |
| CAMERA GAIN        | The camera gain (in ADU/electron)   |
| QUANTUM EFFICIENCY | The efficiency converting photons to electrons in the camera  |
| READ NOISE         | The average Gaussian read noise to add to each pixel (in electrons)   |
| BIAS               | The bias offset to add to the image. Allows negative noise values to be displayed   |
| PSF MODEL          | Specify the PSF model to use. The IMAGE PSF option is only available if a valid PSF image is open   |
| ENTER WIDTH        | <p>Select this option to enter the PSF width (in nm) for the Gaussian/Airy PSF. A second dialog will prompt the user for the PSF SD (Standard Deviation). For an Airy PSF the SD is converted to the Airy pattern width by dividing by 1.323.</p> <p>If not selected a second dialog will prompt the user for the emission wavelength of the fluorophore and the numerical aperture of the microscope. These will be used to define the PSF width</p> |
| DISTRIBUTION       | The random distribution of the particles  |
| PARTICLES          | The number of molecules to simulate   |
| COMPOUND MOLECULES | Select this to allow compound molecules to be configured. See section 9.5.7: Compound molecules   |
| DIFFUSION RATE     | The diffusion rate of the molecules   |
| USE GRID WALK      | Simulate diffusion using a grid walk. Otherwise use movement along random vector (which is a slower computation)  |

| Parameter                  | Description  |
|----------------------------|--|
| FIXED FRACTION             | The fraction of molecules that will not diffuse  |
| CONFINEMENT                | Specify the confinement of the diffusing molecules   |
| P <sub>PHOTONS</sub>       | The average photon emission rate of a fluorophore.<br>The actual emission rate per fluorophore is sampled from a distribution with a mean of the P <sub>PHOTONS</sub> parameter  |
| CUSTOM PHOTON DISTRIBUTION | Select this option to choose a custom photon distribution. The distribution is specified using an input file of photon values, one per line. The photons will be sampled using a probability distribution based on these values but with a mean of the photon rate specified by the P <sub>PHOTONS</sub> parameter.<br><br>If not selected the photon distribution is sampled from a gamma distribution with the specified P <sub>PHOTON SHAPE</sub> parameter. The scale parameter of the gamma distribution will be equal to the photons divided by the shape. |
| P <sub>PHOTON SHAPE</sub>  | The shape parameter for the photon gamma distribution. If no shape parameter is provided then the photon distribution is fixed using the P <sub>PHOTONS</sub> parameter  |
| CORRELATION                | If non zero the total photon budget of a fluorophore will be proportional to the total on time using this correlation. The average photon budget will still be defined by the P <sub>PHOTONS</sub> parameter.<br><br>The CUSTOM PHOTON DISTRIBUTION and P <sub>PHOTON SHAPE</sub> parameters are ignored   |
| O <sub>N</sub> TIME        | The average on time of a fluorophore   |
| O <sub>FF</sub> TIME SHORT | The average off time of a fluorophore in dark state 1 (short)  |
| O <sub>FF</sub> TIME LONG  | The average off time of a fluorophore in the dark state 2 (long)   |
| N B <sub>LINKS</sub> SHORT | The average number of times the fluorophore enters dark state 1 from each repetition of the on-state.<br><br>Note that a blink is caused by the dark state. Set to zero to prevent blinking and all fluorophores will only activate and then bleach  |
| N B <sub>LINKS</sub> LONG  | The average number of times the fluorophore enters dark state 2 from each repetition of the on-state.<br><br>Set to zero to use a single dark state model  |

| Parameter                  | Description   |
|----------------------------|---|
| USE GEOMETRIC DISTRIBUTION | If true the blinks will be sampled from a geometric distribution, otherwise a Poisson distribution is used  |
| MIN PHOTONS                | The minimum number of photons a fluorophore must emit to be included in a time frame  |
| MIN SNR $\tau_1$           | The minimum signal-to-noise ratio for a fluorophore that is on in a single time frame   |
| MIN SNR $\tau_N$           | The minimum signal-to-noise ratio for a fluorophore that is on in consecutive time frames. In theory it should be easier to see a spot that is on for consecutive frames and so this parameter should be lower than MIN SNR $\tau_1$  |
| RAW IMAGE                  | Select this option to output an image using 32-bit floating point numbers. The default is to use 16-bit unsigned integers   |
| SAVE IMAGE                 | Show a dialog allowing the image to be saved as a file  |
| SAVE IMAGE RESULTS         | Show a dialog allowing the image localisations to be saved as a PeakResults file. Note that this does not contain the molecule Z position   |
| SAVE FLUOROPHORES          | Show a dialog allowing the fluorophores to be saved. The file contains the number of blinks and the on and off times for each fluorophore (to the thousandth of a second)   |
| SAVE LOCALISATIONS         | Show a dialog allowing the localisations to be saved. The file contains the time and X,Y,Z positions of each fluorophore when it was in an on state   |
| SHOW HISTOGRAMS            | Show histograms of the generated data   |
| CHOOSE HISTOGRAMS          | Set to true to allow the histograms to be selected, otherwise all histograms are shown  |
| HISTOGRAM BINS             | The number of bins in the histogram   |
| REMOVE OUTLIERS            | Remove outliers before plotting histograms. Outliers are 1.5 times the interquartile range above/below the upper/lower quartiles. Outliers are always removed for the Precision data since low photon signals can produce extreme precision values                            |
| DENSITY RADIUS             | Specify the radius (relative to the Half-Width at Half-Maxima, HWHM, of the PSF) to use when calculating the localisation density around each molecule. The average density is shown in the summary table. The density is the number of molecules within the specified radius |

### 9.5.6 Data Summary

The **CREATE DATA** plugin summarises the dataset when the image has been constructed. The mean of various statistics is shown. These statistics can be plotted using the **SHOW HISTOGRAMS** option. The summary shows the following fields:

| Field                     | Description   |
|---------------------------|---|
| DATASET                   | The number of the dataset   |
| MOLECULES                 | The number of fluorophore molecules that activated during the simulation  |
| PULSES                    | The number of fluorophore pulses (continuous emission from the on-state)  |
| LOCALISATIONS             | Total number of localisations. Equals the number of spots drawn on the image  |
| HWHM                      | The Half-Width at Half-Maxima (HWHM) of the PSF   |
| S                         | The standard deviation of the Gaussian equivalent of the PSF. This is the exact SD of the Gaussian PSF or if using an Airy or Image PSF it is the Gaussian that best matches the width profile of PSF   |
| SA                        | <p>The standard deviation of the Gaussian equivalent of the PSF adjusted for square pixels. The pixel adjustment is computed as:</p> $S_a = \sqrt{s^2 + a^2/12}$ <p>where <math>s</math> is the standard deviation and <math>a</math> is the pixel size, both in nanometers.</p> <p>This should be used as the input width to <b>PEAK FIT</b></p> |
| SIGNAL/FRAME              | The average signal emitted by a fluorophore per frame   |
| SIGNAL/FRAME (CONTINUOUS) | The average signal emitted by the fluorophores that were continuous for the entire frame  |
| TOTAL SIGNAL              | The average total signal for fluorophores   |
| BLINKS                    | The average number of blinks of a fluorophore   |
| T <sub>ON</sub>           | The average on time of a fluorophore  |
| T <sub>OFF</sub>          | The average off time of a fluorophore   |
| SAMPLED BLINKS            | The average number of blinks of a fluorophore if perfectly sampling at integer frame intervals (see 9.5.8)  |

| Field                    | Description  |
|--------------------------|--|
| SAMPLED T <sub>ON</sub>  | The average on time of a fluorophore if perfectly sampling at integer frame intervals (see 9.5.8)  |
| SAMPLED T <sub>OFF</sub> | The average off time of a fluorophore if perfectly sampling at integer frame intervals (see 9.5.8) |
| NOISE                    | The average noise of the region surrounding a fluorophore  |
| SNR                      | The average signal-to-noise ratio (SNR) of a fluorophore   |
| SNR (CONTINUOUS)         | The average signal-to-noise ratio (SNR) of fluorophores that were continuous for the entire frame  |
| DENSITY                  | The localisation density (calculated in the region defined by the DENSITY RADIUS parameter)        |
| P <sub>PRECISION</sub>   | The average precision (in nm)  |
| W <sub>IDTH</sub>        | The average PSF width (in pixels)  |

### 9.5.7 Compound molecules

By default all the molecules are single particles. However it may be desirable to simulate a collection of compound molecules, for example dimers and hexamers. This is possible using the COMPOUND MOLECULES option. If this option is selected the plugin will show a second dialogue where the user can input the molecule configuration using an XML specification.

The specification is a list of all the compounds that should be simulated. Each compound has a fraction parameter. The compound will be represented using the fraction divided by the total sum of all fractions to indicate the proportion of the compound. Each compound also has a diffusion parameter. When using compound molecules the DIFFUSION parameter in the main plugin dialogue is ignored. Note that the FIXED FRACTION parameter is still used to fix a fraction of the compounds. To gain more control over the moving molecules set the FIXED FRACTION parameter so zero. Then simulate a mixed population of diffusing molecules and fixed molecules by specifying the same compounds twice, one with a diffusion coefficient of zero and the other non-zero.

The remaining section of the compound specification is the list of atoms. These are fluorophore positions relative to the origin. The distances are specified in nanometres. The atom mass is used to weight the centre of mass for the compound. If omitted it is assumed all the atoms are the same.

An example compound using a 2:1 ratio of fixed monomer to moving dimer is shown below:

```
<linked-list>
  <Compound fraction="2.0" D="0.0">
    <atoms>
      <Atom mass="10.0" x="0.0" y="0.0" z="0.0"/>
    </atoms>
  </Compound>
```

## GDSC SMLM ImageJ Plugins

```
<Compound fraction="1.0" D="1.0">
  <atoms>
    <Atom mass="30.0" x="0.0" y="0.0" z="0.0"/>
    <Atom mass="20.0" x="1000.0" y="0.0" z="0.0"/>
  </atoms>
</Compound>
</linked-list>
```

When the compound is created the centre of mass is placed at the randomly chosen location. The compound can be rotated around the centre of mass. This rotation can be done once when the compound is created (ROTATE INITIAL ORIENTATION) and during the simulation (ROTATE DURING SIMULATION). If the ENABLE 2D ROTATION option is chosen then the rotation only occurs around the Z-axis, otherwise the axis is a randomly chosen unit vector. The rotation is a random angle from 0-360 degrees at each simulation step. Currently it is not possible to configure the rotation speed of the compound. Note that rotation will not occur if the molecule is part of the fixed fraction but will be allowed if the molecules is not fixed but the diffusion rate is zero.

The compound molecules dialogue contains a SHOW EXAMPLES checkbox. If this is selected the plugin will output a set of examples to the ImageJ log. Cancel the dialogue to allow the ImageJ log to be selected and use the examples as the basis for new compounds.

When the simulation is run the molecules are created and placed randomly in the 3D volume. Each atom in the compound is then modelled as a separate fluorophore. The total lifetime of the compound is computed using the first and last time any fluorophore was active. The entire compound is then subject to the diffusion (and rotation) over the lifetime of the compound.

### 9.5.8 Sampled Statistics

The simulation computes the on and off times for each fluorophore using fractions of a second. However the image is reconstructed using a specified exposure time into image frames. Thus it is possible for a fluorophore to turn off and then back on in the same frame or in consecutive frames. This will be seen on the image as a less intense spot since the fluorophore is emitting for less time. However it will not be possible to see these fast off-times since the spot emission will appear continuous.

The sampled statistics recompute the number of blinks, on and off times for integer frame sampling. Any off-time than does not completely span a frame cannot be viewed and is removed. This means that the average off-time increases as the short off-times are removed. The average on time increases as well since bursts that are joined by a short off time will be joined up to longer bursts. Finally the number of blinks is reduced because very short blinks cannot be counted.

It should be possible to experimentally compute values close to the sampled statistics by using optimised fitting parameters within the P<sub>PEAK</sub> FIT plugin and then using the BLINK ESTIMATOR plugin. An approximate number of molecules and pulses can be estimated using the TRACE MOLECULES plugin.

### 9.5.9 Memory Output

The localisations that are created are stored as various results sets in memory. Each molecule has a unique ID that is stored in each localisation. The results are named as follows:

| Suffix  | Description  |
|---|--|
| LOCALISATION DATA<br>(CREATE DATA)            | A full set of localisations with each assigned the corresponding molecule ID   |
| LOCALISATION DATA<br>(CREATE DATA PULSES)     | A set of centroids, each centroid is composed of the collection of localisations from a single molecule that were continuously visible in consecutive frames of the image. The start and end frame of the pulse is stored. |
| LOCALISATION DATA<br>(CREATE DATA No DENSITY) | The set of localisations where there was no other localisation within the radius used to calculate the density   |
| LOCALISATION DATA<br>(CREATE DATA DENSITY)    | The set of localisations where there was at least one other localisation within the radius used to calculate the density   |

It is possible to save these results to file using the RESULTS MANAGER plugin.

## 9.6 Create Simple Data

Creates an image by simulating single molecule localisations at a specified density.

The CREATE SIMPLE DATA plugin is a modification of the CREATE DATA plugin to remove the simulation of diffusing fluorophores. The simulation draws localisations on frames at a specified density until enough frames have been created to reach the desired number of localisations. Note that at least one localisation is drawn per frame so to achieve a very low density will require using a large image size.

The number of photons per localisation is randomly sampled from the range specified by the minimum to the maximum photons parameters. The output of the plugin is an image and summary table as per the CREATE DATA plugin.

The CREATE SIMPLE DATA plugin records the details of the simulation in memory. This data can be used with the FILTER SPOT DATA plugin to filter the image to identify candidate localisations and report statistics on the recall and precision of the results.

The following parameters can be configured:

| Parameter   | Description  |
|-------------|--|
| PIXEL PITCH | The simulated size of the pixel in the image (in nm)   |
| SIZE        | The width and height of the image in pixels  |
| DEPTH       | The depth of the simulation (in nm). Molecules will only be sampled within this volume. Note that the output image is only 2D. Set to zero to have no depth simulation |



| Parameter          | Description   |
|--------------------|---|
| FIXED DEPTH        | Select this to use the <code>DEPTH</code> parameter as a fixed z-coordinate. This allows simulating out-of-focus spots  |
| BACKGROUND         | The background level in photons. This is subject to Poisson noise. Convert to actual ADU value by multiplying by the product of the camera gain, EM-gain and quantum efficiency   |
| EM GAIN            | The EM-gain of the simulated camera   |
| CAMERA GAIN        | The camera gain (in ADU/electron)   |
| QUANTUM EFFICIENCY | The efficiency converting photons to electrons in the camera  |
| READ NOISE         | The average Gaussian read noise to add to each pixel (in electrons)   |
| BIAS               | The bias offset to add to the image. Allows negative noise values to be displayed   |
| PSF MODEL          | Specify the PSF model to use. The <code>IMAGE PSF</code> option is only available if a valid PSF image is open  |
| ENTER WIDTH        | <p>Select this option to enter the PSF width (in nm) for the Gaussian/Airy PSF. A second dialog will prompt the user for the PSF SD (Standard Deviation). For an Airy PSF the SD is converted to the Airy pattern width by dividing by 1.323.</p> <p>If not selected a second dialog will prompt the user for the emission wavelength of the fluorophore and the numerical aperture of the microscope. These will be used to define the PSF width</p> |
| DISTRIBUTION       | <p>The random distribution of the particles.</p> <p>See 9.5.4: Particle distribution (Note: Some distributions are not supported)</p>   |
| SAMPLE PER FRAME   | The density is used to compute the number of molecules to draw per frame ( $N$ ). The default is to use a fixed number in each frame by rounding $N$ . Select this option to sample each frame using a Poisson distribution with a mean of $N$  |
| PARTICLES          | The number of molecules to simulate   |
| DENSITY            | The density of the molecules in squared micrometres   |
| MIN PHOTONS        | The minimum number of photons for a localisation  |
| MAX PHOTONS        | The maximum number of photons for a localisation  |

| Parameter          | Description   |
|--------------------|---|
| RAW IMAGE          | Select this option to output an image using 32-bit floating point numbers. The default is to use 16-bit unsigned integers   |
| SAVE IMAGE         | Show a dialog allowing the image to be saved as a file  |
| SAVE IMAGE RESULTS | Show a dialog allowing the image localisations to be saved as a PeakResults file. Note that this does not contain the molecule Z position   |
| SAVE LOCALISATIONS | Show a dialog allowing the localisations to be saved. The file contains the time and X,Y,Z positions of each fluorophore when it was in an on state   |
| SHOW HISTOGRAMS    | Show histograms of the generated data   |
| CHOOSE HISTOGRAMS  | Set to true to allow the histograms to be selected, otherwise all histograms are shown  |
| HISTOGRAM BINS     | The number of bins in the histogram   |
| REMOVE OUTLIERS    | Remove outliers before plotting histograms. Outliers are 1.5 times the interquartile range above/below the upper/lower quartiles. Outliers are always removed for the Precision data since low photon signals can produce extreme precision values                            |
| DENSITY RADIUS     | Specify the radius (relative to the Half-Width at Half-Maxima, HWHM, of the PSF) to use when calculating the localisation density around each molecule. The average density is shown in the summary table. The density is the number of molecules within the specified radius |

## 9.7 Create Benchmark Data

Creates an image by simulating single molecule localisations in a fixed location

The **CREATE BENCHMARK DATA** plugin is a modification of the **CREATE DATA** plugin to remove the simulation of diffusing fluorophores. The simulation draws a single localisation on each frame at a fixed position (relative to the image centre).

The number of photons per localisation is randomly sampled from the range specified by the minimum to the maximum photons parameters. The output of the plugin is an image and summary table as per the **CREATE DATA** plugin.

If the minimum and maximum photon limits are the same the **CREATE BENCHMARK DATA** plugin records the details of the simulation in memory. This includes the exact number of photons for each localisation. This data can be used with the **FIT BENCHMARK DATA** plugin to fit the localisations and report statistics on the accuracy and precision of the fit results.

The following parameters can be configured:

| Parameter          | Description   |
|--------------------|---|
| PIXEL PITCH        | The simulated size of the pixel in the image (in nm)  |
| SIZE               | The width and height of the image in pixels   |
| BACKGROUND         | The background level in photons. This is subject to Poisson noise. Convert to actual ADU value by multiplying by the product of the camera gain, EM-gain and quantum efficiency   |
| EM GAIN            | The EM-gain of the simulated camera   |
| CAMERA GAIN        | The camera gain (in ADU/electron)   |
| QUANTUM EFFICIENCY | The efficiency converting photons to electrons in the camera  |
| READ NOISE         | The average Gaussian read noise to add to each pixel (in electrons)   |
| BIAS               | The bias offset to add to the image. Allows negative noise values to be displayed   |
| PSF MODEL          | Specify the PSF model to use. The IMAGE PSF option is only available if a valid PSF image is open   |
| ENTER WIDTH        | <p>Select this option to enter the PSF width (in nm) for the Gaussian/Airy PSF. A second dialog will prompt the user for the PSF SD (Standard Deviation). For an Airy PSF the SD is converted to the Airy pattern width by dividing by 1.323.</p> <p>If not selected a second dialog will prompt the user for the emission wavelength of the fluorophore and the numerical aperture of the microscope. These will be used to define the PSF width</p> |
| PARTICLES          | The number of molecules to simulate   |
| X POSITION         | The X position relative to the centre of the image (in nm). To place the localisation in the centre of a pixel set the SIZE parameter to an odd number  |
| Y POSITION         | The Y position relative to the centre of the image (in nm). To place the localisation in the centre of a pixel set the SIZE parameter to an odd number  |
| Z POSITION         | The Z position relative to the focal plane (in nm)  |
| MIN PHOTONS        | The minimum number of photons for a localisation  |
| MAX PHOTONS        | The maximum number of photons for a localisation  |

| Parameter          | Description   |
|--------------------|---|
| RAW IMAGE          | Select this option to output an image using 32-bit floating point numbers. The default is to use 16-bit unsigned integers   |
| SAVE IMAGE         | Show a dialog allowing the image to be saved as a file  |
| SAVE IMAGE RESULTS | Show a dialog allowing the image localisations to be saved as a PeakResults file. Note that this does not contain the molecule Z position   |
| SAVE LOCALISATIONS | Show a dialog allowing the localisations to be saved. The file contains the time and X,Y,Z positions of each fluorophore when it was in an on state   |
| SHOW HISTOGRAMS    | Show histograms of the generated data   |
| CHOOSE HISTOGRAMS  | Set to true to allow the histograms to be selected, otherwise all histograms are shown  |
| HISTOGRAM BINS     | The number of bins in the histogram   |
| REMOVE OUTLIERS    | Remove outliers before plotting histograms. Outliers are 1.5 times the interquartile range above/below the upper/lower quartiles. Outliers are always removed for the Precision data since low photon signals can produce extreme precision values                            |
| DENSITY RADIUS     | Specify the radius (relative to the Half-Width at Half-Maxima, HWHM, of the PSF) to use when calculating the localisation density around each molecule. The average density is shown in the summary table. The density is the number of molecules within the specified radius |

### 9.7.1 Fitting Limits

The CREATE BENCHMARK DATA plugin will report the theoretical limit (precision) for fitting the signal (number of photons) and the XY coordinates (localisation). These limits use the formulas of Thompson *et al* (2002) for the signal and Mortensen *et al* (2010) for the localisation (see Appendix A : Localisation Precision).

Note that these formulas are derived from modelling the point spread function (PSF) as a 2D Gaussian for both the simulation and the fitting. Given that the true data will have a PSF defined by the microscope parameters these formulas only approximate the precision that can be obtained on image data. However they are useful to allow demonstration that the fitting routines in the SMLM plugins can achieve the theoretical limit, i.e. they are working as well as can be expected.

### 9.8 Fit Benchmark Data

Fit the image created by CREATE BENCHMARK DATA and compute statistics on the accuracy

and precision of fitting.

The **FIT BENCHMARK DATA** plugin will fit a stack image of localisations all created at the same coordinates. This image must be created by the **CREATE BENCHMARK DATA** plugin as the parameters used to create the image are stored in memory and used in the analysis.

The plugin allows the size of the fitting region around the localisation, the fitting method and the initial guess for the fit to be configured. The fitting process is performed and the fit rejected if the coordinates move outside the fitting region, the signal is negative or the fitted width deviates more than 2-fold from the initial estimate. All other results are stored for analysis.

The following parameters can be configured.

| Parameter          | Description  |
|--------------------|--|
| REGION SIZE        | The size of the region around the localisation used for fitting. The actual region is a square of dimensions $2n+1$  |
| PSF WIDTH          | The initial estimate for the 2D Gaussian   |
| FIT SOLVER         | The solver used for fitting.<br><br>The plugin will show a second dialog allowing the fitting to be configured that is specific to the selected solver. Details of configuring each fit solver can be found in the section describing the <b>PEAK FIT</b> plugin (section 5.2) |
| FIT FUNCTION       | The function used for fitting.   |
| OFFSET FIT         | Select this to start fitting at a distance offset from the true localisation centre. The fitting repeated 4 times using the <b>START OFFSET</b> along each of the diagonals (1,1), (1,-1), (-1,1), (-1,-1)   |
| START OFFSET       | The distance to offset the initial estimate  |
| INCLUDE CoM FIT    | Perform fitting by starting at the centre-of-mass of the fit region  |
| BACKGROUND FITTING | Select this to fit the background. If false then fitting will fix the background parameter using the true background   |
| SIGNAL FITTING     | Select this to fit the signal. If false then fitting will fix the signal parameter using the true signal   |
| SHOW HISTOGRAMS    | Show histograms of the results (the difference between the fit results and the true answer).<br><br>If selected a second dialogue is shown allowing the user to choose which histograms to display   |

### 9.8.1 Data Summary

The **FIT BENCHMARK DATA** plugin will compute the difference between the fit result and the

true answer for each parameter that was fitted. Histograms of the differences can be shown using the `SHOW HISTOGRAMS` options. The average and standard deviation are then reported in a summary table:

| Field      | Description   |
|------------|---|
| MOLECULES  | The number of localisations in the benchmark data   |
| N          | The average number of photons per localisation  |
| S          | The standard deviation of the Gaussian profile that matches the PSF                               |
| A          | The pixel size  |
| SA         | The standard deviation adjusted for square pixels computed as:<br>$S_a = \sqrt{s^2 + a^2/12}$     |
| X          | The X position of the localisations, relative to the centre of the image                          |
| Y          | The Y position of the localisations, relative to the centre of the image                          |
| GAIN       | The total gain of the simulation  |
| READNOISE  | The read noise of the simulation  |
| B          | The background number of photons  |
| B2         | The noise per pixel. This is a combination of the read noise and the background number of photons |
| SNR        | The signal to noise ratio ( $N/\sqrt{b2}$ )   |
| LIMIT N    | The theoretical limit of signal precision   |
| LIMIT X    | The theoretical limit of localisation precision for Least Squares Fitting                         |
| LIMIT X ML | The theoretical limit of localisation precision for Maximum Likelihood fitting                    |
| REGION     | The actual size of the region used for fitting  |
| WIDTH      | The PSF width used for fitting  |
| METHOD     | The method used for fitting   |
| OPTIONS    | Additional options for the fitting method   |

| Field               | Description   |
|---------------------|---|
| RECALL              | The fraction of localisations that were successfully fitted. This does not indicate that the fitting was good, only that it returned a result   |
| dB & +/-            | The average and standard deviation of the difference of the fit to the actual background  |
| dB & +/-            | The average difference of the fit to the average signal   |
| DSIGNAL & +/-       | The average and standard deviation of the difference of the fit to the average signal   |
| DANGLE & +/-        | The average and standard deviation of the difference of the fit to the angle. All simulations will use a circular PSF so the actual angle is assumed to be zero. This is only reported for the FREE fit function  |
| DX & +/-            | The average and standard deviation of the difference of the fit to the actual X position  |
| DY & +/-            | The average and standard deviation of the difference of the fit to the actual Y position  |
| DSX & +/-           | The average and standard deviation of the difference of the fit to the actual PSF standard deviation in the X dimension   |
| DSY & +/-           | The average and standard deviation of the difference of the fit to the actual PSF standard deviation in the Y dimension. This is only reported for the FREE and FREE CIRCULAR fit functions   |
| TIME & +/-          | The average and standard deviation of the time for fitting per localisation   |
| DACTUALSIGNAL & +/- | The average and standard deviation of the difference of the fit to the actual signal for the localisation (Note that the CREATE BENCHMARK DATA plugin stores the number of photons that were simulation per localisation after Poisson sampling to allow this comparison) |
| DSAX & +/-          | The average and standard deviation of the difference of the fit to the actual PSF standard deviation in the X dimension, adjusted for square pixels   |
| DSAY & +/-          | The average and standard deviation of the difference of the fit to the actual PSF standard deviation in the Y dimension, adjusted for square pixels. This is only reported for the FREE and FREE CIRCULAR fit functions   |

## 9.9 Benchmark Analysis

Compute statistics on the accuracy and precision of fitting using different methods. Statistics are only computed for all the localisations that were fit successfully by each method.

The **BENCHMARK ANALYSIS** plugin can be used to compare different fitting methods on the same benchmark data. Note that if the **FIT BENCHMARK DATA** plugin is run for multiple fitting methods the recall may be different for each method. This means that a more robust fitting method that has a higher recall may not be fairly compared against a poor fitting method that fails on all the difficult targets. The poor method may have an artificially inflated precision because the worst fits were discarded.

A fair comparison between fitting methods is to store all the fitting results in memory. Then when several different fitting methods have been run the average and standard deviation statistics are recomputed only for those localisations that were successfully fit by every method.

The **BENCHMARK ANALYSIS** plugin requires that the **FIT BENCHMARK DATA** plugin has been run at least twice for a given set of benchmark data. If a new set of benchmark data is created then any results in memory will be discarded. The plugin will analyse the data for all localisations successfully fit by each method and produce a summary table as per the **FIT BENCHMARK DATA** plugin. The only difference is an additional column after the **RECALL** column that provides the original recall (**ORIGRECALL**). The recall column should be the same for each fitting method and in some cases may be much lower than the original recall indicating that some of the fitting methods have performed much worse than others. It is now possible to judge the accuracy and precision of each method in a fair comparison on an equivalent dataset.

### 9.10 Create Spot Data

Creates a sparse image by simulating zero or one localisation per frame

The **CREATE SPOT DATA** plugin is a modification of the **CREATE DATA** plugin to remove the simulation of diffusing fluorophores. The simulation draws localisations on 50% of the image frames. Note that a maximum of one localisation is drawn per frame. Each localisation is randomly positioned in the central 50% of the image.

The number of photons per localisation is randomly sampled from the range specified by the minimum to the maximum photons parameters. The output of the plugin is an image and summary table as per the **CREATE DATA** plugin.

The **CREATE SPOT DATA** plugin records the details of the simulation in memory. This data can be used with the **FILTER SPOT DATA** plugin to filter the image to identify candidate localisations and report statistics on the recall and precision of the results.

The following parameters can be configured:

| Parameter   | Description  |
|-------------|--|
| PIXEL PITCH | The simulated size of the pixel in the image (in nm) |
| SIZE        | The width and height of the image in pixels          |



| Parameter          | Description  |
|--------------------|--|
| DEPTH              | The depth of the simulation (in nm). Molecules will only be sampled within this volume. Note that the output image is only 2D. Set to zero to have no depth simulation   |
| FIXED DEPTH        | Select this to use the DEPTH parameter as a fixed z-coordinate. This allows simulating out-of-focus spots  |
| BACKGROUND         | The background level in photons. This is subject to Poisson noise. Convert to actual ADU value by multiplying by the product of the camera gain, EM-gain and quantum efficiency  |
| EM GAIN            | The EM-gain of the simulated camera  |
| CAMERA GAIN        | The camera gain (in ADU/electron)  |
| QUANTUM EFFICIENCY | The efficiency converting photons to electrons in the camera   |
| READ NOISE         | The average Gaussian read noise to add to each pixel (in electrons)  |
| BIAS               | The bias offset to add to the image. Allows negative noise values to be displayed  |
| PSF MODEL          | Specify the PSF model to use. The IMAGE PSF option is only available if a valid PSF image is open  |
| ENTER WIDTH        | Select this option to enter the PSF width (in nm) for the Gaussian/Airy PSF. A second dialog will prompt the user for the PSF SD (Standard Deviation). For an Airy PSF the SD is converted to the Airy pattern width by dividing by 1.323.<br><br>If not selected a second dialog will prompt the user for the emission wavelength of the fluorophore and the numerical aperture of the microscope. These will be used to define the PSF width |
| PARTICLES          | The number of molecules to simulate  |
| DENSITY            | The density of the molecules in squared micrometres  |
| MIN PHOTONS        | The minimum number of photons for a localisation   |
| MAX PHOTONS        | The maximum number of photons for a localisation   |
| RAW IMAGE          | Select this option to output an image using 32-bit floating point numbers. The default is to use 16-bit unsigned integers  |
| SAVE IMAGE         | Show a dialog allowing the image to be saved as a file   |

| Parameter          | Description  |
|--------------------|--|
| SAVE IMAGE RESULTS | Show a dialog allowing the image localisations to be saved as a PeakResults file. Note that this does not contain the molecule Z position  |
| SAVE LOCALISATIONS | Show a dialog allowing the localisations to be saved. The file contains the time and X,Y,Z positions of each fluorophore when it was in an on state  |
| SHOW HISTOGRAMS    | Show histograms of the generated data  |
| CHOOSE HISTOGRAMS  | Set to true to allow the histograms to be selected, otherwise all histograms are shown   |
| HISTOGRAM BINS     | The number of bins in the histogram  |
| REMOVE OUTLIERS    | Remove outliers before plotting histograms. Outliers are 1.5 times the interquartile range above/below the upper/lower quartiles. Outliers are always removed for the Precision data since low photon signals can produce extreme precision values |

### 9.11 *Filter Spot Data*

Filter the image created by `CREATE SIMPLE DATA` or `CREATE SPOT DATA` and compute statistics on the accuracy and precision of identifying spot candidates

The `FILTER SPOT DATA` plugin will filter a stack image of localisations to identify candidates for fitting. This image must be created by the `CREATE SIMPLE DATA` or `CREATE SPOT DATA` plugin as the parameters used to create the image are stored in memory and used in the analysis.

The plugin allows the spot filter and the analysis settings to be configured. The filtering is then performed on each frame in the image, candidates in the configured border are ignored and the combined list of spot candidates ranked. This ranking depends on the filter but is usually done using the pixel value in the filtered image at the spot candidate position. The ranking will be the same as that used when performing fitting within the `PEAK FIT` plugin, only on all the candidates at the same time and not on the candidates per frame (since `PEAK FIT` processes and fits frames individually). The ranked spot candidates are then analysed to produce scoring metrics of the filter performance by comparison to the simulated spot positions.

The two sets of coordinates are assigned as matches if they are within a configured distance. By default matching is performed using a nearest-neighbour algorithm assigning the closest pairs. This can be changed to process the candidates in order, attempting to find a free localisation that is within the match distance to the candidate. This simulates the way the candidates will be used during fitting with the highest ranked candidates being processed first, thus they will be likely to be fit to a localisation, even if it has a closer candidate (as that will be processed later). Optionally multiple matching can be used where candidate can be matched to more than one localisation. This is useful when simulating high density data with overlapping localisations.

In addition the spot candidates can be assessed as matches if they are within a factor of the true signal for the localisation (Note: all spot filters produce an estimate of the signal, for example using the maximum height of the spot). The relative signal factor is simply the estimated spot signal divided by the true signal. The relative signal factor (rsf) above or below the true signal (which has 1 for a perfect match) is adjusted so that the new signal factor (sf) score is 0 for a perfect match:

$$sf = (rsf < 1) ? 1 - \frac{1}{rsf} : rsf - 1$$

The matches are then assigned a score. The score is created using a ramped function between the LOWER DISTANCE and the MATCH DISTANCE. Any distance below the LOWER DISTANCE is 1. Anything above the MATCH DISTANCE is 0. In between uses a ramp to set the score. The ramp is not linear but uses a cosine function. This smooths the transition at the ends of the range to make selection of the end points for assessment less critical; the ramp is linear in the exact centre of the range. Thus the ideal end points should be above and below the ideal scoring threshold. If using a signal factor a similar ramp is applied to produce a signal score and the final score computed as a product of the two. The match score is then used to accumulate a score for how accurate the fitting was performed.

The match score is counted as true positive. The remaining false positive score is calculated as 1 – score.

Note that the analysis optionally allows a spot candidate to match multiple spots. This is useful when spot density is high and true localisations overlap. In this case the TP score is the sum of individual TP scores. The FP score is 1-TP score but limited to a minimum of 0. Thus a spot candidate can be rewarded for being closest to 2 or more actual localisations but never penalised for being more than a single candidate.

The performance is then calculated by summing:

- The true positives (TP) as the sum of the score for the positives
- The false positives (FP) as the sum of the 1-TP score for all candidates, limited to 0 in the case of multiple matches. If a candidate does not match anything then TP=0 and FP=1
- The false negatives (FN) as the total number of localisations minus the TP total (i.e. how many localisations were missed)

The TP, FP and FN totals can be used to produced scoring metrics to assess the filters.

Note that the use of a ramped score function based on distance (and signal factor) allows the comparison of scores between different filters, since some algorithms may identify spot candidates closer to the true localisation. Also note that if it is not clear at what level to set the match distance and signal factor then using a ramped score will produce the same results as repeating the analysis with multiple thresholds and averaging the score with the same ramped weighting for each scoring threshold.

The following parameters can be configured.

| Parameter          | Description   |
|--------------------|---|
| SPOT FILTER TYPE   | The type of filter to use.<br><br>If a DIFFERENCE or JURY filter is selected then the plugin will present an additional dialogue to configure each additional spot filter   |
| SPOT FILTER        | The name of the first spot filter   |
| RELATIVE DISTANCES | Set to true to make all distances relative to the Half-Width at Half-Maxima (HWHM) of the PSF used to create the data. This is the same as when using the P <sub>EA</sub> K FIT plugin. This applies to the spot identification, and also the match distance. Note that the actual distances used will be recorded in the results table.<br><br>Set to true to use absolute distances enabling fine control over the tested settings. Distances will be in pixels |
| SMOOTHING          | The smoothing parameter for the first filter  |
| SEARCH WIDTH       | Define the region within which to search for a local maxima. The region size is $2n+1$ . This must be at least 1  |
| BORDER             | Define the number of border pixels to ignore. No maxima are allowed in the border   |
| ANALYSIS BORDER    | Define the number of border pixels to ignore during the analysis. Any true or candidate maxima within this border are ignored   |
| MULTIPLE MATCHES   | Set to true to allow a candidate to match more than one localisation  |
| RANKED MATCHES     | Set to true to try and find a matching localisation for each spot candidate, processing the candidates in their ranked order. The default is to simply match the closest candidate and localisation, irrespective of the candidate rank   |
| MATCH DISTANCE     | The maximum allowed distance between a true and candidate maxima to be classed as a match   |
| LOWER DISTANCE     | The distance limit defining the maximum score for a match between a fitted localisation and the true localisation.<br><br>Set to the same as the MATCH DISTANCE to ignore the ramped scoring function   |
| MATCH SIGNAL       | Define the limit for the difference between the estimated signal and the actual signal for a match. A value of (N-1) indicates the fit is allowed to be N-fold different, i.e. use 2 for a 3-fold difference.<br><br>Set to zero to ignore  |

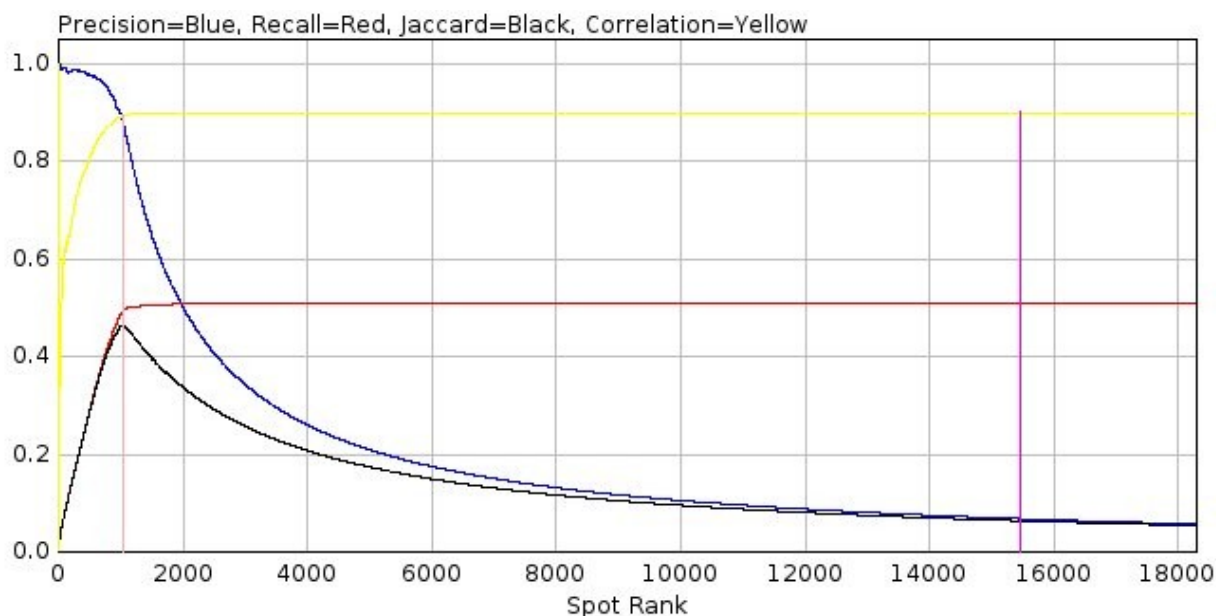
| Parameter              | Description  |
|------------------------|--|
| LOWER SIGNAL           | The limit defining the maximum score for a match between the estimated signal and the true signal.<br><br>Set to the same as the MATCH SIGNAL to ignore the ramped scoring function  |
| RECALL FRACTION        | The fraction of the maximum recall to use for calculating the second set of recall and precision scores  |
| SHOW PLOTS             | Select this option to show plots of the results (the match statistics verses the spot candidate rank and the precision-recall curve)   |
| PLOT RANK BY INTENSITY | The default x-axis rank when plotting is an integer series starting from 1.<br><br>Set this option to true to show the ranking using the candidate spot intensity relative to the intensity range of all the spots. Candidates are ordered by intensity so the order will be the same, the relative positions will change as many candidates have low signal |
| SHOW FAILURE PLOTS     | Select this option to show a histogram of the count of false positives before each true positive (i.e. failures)   |

### 9.11.1 Data Summary

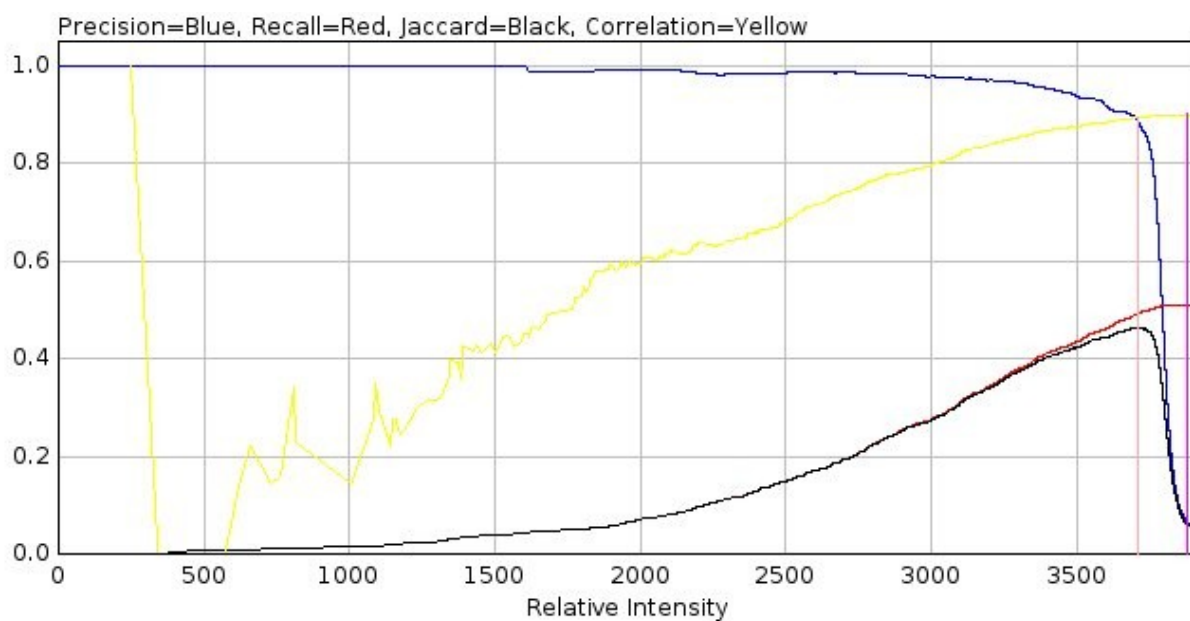
The FILTER SPOT DATA plugin will compare the spot candidates to the actual localisation positions in each frame. Any spot candidate that is within the MATCH DISTANCE of an actual localisation is marked as a match (true positive). The comparison is done using a closest pairs algorithm where the first pair are eliminated from subsequent pairings. All other spot candidates that fail to match an actual localisation are false positives. The recall, precision and Jaccard score (see Appendix B : Comparison Metrics) are computed for the entire collection.

Given that the spot candidates are ranked, it is possible to compute the scoring metrics for each additional spot added to the collection starting from the highest ranked spot. A plot is shown of the recall, precision and Jaccard score against the rank. The correlation between the candidate intensity and actual intensity is also shown:

## GDSC SMLM ImageJ Plugins

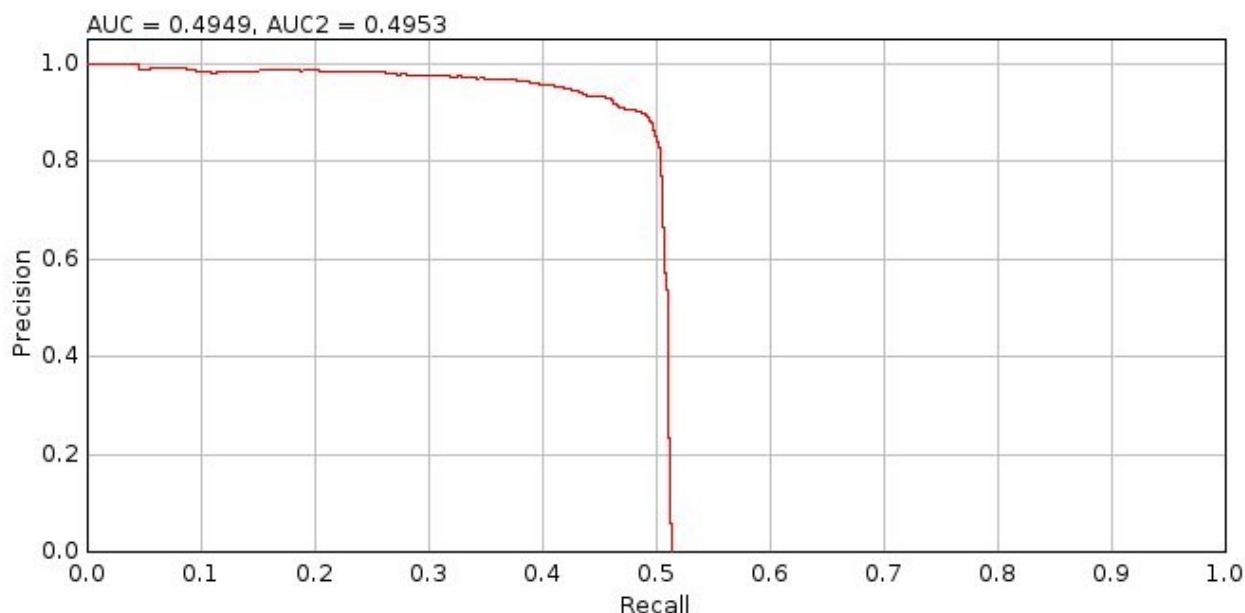


The same plot using the **PLOT RANK BY INTENSITY** option:

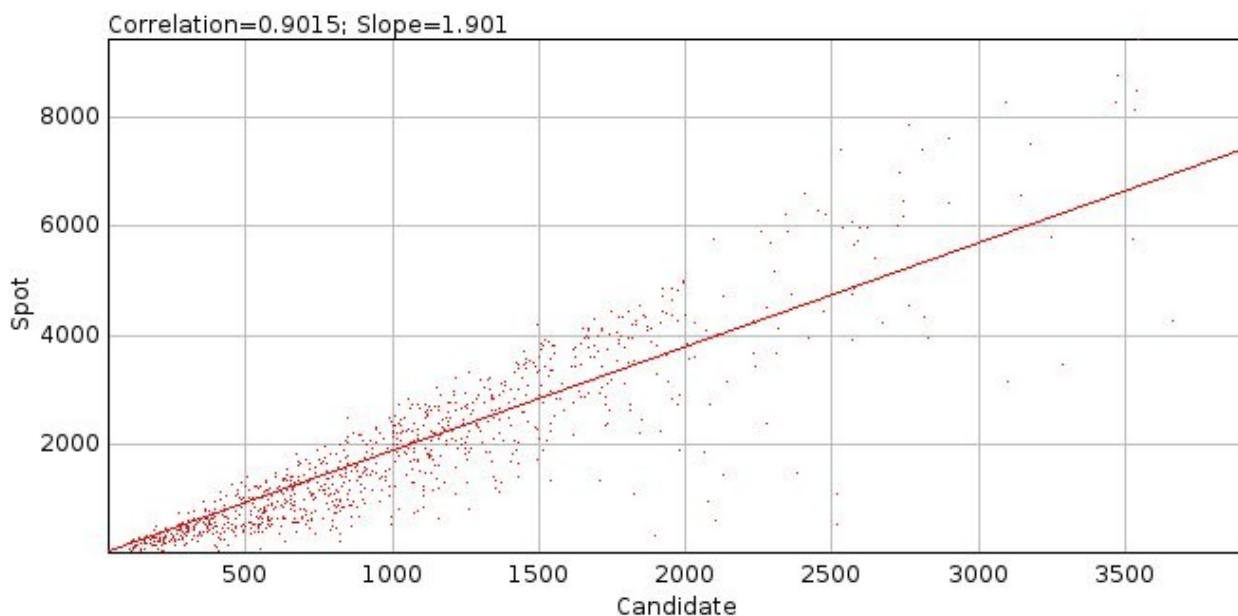


Since the spot filter will identify many false positives the overall precision score is very low. Thus the plugin also reports the precision at a fraction of the maximum recall. This fraction can be configured and is shown on the plot using a magenta line. In the example above the line is drawn at 98% of the maximum recall.

In addition the precision can be plotted directly against the recall. The **area under the precision-recall curve (AUC)** is the average precision at each possible recall value. This provides a single score on how good the filter is at finding and ranking spot candidates. An example precision-recall chart is shown below. The AUC2 score is computed using a modified precision curve that uses the highest precision at that recall or above. This smooths a noisy curve that may occur at low recall values. The AUC2 score is always above the AUC score.



A scatter plot of the candidate intensity against the true spot intensity is shown for all matches. The correlation and slope of the line (assuming an intercept at 0,0) is shown:



An ideal filter should produce a slope of 1 meaning that the estimated signal is a good starting approximation for a fitting algorithm. However as the smoothing increases the intensity of spots is reduced due to merging of the spot pixels with surrounding background. Thus higher smoothing values tend to increase the slope as the intensity is underestimated.

An additional analysis is performed using the ranked candidates. For each frame in the input image a count is made of the number of false positives (failures) before each additional true positive. The counts are plotted as a histogram and a cumulative histogram if the `SHOW FAILURE PLOTS` option is selected. The cumulative histogram is then used to determine the number of consecutive false positives to accept to achieve a set fraction of the maximum true positives. The fractions are reported for 80, 80, 95, 99 and 100%. This value can be used to determine the `FAIL LIMIT` parameter for the `PEAK FIT` plugin for different imaging conditions (i.e. how many failures to allow before processing of fit candidates is stopped).

The analysis results are then reported in a summary table:

| Field     | Description   |
|-----------|---|
| FRAMES    | The number of frames in the simulated image   |
| W         | The width of the simulated image minus the analysis border  |
| H         | The height of the simulated image minus the analysis border                                       |
| MOLECULES | The number of molecules that occur within the bounds of the analysis border                       |
| DENSITY   | The molecule density within the analysis region   |
| N         | The average number of photons per localisation  |
| S         | The standard deviation of the Gaussian profile that matches the PSF                               |
| A         | The pixel size  |
| DEPTH     | The z-depth of the localisations  |
| FIXED     | True if the simulation used a fixed depth   |
| GAIN      | The total gain of the simulation  |
| READNOISE | The read noise of the simulation  |
| B         | The background number of photons  |
| B2        | The noise per pixel. This is a combination of the read noise and the background number of photons |
| SNR       | The signal to noise ratio ( $N/\sqrt{b2}$ )   |



| Field       | Description  |
|-------------|--|
| S (PX)      | The standard deviation of the Gaussian profile that matches the PSF in pixels  |
| TYPE        | The type of filter   |
| SEARCH      | The search width   |
| BORDER      | The border   |
| WIDTH       | The effective width of the filter, i.e. the number of pixels around a point that are used in the filter. The actual region width is $2w+1$ . This allows comparison of the size of different filters                               |
| FILTER      | The name of the first filter   |
| PARAM       | The parameter of the first filter  |
| DESCRIPTION | The full description of the filter. For a DIFFERENCE or JURY filter the full set of filters will be listed   |
| A.BORDER    | The analysis border  |
| MULTI       | True if multiple matches were allowed  |
| RANKED      | True if matching was performed using the candidate in their ranked order, i.e. higher ranked candidates are matched to true results first. The default is to use nearest neighbour matching, irrespective of the candidate ranking |
| LOWER D     | The lower match distance   |
| D           | The match distance   |
| LOWER SF    | The lower signal factor  |
| SF          | The signal factor  |
| SLOPE       | The slope of the regression between the candidate intensity and the intensity of matching localisations  |
| TP          | The overall number of true positives   |
| FP          | The overall number of false positives  |
| RECALL      | The overall recall   |
| PRECISION   | The overall precision  |

| Field     | Description  |
|-----------|--|
| JACCARD   | The overall Jaccard  |
| R         | The overall correlation between the candidate intensity and the intensity of matching localisations  |
| TP        | The number of true positives at the configured fraction of the maximum recall  |
| FP        | The number of false positives at the configured fraction of the maximum recall   |
| RECALL    | The recall at the configured fraction of the maximum recall  |
| PRECISION | The precision at the configured fraction of the maximum recall   |
| JACCARD   | The Jaccard at the configured fraction of the maximum recall   |
| R         | The overall correlation between the candidate intensity and the intensity of matching localisations at the configured fraction of the maximum recall |
| TP        | The number of true positives at the maximum Jaccard score  |
| FP        | The number of false positives at the maximum Jaccard score   |
| RECALL    | The recall at the maximum Jaccard score  |
| PRECISION | The precision at the maximum Jaccard score   |
| JACCARD   | The maximum Jaccard score  |
| R         | The overall correlation between the candidate intensity and the intensity of matching localisations at the maximum Jaccard score                     |
| TIME      | The total run-time for filtering the image and ranking the candidates  |
| AUC       | The area under the precision-recall curve  |
| AUC2      | The area under the adjusted precision-recall curve. The adjustment is made by using the highest precision at that recall or above                    |
| FAIL80    | The failure count that must be allowed to achieve 80% of the maximum true positive count   |
| FAIL90    | The failure count that must be allowed to achieve 90% of the maximum true positive count   |

| Field   | Description   |
|---------|---|
| FAIL95  | The failure count that must be allowed to achieve 95% of the maximum true positive count  |
| FAIL99  | The failure count that must be allowed to achieve 99% of the maximum true positive count  |
| FAIL100 | The failure count that must be allowed to achieve 100% of the maximum true positive count |

## 9.12 *Fit Spot Data*

Fits all the candidate spots identified by the `FILTER SPOT DATA` plugin. This plugin requires the results generated by the `FILTER SPOT DATA` plugin. If these results are not available an error will be displayed.

### 9.12.1 *Analysis*

The `FIT SPOT DATA` plugin fits each candidate spot identified in the benchmark image by the `FILTER SPOT DATA` plugin. The spot candidates are identified in each frame in the image and ranked by the filter, for example by estimated intensity. However there will be many candidates that are not valid spots since the `CREATE SIMPLE DATA` or `CREATE SPOT DATA` plugin generates an image with noise which may be identified as a spot candidate. Thus it is usually possible to stop processing candidates when a successive number of candidates fail. This is the method employed by the main `PEAK FIT` plugin when processing single-molecule localisation images.

For the purpose of benchmarking it is possible to speed up processing by ignoring many of the candidates. Since the spots were simulated it is known which candidates are correct (positives) and which are incorrect (negatives). The `FIT SPOT DATA` plugin allows the user to specify the fraction of positives in each frame that will be processed. This sets a target limit for the positives. When this target has been reached the plugin will continue processing candidates until: (1) a set fraction of the total number of candidates processed are negatives; and (2) a minimum number of negatives after the positive target have been processed. The rest of the candidates are then ignored and marked as not-fitted.

The algorithm used to fit the spots can be configured in the plugin options. However all fit results are accepted as long as the fitted signal is above zero, the coordinates are within the fit region and the fitted width is within a factor of 5 of the initial estimate. There is no further filtering performed by the plugin on the fit results. This provides an upper limit for the recall that is possible using this fitting configuration. Filtering of the results using limits on the signal, peak width, precision, etc. is done using the `BENCHMARK FILTER ANALYSIS` plugin.

When all the fitting has been done the fitted coordinates and any candidates coordinates that were not fitted or failed to fit are compared with the actual coordinates that were simulated. The two sets of coordinates are assigned as matches if they are within a configured distance. In addition the fitted candidates can be set as matches if they are within a factor of the true signal for the localisation. The relative signal factor is simply the fitted signal divided by the true signal. The relative signal factor (rsf) above or below the true signal (which has 1 for a perfect match) is adjusted so that the new signal factor (sf)

score is 0 for a perfect match:

$$sf = (rsf < 1) ? 1 - \frac{1}{rsf} : rsf - 1$$

The matches are then assigned a score. The score is created using a ramped function between the LOWER DISTANCE and the MATCH DISTANCE. Any distance below the LOWER DISTANCE is 1. Anything above the MATCH DISTANCE is 0. In between uses a ramp to set the score. The ramp is not linear but uses a cosine function. This smooths the transition at the ends of the range to make selection of the end points for assessment less critical; the ramp is linear in the exact centre of the range. Thus the ideal end points should be above and below the ideal scoring threshold. If using a signal factor a similar ramp is applied to produce a signal score and the final score computed as a product of the two. The match score is then used to accumulate a score for how accurate the fitting was performed.

If the match is between a fitted candidate and a true localisation then the score is a true positive. When using a ramped distance function to create the score, the remaining score is calculated as 1 – score and assigned as false positive.

The TP and FP totals thus represent the score that can be achieved for a perfect results filter that is able to correctly allow any fit results that are within the match distance, and optionally signal factor, of a true localisation and reject all other fit results. If a candidate fails to be fitted but is close to a true localisation it will not be included in the TP/FP assessment scores. However the original number of localisations that were simulated is known and this can be used to produce a False Negative (FN) score as the number of localisations minus the TP, i.e. how many spots were missed.

Note that the use of a ramped score function based on distance and signal allows the comparison of scores between different fitting algorithms, since some algorithms may fit the spots closer to the true localisation. Also note that if it is not clear at what level to set the scoring thresholds then using a ramped distance score will produce the same results as repeating the analysis with multiple thresholds and averaging the score (with the same weighting applied by the ramp).

### 9.12.2 Parameters

The following parameters can be configured:

| Parameter                          | Description  |
|------------------------------------|--|
| FRACTION POSITIVES                 | Set the limit (in percent) on the number of filter candidates that match true localisations that must be processed   |
| FRACTION NEGATIVES AFTER POSITIVES | After the positive target has been reached, set the limit (in percent) on the fraction of filter candidates that must be negatives (i.e. how many of the candidates for fitting should be incorrect) |
| MIN NEGATIVES AFTER POSITIVES      | After the positive target has been reached, set a minimum number of filter candidates that must be negatives (i.e. how many extra incorrect candidates should be included in the fitting process)    |

| Parameter          | Description   |
|--------------------|---|
| MATCH DISTANCE     | The distance limit defining the minimum score for a match between a fitted localisation and the true localisation. The distance is expressed relative to the PSF width used to generate the data  |
| LOWER DISTANCE     | <p>The distance limit defining the maximum score for a match between a fitted localisation and the true localisation. The distance is expressed relative to the PSF width used to generate the data.</p> <p>Set to the same as the MATCH DISTANCE to ignore the ramped scoring function</p>   |
| MATCH SIGNAL       | <p>Define the limit for the difference between the fitted signal and the actual signal for a match. A value of (N-1) indicates the fit is allowed to be N-fold different, i.e. use 2 for a 3-fold difference.</p> <p>Set to zero to ignore</p>  |
| LOWER SIGNAL       | <p>The limit defining the maximum score for a match between a fitted signal and the true signal.</p> <p>Set to the same as the MATCH SIGNAL to ignore the ramped scoring function</p>   |
| INITIAL STDDEV     | The initial 2D Gaussian standard deviation for fitting. The width is expressed in pixels. By default it is set using the configured width of the PSF used to generate the data and should not need adjusting (unless it is intended to benchmark an incorrectly calibrated fitting algorithm) |
| FITTING WIDTH      | Define the size of the region around a candidate to use for fitting. The region size in pixels is set using the FITTING WIDTH multiplied by the INITIAL STDDEV  |
| FIT SOLVER         | <p>Define the solver used for fitting. Depending on the chosen solver a second dialog box will be presented to allow further configuration.</p> <p>See section 5.2.4: Fitting Parameters</p>  |
| FIT FUNCTION       | <p>The fit function.</p> <p>See section 5.2.4: Fitting Parameters</p>   |
| INCLUDE NEIGHBOURS | See section 5.2.5: Multiple Peak Fitting Parameters   |
| NEIGHBOUR HEIGHT   | See section 5.2.5: Multiple Peak Fitting Parameters   |
| DUPLICATE DISTANCE | See section 5.2.5: Multiple Peak Fitting Parameters   |

| Parameter              | Description  |
|------------------------|--|
| SHOW SCORE HISTOGRAMS  | Show histograms and cumulation histograms of the metrics used to assess fit results: Signal, SNR, MinWidth, MaxWidth, XY Shift, Euclidian Shift, and Precision. Histograms are also shown of the number of iterations and function evaluations used by the fitting algorithm   |
| SHOW CORRELATION       | Show a plot of the correlation between the candidate intensity against the true spot intensity for all matches, as a function of the spot candidate rank   |
| PLOT RANK BY INTENSITY | The default x-axis rank when plotting the correlation is an integer series starting from 1.<br><br>Set this option to true to show the ranking using the spot intensity relative to the intensity range of all the spots. Candidates are ordered by intensity so the order will be the same but the relative positions will change |
| SAVE FILTER RANGE      | Save the estimated ranges for each spot metric as a filter set file that can be used in the <b>BENCHMARK FILTER ANALYSIS</b> plugin.<br><br>Note: The range increment in the filter set file is set to 1/10 of the range and certain filters are disabled as they are similar (e.g. Shift and EShift)                              |

### 9.12.3 Results

After spot candidates have been fitted and scored using their distance to the true localisations the results are summarised.

For all the fitted results that were within the **MATCH DISTANCE** of a true localisation, a histogram is computed of the distances, the z-depth of the original localisations and the computed precision of the fit using the Mortensen formula. This allows a visualisation of how close the fitted results are to the actual localisations and also of the depth-of-field of the localisation algorithm.

Note that the Match Depth histogram will not be useful if the data were originally simulated with a fixed depth or the PSF does not vary with z-depth. If results were simulated using a fixed photon level then the standard deviation of the match distance should be approximately the same as the average of the calculated precision (since the precision calculation represents the standard deviation of the distance from the true location assuming fitting a 2D Gaussian PSF with Poisson noise).

A scatter plot of the spot intensity against the true spot intensity is shown for all matches. The correlation and slope of the line (assuming an intercept at 0,0) is shown on the plot. An ideal fit algorithm should produce a slope of 1.

For all of the candidates that were fit the system compiles histograms of each of the common metrics used to assess the fit of the 2D Gaussian to the data (Signal, SNR, MinWidth, MaxWidth, XY Shift, Euclidian Shift, and Precision). Three histograms are

computed for: all the fitting results; only the matches; and the non-matched results. The histogram are then used to produce an estimate for the range that should be used for each metric as a filter to separate good fits from bad fits. This is done by comparing the values in the matches histogram with the values in the non-matches histogram, for example using the largest separation between the cumulative histograms for each. This may for example produce an estimate that a SNR filter should be in the range of 25 to 300. These ranges are reported to the results table and can optionally be saved to a file for use in the **BENCHMARK FILTER ANALYSIS** plugin.

The results of match scoring are used computed the match statistics Recall, Precision, Jaccard and F1-score. See Appendix B for more details. The first set of statistics are for the raw candidates (before fitting). These show how effective the filter was at identifying candidates that were processed and so set an upper limit on the performance of the fitted results. The second set summarise the performance of the fitted results. Note that the table does not show the False Negative score since this is equal to the number of simulated molecules minus the True Positives.

The analysis results are then reported in a summary table

| Field     | Description   |
|-----------|---|
| FRAMES    | The number of frames in the simulated image   |
| W         | The width of the simulated image minus the analysis border  |
| H         | The height of the simulated image minus the analysis border                                       |
| MOLECULES | The number of molecules that occur within the bounds of the analysis border                       |
| DENSITY   | The molecule density within the analysis region   |
| N         | The average number of photons per localisation  |
| S         | The standard deviation of the Gaussian profile that matches the PSF                               |
| A         | The pixel size  |
| DEPTH     | The z-depth of the localisations  |
| FIXED     | True if the simulation used a fixed depth   |
| GAIN      | The total gain of the simulation  |
| READNOISE | The read noise of the simulation  |
| B         | The background number of photons  |
| B2        | The noise per pixel. This is a combination of the read noise and the background number of photons |

| Field   | Description   |
|---------|---|
| SNR     | The signal to noise ratio ( $N/\sqrt{b^2}$ )  |
| S (PX)  | The standard deviation of the Gaussian profile that matches the PSF in pixels   |
| FILTER  | The full description of the filter used to identify the fitting candidates. For a DIFFERENCE or JURY filter the full set of filters will be listed  |
| SPOTS   | The number of filter candidates   |
| NP      | The number of spot filter candidates that identify a true localisation (positives).<br><br>Note: This result is computed with the distance thresholds set in the FILTER SPOT DATA plugin  |
| NN      | The number of spot filter candidates that do not identify a true localisation (negatives)   |
| FP      | The number of spot filter candidates that identify a true localisation (positives). This is a fractional score that uses the partial matching score system with lower and upper distance thresholds.<br><br>Note: This result is computed with the distance thresholds set in the FILTER SPOT DATA plugin |
| FN      | The number of spot filter candidates that do not identify a true localisation (positives). This is the number of spot candidates minus FP   |
| SOLVER  | The fit solver used for fitting   |
| FITTING | The fit window size used for fitting  |
| % NP    | The fraction of positive candidates that were processed   |
| % NN    | The fraction of negative candidates that were processed   |
| TOTAL   | The number of candidates that were processed  |
| cTP     | The number of candidates that match a true localisation (Candidate True Positives)  |
| cFP     | The number of candidates that do not match a true localisation (Candidate False Positives)  |
| cRECALL | The recall of the candidates  |



| Field        | Description   |
|--------------|---|
| cPRECISION   | The precision of the candidates   |
| cJACCARD     | The Jaccard of the candidates   |
| cF1          | The F1-score of the candidates  |
| FAIL cTP     | The number of candidates that match a true localisation where fitting failed, i.e. the algorithm did not return a result  |
| FAIL cFP     | The number of candidates that do not match a true localisation where fitting failed, i.e. the algorithm did not return a result   |
| TP           | The sum of the match score for fit results that can be matched to a localisation  |
| FP           | The sum of the remaining match score for fit results that can be matched to a localisation.<br><br>This will be zero if not using a ramped match score with MATCH DISTANCE above LOWER DISTANCE |
| RECALL       | The recall of the fitted candidates   |
| PRECISION    | The precision of the fitted candidates  |
| F1           | The F1-score of the fitted candidates   |
| JACCARD      | The Jaccard of the fitted candidates  |
| pF1          | The F1-score of the fitted candidates assuming perfect precision (i.e. all false positives are removed). Recall is unchanged  |
| pJACCARD     | The Jaccard of the fitted candidates assuming perfect precision (i.e. all false positives are removed). Recall is unchanged   |
| MED.DISTANCE | The median of the histogram of the distance between fitted candidates and the localisations   |
| MED.DEPTH    | The median of the histogram of the z-depth of localisations that were fitted  |
| CORRELATION  | The correlation between the fitted signal and the actual signal for match results   |
| RANKED       | The Spearman's rank correlation between the fitted signal and the actual signal for match results   |
| SLOPE        | The slope of the regression between the fitted signal and the actual signal for match results (assuming an intercept at 0,0)  |

| Field       | Description   |
|-------------|---|
| SIGNAL      | The lower and upper range estimate for a signal filter  |
| SNR         | The lower and upper range estimate for a SNR filter*  |
| MINWIDTH    | The lower and upper range estimate for a minimum width filter*  |
| MAXWIDTH    | The lower and upper range estimate for a maximum width filter*  |
| SHIFT       | The lower and upper range estimate for an absolute max X or Y shift filter*   |
| EShift      | The lower and upper range estimate for an Euclidian shift filter*   |
| PRECISION   | The lower and upper range estimate for a precision filter*  |
| ITERATIONS  | The lower and upper range of iterations used for fitting results that match (uses 1% and 99.9% percentile values)           |
| EVALUATIONS | The lower and upper range of function evaluations used for fitting results that match (uses 1% and 99.9% percentile values) |
| RUN TIME    | The length of time taken to perform fitting   |

\* The lower and upper bounds are set using a variety of measures suitable for the metric including percentile values of the data range and the maximum difference between the cumulative histograms of matched and non-matched fit results. Full details can be found in the source code or on request from Alex Herbert. The ranges are only a suggested start point for generation of a filter to classify fit results. The ranges can be tested on the fit results using the **BENCHMARK FILTER ANALYSIS** plugin to find an optimum filter for the simulated data.

## 9.13 Benchmark Filter Analysis

Run different filtering methods on a set of benchmark fitting results produced by **FIT SPOT DATA** outputting performance statistics on the success of the filter. If these results are not available an error will be displayed when running the plugin.

The **BENCHMARK FILTER ANALYSIS** plugin is designed to test the results filtering available in the **PEAK FIT** plugin. The principle is that simulated localisations are identified as candidates for fitting and then fitted using the same routines available in **PEAK FIT**. This is done using the **FILTER SPOT DATA** and **FIT SPOT DATA** plugins. The results can then be subjected to different filters to determine the best filter.

### 9.13.1 Input Filters

The plugin is able to process thousands of filters by loading the filters from a file. The file describing the filters can be created using the **CREATE FILTERS** plugin. When all the parameters for the plugin are configured the plugin prompts the user for a filter file. Loading filters may take a long time so if the filename selected for the filters is the same as

the last set of loaded filters the plugin will ask if you would like to re-use the filters that were previously loaded.

Filters are grouped into sets. Each set is processed separately. This allows results to be shown per filter and then as a summary of the best results per filter set. This allows different types of filters to be compared (e.g. SNR, Precision, etc) in a summary table.

### 9.13.1.1 Expanding Filter Sets

Note that the plugin will detect if a filter set only contains 3 filters and determine if it can be expanded. The criteria for expansion are that the second filter has a value for each parameter equal or above the first filter. The first filter then forms the minimum value and the second filter the maximum value. The third filter must then have a value that is positive for each parameter where the second filter value was above the first filter value. Note that if the increment is infinity then it is ignored. The third filter then forms the increment for the parameters. This means that it is possible to create a series of values from minimum to maximum using the increment. Note that the increment does not have to be an exact factor of the range. The value is just incremented from the minimum until the maximum is reached (or exceeded).

Note that a filter set file can be created using suitable ranges for the current fit results by the FIT SPOT DATA plugin. (Use the SAVE FILTER RANGE option.) To avoid testing filters that assess similar properties the FIT SPOT DATA plugin disable certain filters. Currently EShift and Signal filtering are disabled as they are similar to Shift and SNR. They can be reenabled by changing the third filter value from INFINITY to a suitable increment. If a filter set file is saved using FIT SPOT DATA then the file will be pre-selected in the dialog for convenience.

If the filter set can be expanded the plugin will compute the number of combinations that will be created after expansion. It will then ask the user if they would like to expand the filters.

Expanding filters is much faster than reading a large number of filters in from a file and so is the preferred method of loading a large evenly spaced filter set. Any filter set that has been expanded is also available for a step search optimisation of the parameters (see below).

Note that advanced filters using AND or OR filters can be constructed using expansion. The user should note that the parameters for the combined filters are represented as a single linear array, for example:

```
<list>
<FilterSet name="Multi2">
  <filters>
    <AndFilter><filter1 class="SNRFilter" snr="10"/><filter2 class="WidthFilter"
width="1.5"/></AndFilter>
    <AndFilter><filter1 class="SNRFilter" snr="20"/><filter2 class="WidthFilter"
width="2.5"/></AndFilter>
    <AndFilter><filter1 class="SNRFilter" snr="1"/><filter2 class="WidthFilter"
width="0.1"/></AndFilter>
  </filters>
</FilterSet>
</list>
```

would expand into a SNR and Width filter using a range of 10-20 for SNR and 1.5 to 2.5 for Width.

The filtered results are then scored. The scoring is designed to find the best fitter & filter combination for the given spot candidates. The ideal combination would correctly fit & pick all the candidate positions that are close to a localisation and reject all other candidates.

Candidates

Actual matches

**FN**

**TP**

**FN**

Fitted Spots

**TN**

Positives

Negatives

|                |  |
|----------------|--|
| Candidates     | All the spot candidates  |
| Actual matches | Any spot candidate or fitted spot candidate that matches a localisation  |
| Fitted spots   | Any spot candidate that was successfully fitted  |
| Positives      | Any fitted spot that was accepted by the filter  |
| Negatives      | Any fitted spot that was rejected by the filter  |
| TP             | True Positive: A spot candidate that was fitted and matches a localisation and is accepted   |
| FP             | False Positive: A spot candidate that was fitted but does not match a localisation and is accepted   |
| FN             | False Negative: A spot candidate that failed to be fitted but matches a localisation; or a spot candidate that was fitted and matches a localisation |

and is rejected

TN True Negative: A spot candidate that failed to be fitted and does not match a localisation; or a spot candidate that was fitted and does not match a localisation and is rejected

Classically a match is assigned if a predicted result and a localisation are within a distance threshold. This makes the choice of distance threshold critical. It also means that methods that get very close to the answer are not scored better than methods that get just close enough to the answer. This can be overcome by repeating the analysis multiple times with different distance thresholds and averaging the scores. An alternative is to use a ramped scoring function where the degree of match can be varied from 0 to 1. When using ramped scoring functions the fractional allocation of scores using the above scheme is performed, i.e. candidates are treated as if they both match and unmatch and the scores accumulated using fractional counts. This results in an equivalent to multiple analysis using different thresholds and averaging of the scores, but it can be performed in one iteration.

The choice of the distance thresholds for benchmarking a microscope setup can be made using the wavelength of light ( $\lambda$ ) and the expected number of photons. The upper threshold can be set using the Abbe limit:

$$Resolution = \frac{\lambda}{2 NA}$$

where NA is the numerical aperture of the microscope [Abbe, 1873]. Any match below this distance is closer than the standard resolution of an optical microscope and so can be classed as a super-resolution match. The lower threshold can be set using the Mortensen formula for the precision of fitting a 2D Gaussian to 2D Gaussian data corrupted by Poisson noise (see Appendix A : Localisation Precision). Although only an approximation of fitting to a microscope PSF the formula provides an estimate of the limit of super-resolution fitting. Any fitting algorithm will not be expected to perform matches consistently below this limit. In practice though it is usually fair to set the lower distance as 33-50% of the Abbe limit.

As well as matching the localisation position it is possible to assign matches using the fitted signal. The signal-factor is computed within the FIT SPOT DATA plugin and is a measure of how far the fitted signal was from the true number of photons. Matches can be rejected if they are above a threshold and, as for the distance match, a ramped score is available using a ramped scoring function. In the case that ramped scoring is used for both distance and signal-factor then the final match score is the product of the two ramped scores. The remaining unmatched score is set so the total for the result is 1.

The totals TP+FP+TN+FN must equal the number of spot candidates. This allows different fitting methods to be compared since the total number of candidates is the same. The TP, FP, TN and FN totals can be used to compute scores as detailed in Appendix B : Comparison Metrics.

As an alternative scoring system, different fitting methods can be compared using the same TP value but calculating FN = localisations - TP and FP as Positives - TP. In this scheme FN represents any original spots that were missed and FP represents the number of accepted fits that do not match an original spot. This creates a score against the original simulated localisations using everything that was passed through the filter (Positives). The results table indicates this by prefixing the score column titles with o for Original. This

score is comparable when a different spot candidate filter has been used and the total number of candidates is different, e.g. Mean filtering vs. Gaussian filtering. Note that in this scheme there is no TN total and so the number of comparison metrics that can be computed are reduced to recall, precision, Jaccard and F1-score. The original score metrics are used by default for selection and ranking.

### 9.13.2.1 Ranking filters

When all the scores have been computed for the filters in a filter set, the filters are ranked. Ranking is performed using two chosen scoring metrics. The first metric is chosen as a minimum limit that must be achieved; this is the `CRITERIA` metric. The second metric is chosen to rank all the filters that pass the criteria; this is the `SCORE` metric. If no filters pass the criteria then a warning is written to the `IMAGEJ` log window.

This ranking system allows filters to be restricted to those that function at a minimum desired level and then ranked. For example assessing all filters that achieve 95% precision and then ranking by recall would pick the best filter for high confidence in the results, and assessing all filters that achieve 80% recall and then ranking by precision would pick the best filter for returning a high number of localisations but potentially also returning many false positives. Note that some scores are better as they get lower. In this case the plugin reverses the ordering to pick the best filter.

The original score metrics are used by default for selection and ranking. The precision is used to set the `CRITERIA LIMIT`. The results are then ranked using the Jaccard score. Using the Jaccard ensures that the best possible filter is chosen since this score maximises the overlap between predicted and actual results. Note that when ranking filters if the score is the same then the rank is determined by the criteria metric.

The use of the `CRITERIA` filter can be disabled by setting the `CRITERIA LIMIT` to a value achievable by any filter, e.g. Precision = 0.

The best filter from each filter set is recorded to the summary table.

### 9.13.2.2 Sensitivity Analysis

The sensitivity analysis aims to show how much the scores will change when the filter parameters are adjusted. An ideal filter would be one with low sensitivity, i.e. changing the parameters away from the optimal does not alter performance.

Sensitivity can be calculated for the best filter from each filter set. This is done by altering the parameters of the filter by a small change (delta) and recomputing the scores. This can be used to express the relative change in the score with a change in the parameters, i.e. the gradient. The gradient of each parameter is reported; those with lower gradients are more robust and those with higher gradients are the ones that cannot be varied very much for optimal performance of the filter.

### 9.13.2.3 Optimisation

Searching for the best parameters for a filter is an optimisation problem. The more parameters there are in the filter the more combinations are possible. It may not be feasible to enumerate all the possible parameter combinations to find the best parameters. The plugin offers two methods for searching for the optimal parameters: evolution using a genetic algorithm; and a step search. Once the optimal parameters are found for the filter

set then the plugin produces the summary results as per normal.

### Evolution using a Genetic Algorithm

All the filters in a filter set are used to create a population of filters. The parameters are listed in order to create a genome for each individual. The fitness of each individual is the score that is achieved using the ranking metric. If the criteria is not reached then the fitness is set to zero. Optionally this can be relaxed to allow all filters achieving the criteria to be ranked ahead of all filters not achieving the criteria.

After computing the fitness of individuals in a population, the population is allowed to change. The population is first reduced to a subset of the individuals using a selection process; the best scoring individual is always kept at each iteration. The subset forms a new population that is expanded by breeding. Pairs are selected for crossover and then new individuals are created by crossing their parent parameter values at random points along the genome. These new individuals may also mutate to change parameters in their genome. When enough new individuals have been created the fitness is evaluated again and the process repeated until no improvement can be made. Note that the individuals selected for the subset are not changed and only new individuals can mutate. This means that the population should not get worse if the top fraction is selected each iteration. If selection is by sampling from the population then it is possible the population can get worse, but the top individual is always retained.

Due to the use of mutation the genetic algorithm is able to produce parameter values that were not in the original filter set.

### Step Search

If a filter set has been expanded using 3 filters defining minimum, maximum and increment parameters then the plugin can perform a step search using a fraction of the increment as the step size.

All of the filters in a filter set are evaluated and then ranked. A search is then performed from the best filter using a fraction of the increment. New filters are created that use the step increment to push the parameters to new values outside the original range. The new filters are evaluated and this process is repeated until no improvement in score is made.

The `STEP_SEARCH` option is a fast method for taking an initial parameter range and producing an optimal set of parameters that may not be within that range. If the range and increment for the parameters is well chosen then the optimal parameters can be found without computing the results for a large set of filters.

#### 9.13.3 Parameters

When the plugin is run a dialog is present allowing parameters to be configured.

The dialog message shows a summary of the results computed by `FIT_SPOT_DATA` that will be analysed. The number of results is shown along with the number of true positives within the results if they are scored with the current values for the match distances. If the match distances are altered then the number of true positives will be recomputed when the plugin is run but will not update in the plugin message until it is next displayed.

The expected signal and localisation precision of the simulation localisations is computed

using the formulas of Thompson (for signal) and Mortensen (for localisation), see Appendix A : Localisation Precision for more details. If analysis has been previously run on this data then the score for the best filter result held in memory is shown. (The results in memory functionality is used for choosing the optimum fail count settings for a filter set.)

Note that the match distance parameters are expressed relative to those used in the `FIT SPOT DATA` plugin. This is because the matches are assigned within that program and cannot be recomputed. A lower distance is allowed as it will change a match to a non-match. However higher values are not allowed as this require recomputation of all match distances. The distances that were used by the `FIT SPOT DATA` plugin are shown in a message above the section where the distances are configured to remind the user of the values. Re-run the `FIT SPOT DATA` plugin if you wish to use higher values.

The following parameters can be adjusted:

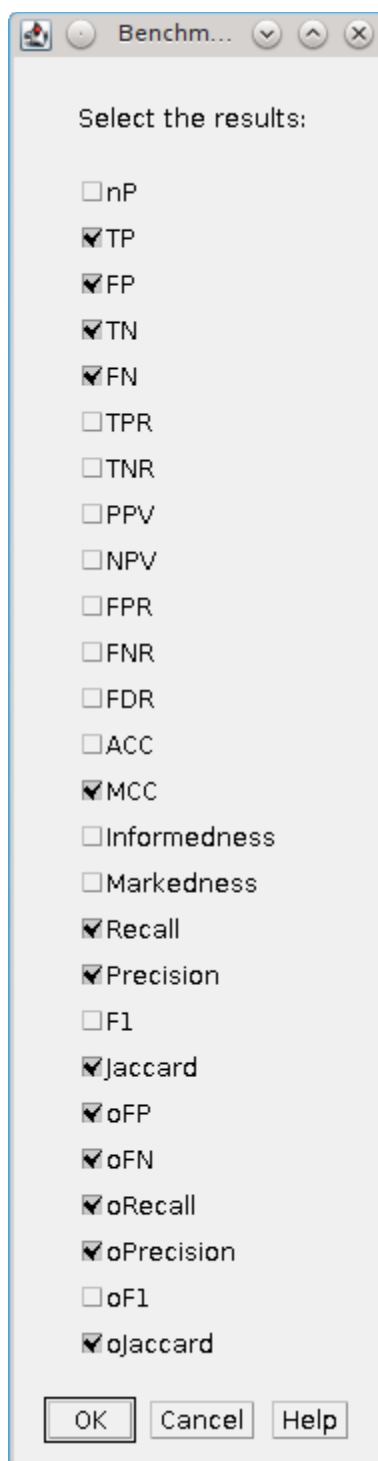
| Parameter        | Description   |
|------------------|---|
| FAIL COUNT       | The number of failures to accept before rejecting the remaining results from the current frame  |
| FAIL COUNT RANGE | If specified, the filter will be evaluated using each fail count from <code>FAIL COUNT</code> to <code>FAIL COUNT + FAIL COUNT RANGE</code> .<br><br>This setting significantly slows down computation as the filters must be evaluated multiple times and results pre-processing is not performed  |
| RESET            | Set to true to clear the best filter scores from memory. The memory is also cleared if the fit results have changed, or the ranking and scoring thresholds are changed  |
| RANK BY SIGNAL   | By default results are ranked using the order determined by the spot filter that identified the candidates.<br><br>Choose this option to rerank the results using the fitted signal. Note that ranking spot candidates by fitted signal is not possible in the <code>PEAK FIT</code> plugin since fitting results have not been computed. This option allows the user to test if fitting every possible candidate spot, re-ranking and then filtering would improve results |
| SHOW TABLE       | Show a result table with the scores of each filter  |
| SHOW SUMMARY     | Show a result table with the best filter(s) from each filter set  |
| CLEAR TABLES     | Remove any results in results tables that are already open  |
| SUMMARY TOP N    | Show the top N filters from each filter set in the summary table. Set to zero to disable (show only the top 1 filter)   |



| Parameter             | Description  |
|-----------------------|--|
| SUMMARY DEPTH         | Specify the depth of localisations used for the depth recall analysis in the summary table, i.e. the recall of localisations within the specified depth-of-field   |
| PLOT TOP N            | <p>Show a plot of the performance score against the parameter value for each filter in the top N scoring filter sets. This is useful for filters that have one main parameter, e.g. signal-to-noise or localisation precision. However combination filters that test many fit parameters together are only partially supported. In this case the main parameter value for the filter will determine the x-axis value.</p> <p>This option is not available when an optimisation search has been performed on the filter set</p> |
| SAVE BEST FILTER      | Select this to save the best filter from all filter sets to file. The user will be prompted for a filename when the analysis is complete   |
| SAVE TEMPLATE         | Set to true to save a template containing the fit and filter settings from the best filter to a template file. Template files can be loaded by the TEMPLATE CONFIGURATION plugin for use in PEAK FIT   |
| CALCULATE SENSITIVITY | Select this to perform sensitivity analysis on the top filter from each filter set   |
| DELTA                 | The relative change in the parameter value to use to calculate the sensitivity   |
| CRITERIA              | The metric used for the minimum criteria   |
| CRITERIA LIMIT        | The value for the minimum criteria   |
| SCORE                 | The metric used to rank all filters that reach the minimum criteria  |
| UPPER MATCH DISTANCE  | The distance limit defining the minimum score (0) for a match between a fitted localisation and the true localisation. The distance is expressed relative to the match distance used with the FIT SPOT DATA plugin (since the matches of results to localisations are computed within that program)  |

| Parameter              | Description   |
|------------------------|---|
| PARTIAL MATCH DISTANCE | <p>The distance limit defining the maximum score (1) for a match between a fitted localisation and the true localisation. The value is expressed relative to the match distance used with the FIT SPOT DATA plugin (since the matches of results to localisations are computed within that program).</p> <p>Set to the same as the UPPER MATCH DISTANCE to ignore the ramped scoring function. Otherwise matches at a distance between PARTIAL and UPPER MATCH DISTANCE will have a score between 0 and 1</p> |
| UPPER SIGNAL FACTOR    | <p>The signal factor defining the minimum score for a match between a fitted localisation and the true localisation. The value is expressed relative to the signal factor used with the FIT SPOT DATA plugin (since the matches of results to localisations are computed within that program).</p>  |
| PARTIAL SIGNAL FACTOR  | <p>The signal factor defining the maximum score for a match between a fitted localisation and the true localisation. The value is expressed relative to the signal factor used with the FIT SPOT DATA plugin (since the matches of results to localisations are computed within that program).</p> <p>Set to the same as the UPPER SIGNAL DISTANCE to ignore the ramped scoring function. Otherwise matches with a signal-factor PARTIAL and UPPER SIGNAL FACTOR will have a score between 0 and 1</p>        |
| DEPTH RECALL ANALYSIS  | <p>Produce a histogram of the recall verses z-depth of the original localisations.</p> <p>This option is only available if the simulation had variable z-depth</p>  |
| SCORE ANALYSIS         | <p>Produce a histogram of the distance and signal factor for all fit matches to the original localisations</p>  |
| EVOLVE                 | <p>Perform evolution optimisation with a genetic algorithm</p>  |
| STEP SEARCH            | <p>Set above zero to do a step search algorithm. The step search is performed using a step size of increment/N, where N is the STEP SEARCH parameter. Note if the step size is 1, a search will only be performed if the best filter is at the edge of the range tested (otherwise each step will move to an already evaluated filter).</p> <p>This can only be performed if the filters were expanded from an input filter set and the EVOLVE option is false</p>  |
| TITLE                  | <p>Add a title for the analysis to the results tables. This can be used when running multiple repeats of the plugin with results from different filters and fitting algorithms</p>  |

Once the main parameters have been chosen a second dialog is presented where the scoring metrics that are recorded in the results table can be specified. This allows the user to remove many of the results from the table if they do not need them to save space:



If the `EVOLVE` option is selected a dialog will be presented for each filter set to be optimised. The parameters are all prefixed with the filter set number. This allows the support for recording the plugin using the `IMAGEJ` recorder for use in the `IMAGEJ` macro language. The following parameters are available:

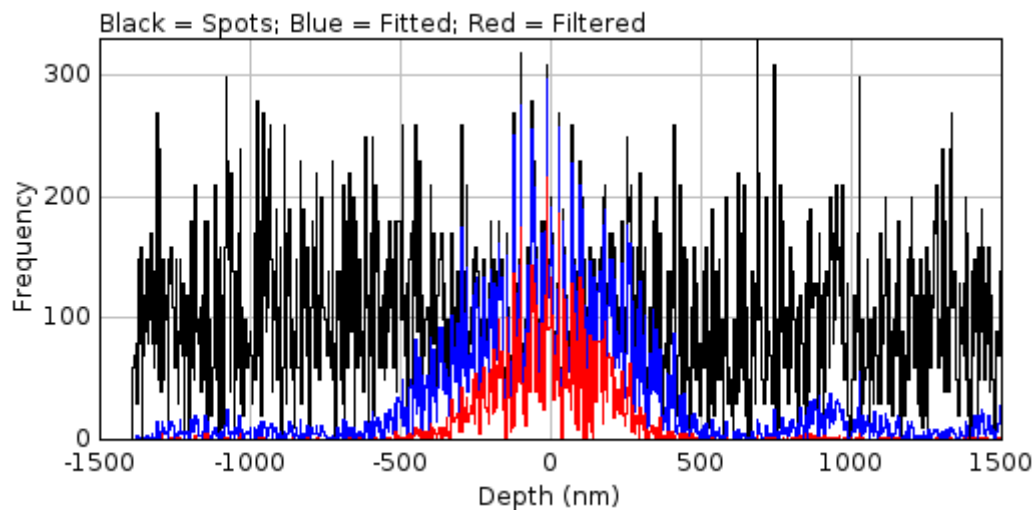
| Parameter          | Description  |
|--------------------|--|
| POPULATION SIZE    | The size of the population to reach when producing new individuals. A higher number will slow down the optimisation but may allow a better solution to be found  |
| FAILURE LIMIT      | The limit on the number of failed attempts to create a new individual before stopping growing the population. Note that new individuals are checked to be unique so it is common in a highly similar population that many crossovers will no create unique combinations  |
| TOLERANCE          | The relative difference below which two fitness scores are equal   |
| CONVERGED COUNT    | The number of iterations with no change in fitness for convergence.<br><br>Warning: If this is set to zero then convergence checking is disabled and the algorithm must be stopped using the IMAGEJ interrupt by pressing the ESCAPE button  |
| MUTATION RATE      | The mean fraction of the genome positions that will mutate. The number of mutations is a Poisson variable sampled using a mean equal to the mutation rate multiplied by the genome length. Note that the same position can mutate multiple times   |
| CROSSOVER RATE     | The mean fraction of the genome positions that will crossover. The number of crossovers is a Poisson variable sampled using a mean equal to the crossover rate multiplied by the genome length. It is not possible to crossover more times than the number of parameters in the genome   |
| MEAN CHILDREN      | When performing crossover between two selected individuals, the number of children will be equal to a Poisson variable sampled using this mean. At least 1 child is always produced  |
| SELECTION FRACTION | At each iteration the population is reduced to a fraction of the target population size. The new size will be at least 2   |
| RAMPED SELECTION   | Select individuals using a weighting so that each individual is weighted according to the rank of the fitness score. This allows unfit individuals to be selected when reducing the population size, albeit with lower probability. When selecting for crossovers the selection is biased towards the highest ranking individuals.<br><br>If not selected the individuals will be selected in order of fitness when reducing the population and randomly from the population when selecting for crossovers |

| Parameter      | Description   |
|----------------|---|
| STRICT FITNESS | <p>Only allow individuals that pass the <code>CRITERIA LIMIT</code> to be selected. This may results in an empty population after selection.</p> <p>If not selected then the fitness score for the individual is:<br/> <math>\text{score metric} + 1 + \text{criteria metric}</math></p> <p>This will work for any criteria metric that is in the range of 0-1 allowing all those individuals that pass the criteria to be ranked above the others</p>  |
| SAVE OPTION    | Allow the final population of filters (after convergence) to be saved to file. This allows restarting the plugin to continue with the same population   |
| PARAM NAME     | <p>The mutation range.</p> <p>The dialog will present a field for each named parameter of the filter. The mutation range is the standard deviation of the Gaussian distribution used to mutate the parameter.</p> <p>Note: If the filter set was created using expansion from min; max; increment values then the range defaults to zero for any parameter that was not expanded. (It is assumed that the user does not want to explore this parameter.) Otherwise the value is a default for the filter parameter adjusted by the <code>DELTA</code> parameter</p> |

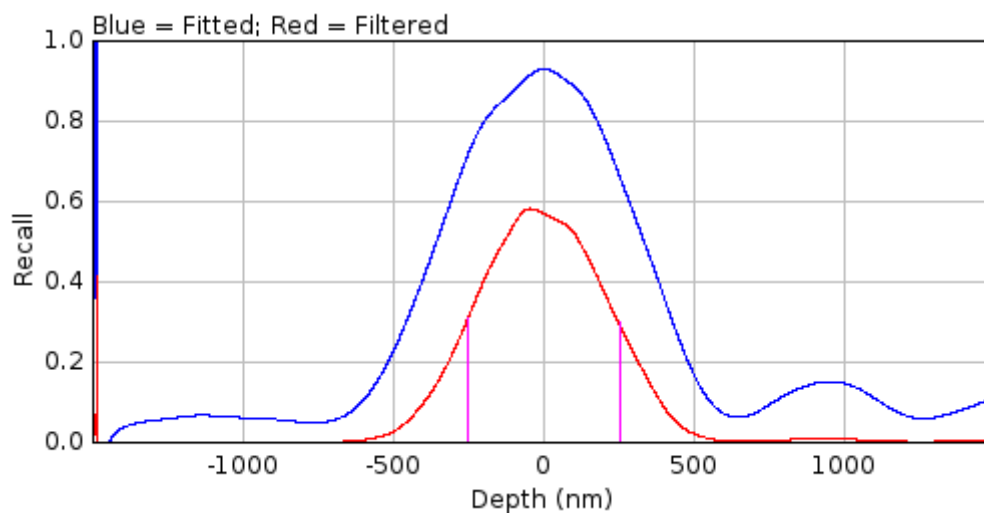
Once the genetic algorithm is started a results table is created named 'Benchmark Filter Analysis Evolution'. This table contains the same columns as the main results table with an extra column for the iteration of the algorithm. The results from the best filter per iteration are added to the table. This table can be used to track the progress of the algorithm. If desired the genetic algorithm can be stopped manually by pressing the `ESCAPE` character.

### 9.13.4 Results

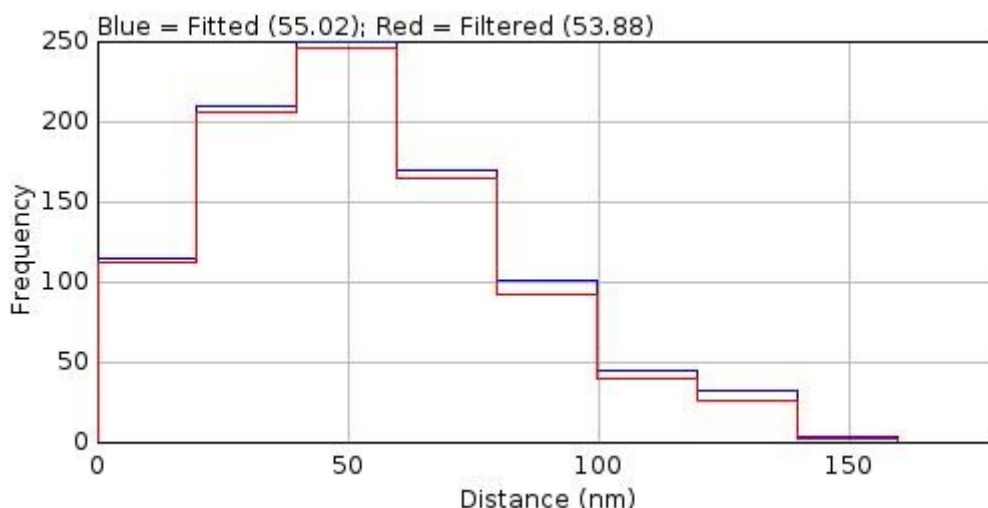
If the `DEPTH RECALL ANALYSIS` option was selected the plugin will compute a histogram of the recall verses z-depth. Three histograms are computed. The first is for all the localisations. This is plotted in black and provides the upper limit for performance. The second is the histogram for all the fitted results that match a localisation. This is plotted in blue and provides the upper limit that can be achieved by fitting. The third is the histogram for all the fitted results that were accepted by the filter. This is plotted in red. An example is shown below:



Since the number of localisations at different depths is highly variable the histograms are smoothed and then normalised by the number of localisations. In the normalised plot only the fitted and filtered lines are shown; the magenta lines represent the depth-of-field specified by the SUMMARY DEPTH field:



If the **SCORE ANALYSIS** option was selected the plugin will compute a histogram of the distance and signal-factor for all matches. Two histograms are computed. The first is for all the fitted results that match a localisation. This is plotted in blue and provides the upper limit that can be achieved by filtering. The second is the histogram for all the fitted results that were accepted by the filter. This is plotted in red. The label shows the mean of each histogram in braces:



If the **SHOW TABLE** or **SHOW SUMMARY** options are selected the plugin will display the results in a table. The following fields are shown in the results tables. Note that for brevity not all the metrics that can be selected by the user are described here. See Appendix B : Comparison Metrics for details of the metrics.

| Field     | Description   |
|-----------|---|
| FRAMES    | The number of frames in the simulated image                                 |
| W         | The width of the simulated image minus the analysis border                  |
| H         | The height of the simulated image minus the analysis border                 |
| MOLECULES | The number of molecules that occur within the bounds of the analysis border |
| DENSITY   | The molecule density within the analysis region                             |
| N         | The average number of photons per localisation                              |
| S         | The standard deviation of the Gaussian profile that matches the PSF         |
| A         | The pixel size  |
| DEPTH     | The z-depth of the localisations  |
| FIXED     | True if the simulation used a fixed depth                                   |

| Field     | Description  |
|-----------|--|
| GAIN      | The total gain of the simulation   |
| READNOISE | The read noise of the simulation   |
| B         | The background number of photons   |
| B2        | The noise per pixel. This is a combination of the read noise and the background number of photons  |
| SNR       | The signal to noise ratio ( $N/\sqrt{B2}$ )  |
| S (PX)    | The standard deviation of the Gaussian profile that matches the PSF in pixels  |
| FILTER    | The full description of the filter used to identify the fitting candidates. For a DIFFERENCE or JURY filter the full set of filters will be listed   |
| SPOTS     | The number of filter candidates  |
| NP        | The number of filter candidates that identify a true localisation (positives).<br>Note: This result is computed with the distance thresholds set in the FILTER SPOT DATA plugin  |
| NN        | The number of filter candidates that do not identify a true localisation (negatives)   |
| FP        | The number of filter candidates that identify a true localisation (positives). This is a fractional score that uses the partial matching score system with lower and upper distance thresholds.<br>Note: This result is computed with the distance thresholds set in the FILTER SPOT DATA plugin |
| FN        | The number of filter candidates that do not identify a true localisation (positives). This is the number of spot candidates minus FP   |
| SOLVER    | The fit solver used for fitting  |
| FITTING   | The fit window size used for fitting   |
| TITLE     | The TITLE field from the plugin dialog is copied here  |
| NAME      | The name of the filter   |



| Field        | Description  |
|--------------|--|
| FAIL         | The fail count. If using a range then the min and maximum of the range will be shown                                     |
| LOWER D      | The lower match distance   |
| UPPER D      | The upper match distance   |
| LOWER FACTOR | The lower match signal-factor  |
| UPPER FACTOR | The upper match signal-factor  |
| TP           | The true positive score  |
| FP           | The false positive score   |
| TN           | The true negative score  |
| FN           | The false negative score   |
| METRICS      | Configured score metrics computed using TP, FP, TN and FN. See Appendix B : Comparison Metrics                           |
| oFP          | The original false positives score. This is equal to the number of results that were accepted by the filter minus the TP |
| oFN          | The original false negatives score. This is equal to the number of localisations that were simulated minus the TP        |
| oMETRICS     | The original score metrics computed using TP, oFP and oFN. See Appendix B : Comparison Metrics                           |

The summary table contains the same fields as the results table. The following additional columns are present:

| Field         | Description  |
|---------------|--|
| DEPTH RECALL  | The recall of localisations that were within the depth specified by the SUMMARY DEPTH parameter. Note that the parameter is halved to set a range for z set above and below zero so that the depth-of-field equals the SUMMARY DEPTH |
| DISTANCE      | The average distance between fitted results and the true localisation  |
| SIGNAL FACTOR | The average signal factor between fitted results and the true localisation   |

| Field    | Description   |
|----------|---|
| AT LIMIT | A flag to indicate if the optimal filter from the filter set was at the limit of the range for any of the expanded parameters. If this is Y (yes) then a better filter may exist if the ranges are changed. The parameter(s) at the edge of the range are recorded in the IMAGEJ log window |

## 9.14 Doublet Analysis

Fits candidate spots identified by the `FILTER SPOT DATA` plugin as a single or double spot (doublet) and scores the results. This plugin requires the results generated by the `FILTER SPOT DATA` plugin. If these results are not available an error will be displayed.

Super-resolution data may contain overlapping localisations. These may be so close they appear as a single spot. The fitting algorithm within the SMLM code is capable of analysing the difference (fit residuals) between the fitted 2D Gaussian and the image data. If the residuals show an assymmetric pattern (e.g. an elliptical shape) then it is possible that the image contains two spots in close proximity that have been identified and fit as a single spot. The software can refit the image using a two peak model. The results are compared to the single peak model and accepted if they pass set criteria. The `DOUBLET ANALYSIS` plugin allows the criteria for choosing doublet fits to be tested.

### 9.14.1 Analysis

The `DOUBLET ANALYSIS` plugin fits a subset of the candidate spots identified in the benchmark image by the `FILTER SPOT DATA` plugin. Localisation are assigned to their nearest spot candidate. Their is no distance threshold but the localisation must be within the fit window that will be used when fitting the candidate, i.e could this localisation be fit by the candidate spot. All candidates that match one or more localisations are processed using the fitting routine since these have the possibility of being a doublet (a match to two localisations). Candidates that do not match any localisations are ignored, simplifying analysis; these are results that should be eliminated as a single fit by a good results filter preventing them from entering doublet fitting.

Optionally any spot candidate that has a neighbour spot candidate in the fit window can be ignored. This is a situation where the fitting algorithm could be set to fit multiple peaks, which is the preferred option to a guess between a single or doublet fit.

The plugins stores the count of the number of actual results that match a spot, and the number of spots with neighbours within the fit region or within the fit region expanded 50%. These can be reported in a results table for each candidate and plotted as summary histograms.

The candidates are then fit as a single and a doublet (irrespective of what the true number of localisations is) and the results stored for analysis. Various metrics are computed including the residual score for the single fit, and the distances from the original fit centre of the doublet centres.

Once all the fitting has been completed the results are compared to the actual localisations and scored. Scoring uses distance and signal factor thresholds with an upper and lower

bounds and a ramped scoring function. This is same as the ramped scoring used in the `BENCHMARK FILTER ANALYSIS` plugin (see 9.13.2).

Results are scored as if all fits were accepted a single fit, or all results accepted as a doublet fit. The results are stored for use in the `DOUBLET FILTER ANALYSIS` plugin. This accumulates the scores for singles or doublets depending on the fit results that are accepted.

Note that precomputing scoring results as only two sets (singles or doublets) is a compromise for computation speed. Ideally the analysis would filter the results to accept and then compute match scores for the coordinates that remain. However in practice the coordinates of one of the doublet results is likely to match the same localisation as the corresponding single result, so totalling scores from either the single fit or doublet fit is valid. Using the option to ignore any candidates with neighbour candidates will ensure that no other fit results encroach within the fit window, and reduce scoring errors when summing single and doublet score results.

Matching of fit results to localisations can be performed using different methods:

| Method                    | Description  |
|---------------------------|--|
| <code>SIMPLE</code>       | Match localisations to their closest fit result  |
| <code>BY RESIDUALS</code> | <p>Process the fit results in the order determined by the residuals score. Each fit result (or pair of results) is matched to free localisations if within the match distance.</p> <p>This is valid for comparing the score between residuals=1 (all singles) and the effect of altering the residuals threshold to allow more doublets. It is not a true effect as doublets with a higher residuals score may not be first spot candidates that are fit</p> |
| <code>BY CANDIDATE</code> | Process the fit results in the order determined by the spot candidate rank. Each fit result (or pair of results) is matched to free localisations if within the match distance. This means results from higher ranking spot candidates may match a localisation that is actually closer to a lower ranking spot candidate results  |

### 9.14.2 Parameters

The following parameters can be configured:

| Parameter               | Description  |
|-------------------------|--|
| <code>T_TEMPLATE</code> | Select a template to configure the spot filter and fitting settings. Templates are loaded using the <code>T_TEMPLATE CONFIGURATION</code> plugin |

| Parameter          | Description   |
|--------------------|---|
| BENCHMARK SETTINGS | <p>Use the spot filter and fitting settings from the best filter identified by the BENCHMARK FILTER ANALYSIS plugin.</p> <p>This allows testing the residuals threshold to use given the optimum filter for fitting single spots</p>  |
| INITIAL STDDEV     | <p>The initial 2D Gaussian standard deviation for fitting. The width is expressed in pixels. By default it is set using the configured width of the PSF used to generate the data and should not need adjusting (unless it is intended to benchmark an incorrectly calibrated fitting algorithm)</p>  |
| SPOT FILTER TYPE   | <p>The type of filter to use.</p> <p>If a DIFFERENCE or JURY filter is selected then the plugin will present an additional dialogue to configure each additional spot filter</p>  |
| SPOT FILTER        | The name of the first spot filter   |
| RELATIVE DISTANCES | <p>Set to true to make all distances relative to the Half-Width at Half-Maxima (HWHM) of the PSF used to create the data. This is the same as when using the PEAK FIT plugin. This applies to the spot identification, and also the match distance. Note that the actual distances used will be recorded in the results table.</p> <p>Set to true to use absolute distances enabling fine control over the tested settings. Distances will be in pixels</p> |
| SMOOTHING          | The smoothing parameter for the first filter  |
| SEARCH WIDTH       | Define the region within which to search for a local maxima. The region size is $2n+1$ . This must be at least 1  |
| BORDER             | Define the number of border pixels to ignore. No maxima are allowed in the border   |
| FITTING WIDTH      | Define the size of the region around a candidate to use for fitting. The region size in pixels is set using the FITTING WIDTH multiplied by the INITIAL STDDEV  |
| FIT SOLVER         | <p>Define the solver used for fitting. Depending on the chosen solver a second dialog box will be presented to allow further configuration.</p> <p>See section 5.2.4: Fitting Parameters</p>  |
| FIT FUNCTION       | <p>The fit function.</p> <p>See section 5.2.4: Fitting Parameters</p>   |

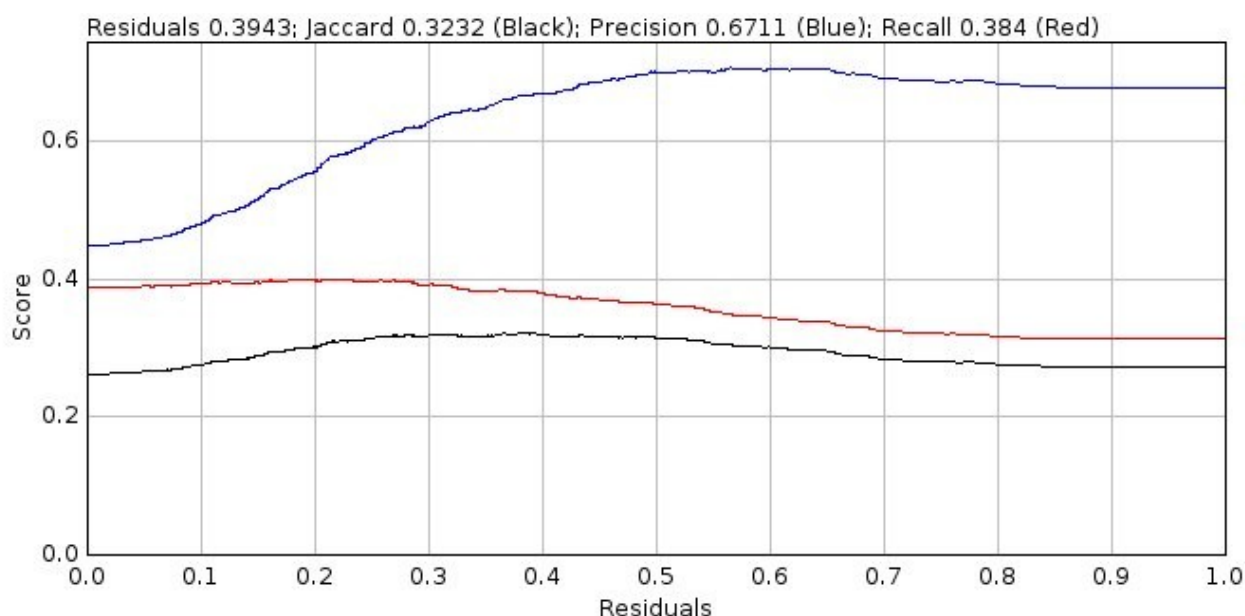
| Parameter          | Description  |
|--------------------|--|
| ITERATION INCREASE | The fitting algorithm normally increases the number of iterations when fitting multiple peaks. This increase is based on the results of testing on many datasets. However this factor can be increased by an addition factor to ensure all doublets are fit                          |
| SHOW OVERLAY       | Show an overlay on the image of the spot candidates that will be processed. Candidates are coloured using the number of localisations they match: 1 (red); 2 (green); over 2 (blue)  |
| SHOW HISTOGRAMS    | Show histograms of the number of spot matching results (number of localisations, number of neighbours, etc)  |
| SHOW RESULTS       | Show a results table with details of the fit results for each spot candidate   |
| SHOW JACCARD PLOT  | Show a plot of the performance scores verses the residuals threshold   |
| USE MAX RESIDUALS  | Use the maximum residuals score. This is a measure of the maximum asymmetry in the fit residuals.<br><br>The alternative is the average of the score for the two quadrants for which it is computed (i.e. the average asymmetry)   |
| MATCH DISTANCE     | The distance limit defining the minimum score for a match between a fitted localisation and the true localisation. The distance is expressed relative to the PSF width used to generate the data   |
| LOWER DISTANCE     | The distance limit defining the maximum score for a match between a fitted localisation and the true localisation. The distance is expressed relative to the PSF width used to generate the data.<br><br>Set to the same as the MATCH DISTANCE to ignore the ramped scoring function |
| SIGNAL FACTOR      | Define the limit for the difference between the fitted signal and the actual signal for a match. A value of (N-1) indicates the fit is allowed to be N-fold different, i.e. use 2 for a 3-fold difference.<br><br>Set to zero to ignore  |
| LOWER SIGNAL       | The limit defining the maximum score for a match between a fitted signal and the true signal.<br><br>Set to the same as the MATCH SIGNAL to ignore the ramped scoring function   |

| Parameter | Description                   |
|-----------|-------------------------------|
| MATCHING  | Select the matching algorithm |

### 9.14.3 Results

After spot candidates have been fitted and scored using their distance to the true localisations the results are summarised. Each candidate that was fitted as a single spot has a residuals score. The results are sorted by their residuals score and the binary scoring statistics (Precision, Recall, Jaccard) are computed for each possible threshold for the residuals score; for a given threshold all spots below the threshold are accepted as single fits and all above are accepted as double fits.

If the **SHOW JACCARD PLOT** option was selected the scoring statistics can be plotted against the residuals threshold:



The plot usually shows a region where the Jaccard score is optimal. Measures of the size of this region are calculated for the summary table. An ideal result would be the maximum Jaccard score when the residuals threshold is zero, i.e. the system fits all spots as singles and doublets and can pick the best result from each fit. However no filtering of fit results is performed and so many doublet results may contain poor spot data that can be filtered out using thresholds such as width, precision and SNR. Consequently the performance is often better when the single fits that have symmetric residuals (low residuals score) are accepted as singles and not as doublets, i.e. the residuals threshold is above 0. Unless the density is extremely low (and there are no doublets) it is unlikely the best score will be at residuals score 1, i.e. accept all fits as a single spot.

The analysis results are reported in a summary table:

| Field     | Description                                   |
|-----------|---|
| MOLECULES | The number of molecules in the simulated data |

| Field            | Description  |
|------------------|--|
| MATCHED          | The number of molecules that matched to a spot candidate   |
| DENSITY          | The density of the molecules   |
| MINN             | The minimum number of photons for a flourophore  |
| MAXN             | The maximum number of photons for a flourophore  |
| N                | The average number of photons for a flourophore  |
| S (NM)           | The PSF standard deviation   |
| A (NM)           | The pixel pitch  |
| SA (NM)          | The adjusted PSF standard deviation accounting for square pixels   |
| GAIN             | The total gain of the simulation   |
| READNOISE (ADUs) | The read noise of the simulation   |
| B (PHOTONS)      | The background photons   |
| NOISE (ADUs)     | The combined noise from read noise and background photons  |
| SNR              | The signal-to-noise ratio (using the average photons for signal)   |
| WIDTH            | The size of the fitting region. The actual region is $2N+1$  |
| METHOD           | The fit method   |
| OPTIONS          | Options for the fit method   |
| SCORE N=1        | The Mean+/Std.Dev (N) Median residuals score of candidates that match 1 result                                   |
| SCORE N=2        | The Mean+/Std.Dev (N) Median residuals score of candidates that match 2 results                                  |
| SCORE N=N        | The Mean+/Std.Dev (N) Median residuals score of candidates that match more than 2 results                        |
| ITER N=1         | The Mean+/Std.Dev (N) Median number of iterations to fit a candidate that matches only 1 actual result           |
| EVAL N=1         | The Mean+/Std.Dev (N) Median number of function evaluations to fit a candidate that matches only 1 actual result |

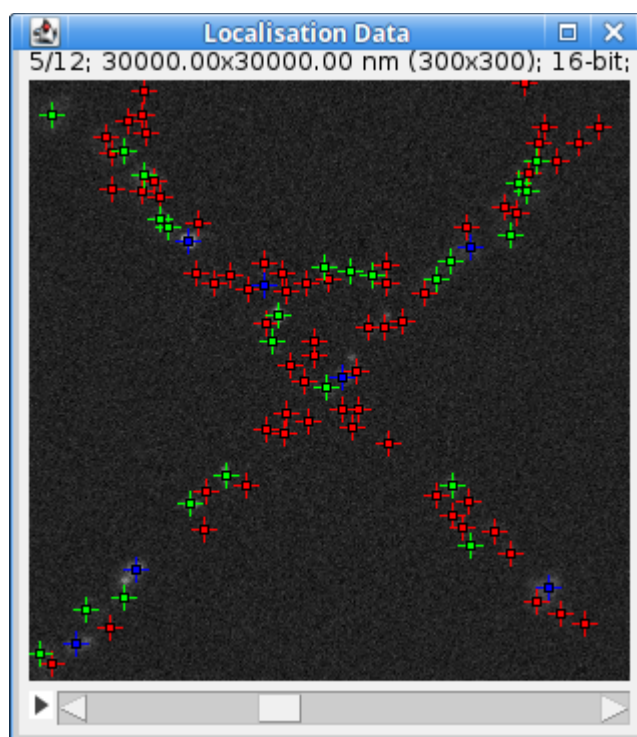
| Field        | Description   |
|--------------|---|
| ITER N>1     | The Mean+/-Std.Dev (N) Median number of iterations to fit a candidate that matches more than 1 actual result  |
| EVAL N>1     | The Mean+/-Std.Dev (N) Median number of function evaluations to fit a candidate that matches more than 1 actual result                                      |
| MATCHING     | The match mode  |
| BESTJ        | The best jaccard score if the system was able to correctly filter single and doublet fit results. This sets an upper bounds for what is possible            |
| J (R=1)      | The Jaccard score for fitting everything as singles (residuals threshold = 1)   |
| MAXJ         | The maximum Jaccard score   |
| RESIDUALS    | The residuals threshold used to obtain the maximum Jaccard score  |
| AREA +/- 15% | The average Jaccard score using the region around the maximum Jaccard score with a width of +/- 0.15  |
| AREA 98%     | The average Jaccard score using the region around the maximum Jaccard score with a Jaccard above 98% of the maximum   |
| MIN 98%      | The lower bounds of the region around the maximum Jaccard score with a Jaccard above 98% of the maximum   |
| MAX 98%      | The upper bounds of the region around the maximum Jaccard score with a Jaccard above 98% of the maximum   |
| RANGE 98%    | The range of the region around the maximum Jaccard score with a Jaccard above 98% of the maximum  |
| WMEAN 98%    | The weighted mean Jaccard score using the region around the maximum Jaccard score with a Jaccard above 98% of the maximum                                   |
| AREA >90%    | The average Jaccard score using the region around the maximum Jaccard score with a Jaccard above 90% of the performance improvement over no doublet fitting |
| MIN >90%     | The lower bounds of the region around the maximum Jaccard score with a Jaccard above 90% of the performance improvement over no doublet fitting             |



| Field      | Description   |
|------------|---|
| MAX >90%   | The upper bounds of the region around the maximum Jaccard score with a Jaccard above 90% of the performance improvement over no doublet fitting                   |
| RANGE >90% | The range of the region around the maximum Jaccard score with a Jaccard above 90% of the performance improvement over no doublet fitting                          |
| wMEAN >90% | The weighted mean Jaccard score using the region around the maximum Jaccard score with a Jaccard above 90% of the performance improvement over no doublet fitting |
| RUN TIME   | The run time for fitting  |

#### 9.14.3.1 Overlay

If the **SHOW OVERLAY** option was selected the plugin will put a cross on the input localisation image coloured using the number of localisations they match: 1 (red); 2 (green); over 2 (blue):



The overlay can be removed using 'IMAGE > OVERLAY > REMOVE OVERLAY'.

#### 9.14.3.2 Histograms

If the **SHOW HISTOGRAMS** option was selected the plugin prompts the user to select which histograms to display. The following histograms are available:

| Field                                      | Description  |
|--|--|
| CANDIDATE:N RESULTS IN CANDIDATE           | The number of actual results that match to a spot candidate  |
| ASSIGNED RESULT:N RESULTS IN ASSIGNED SPOT | The number of results that match to a spot candidate with n results (the sum of this histogram is the total number of results that match candidates) |
| SINGLES:NEIGHBOURS                         | The number of neighbour candidates in the fit region for candidates that match 1 actual result   |
| DOUBLETS:NEIGHBOURS                        | The number of neighbour candidates in the fit region for candidates that match 2 actual results  |
| MULTIPLES:NEIGHBOURS                       | The number of neighbour candidates in the fit region for candidates that match more than 2 actual results  |
| SINGLES:ALMOST                             | The number of neighbour candidates in the expanded fit region for candidates that match 1 actual result  |
| DOUBLETS:ALMOST                            | The number of neighbour candidates in the expanded fit region for candidates that match 2 actual results   |
| MULTIPLES:ALMOST                           | The number of neighbour candidates in the expanded fit region for candidates that match more than 2 actual results                                   |
| SCORE N=1                                  | The residuals score for candidates that match only 1 result  |
| SCORE N=2                                  | The residuals score for candidates that match 2 results (doublets)   |
| SCORE N=N                                  | The residuals score for candidates that match more than 2 results  |
| ITER N=1                                   | The number of iterations to fit a candidate that matches only 1 actual result  |
| EVAL N=1                                   | The number of function evaluations to fit a candidate that matches only 1 actual result  |
| ITER N>1                                   | The number of iterations to fit a candidate that matches more than 1 actual result   |
| EVAL N>1                                   | The number of function evaluations to fit a candidate that matches more than 1 actual result   |

### 9.14.3.3 Results Table

If the **SHOW RESULTS** option was selected the plugin outputs information about the fit of each spot candidate to a table. The values contained in the table can be used to filter the fit results to accept either the single or doublet fit. This can be done using the **DOUBLET FILTER ANALYSIS** plugin.

The following data is shown:

| Field      | Description   |
|------------|---|
| FRAME      | The frame containing the spot candidate   |
| X          | The x position of the candidate   |
| Y          | The x position of the candidate   |
| I          | The intensity of the candidate  |
| N          | The number of results that match to the candidate   |
| NEIGHBOURS | The number of spot candidates that are inside the fit region for the candidate  |
| ALMOST     | The number of spot candidates that are inside the expanded fit region for the candidate                                     |
| SCORE1     | The 1 <sup>st</sup> residuals score (calculated using quadrants based around a cross through the fitted position)           |
| SCORE2     | The 2 <sup>nd</sup> residuals score (calculated using quadrants based around a diaogonal cross through the fitted position) |
| R1         | The result of fitting the candidate as a single spot  |
| R2         | The result of fitting the candidate as a doublet (this is empty if the spot cannot be fit as a single)                      |
| ss1        | The residual sum-of-squares when fitting as a single spot   |
| ss2        | The residual sum-of-squares when fitting as a doublet   |
| v1         | The optimised value when fitting as a single spot (this may be the Maximum Likelihood Score or the sum-of-squares)          |
| v2         | The optimised value when fitting as a doublet (this may be the Maximum Likelihood Score or the sum-of-squares)              |
| R1         | The adjusted coefficient of determination when fitting a single spot  |
| R2         | The adjusted coefficient of determination when fitting a doublet  |
| AIC1       | The adjusted Akaike Information Criterion when fitting a single spot using the likelihood estimated from the residuals      |
| AIC2       | The adjusted Akaike Information Criterion when fitting a doublet using the likelihood estimated from the residuals          |

| Field | Description  |
|-------|--|
| BIC1  | The Bayesian Information Criterion when fitting a single spot using the likelihood estimated from the residuals  |
| BIC2  | The Bayesian Information Criterion when fitting a doublet using the likelihood estimated from the residuals  |
| MAIC1 | The adjusted Akaike Information Criterion when fitting a single spot using the likelihood from Maximum Likelihood fitting. If using Least Squares fitting this will be the same as AIC |
| MAIC2 | The adjusted Akaike Information Criterion when fitting a doublet using the likelihood from Maximum Likelihood fitting. If using Least Squares fitting this will be the same as AIC     |
| MBIC1 | The Bayesian Information Criterion when fitting a single spot using the likelihood from Maximum Likelihood fitting. If using Least Squares fitting this will be the same as BIC        |
| MBIC2 | The Bayesian Information Criterion when fitting a doublet using the likelihood from Maximum Likelihood fitting. If using Least Squares fitting this will be the same as BIC            |
| A1    | The angle between the vector defining the assymetry in the fit residuals and the position of the first spot of the doublet   |
| A2    | The angle between the vector defining the assymetry in the fit residuals and the position of the second spot of the doublet  |
| GAP   | The distance between the two spots in the doublet  |
| x1    | The X distance between the centres of the single fit and the position of the first spot of the doublet   |
| y1    | The Y distance between the centres of the single fit and the position of the first spot of the doublet   |
| x2    | The X distance between the centres of the single fit and the position of the second spot of the doublet  |
| y2    | The Y distance between the centres of the single fit and the position of the second spot of the doublet  |
| i1    | The number of iterations to fit as a single spot   |
| i2    | The number of iterations to fit as a doublet   |
| E1    | The number of function evaluations to fit as a single spot   |

| Field   | Description  |
|---------|--|
| E2      | The number of function evaluations to fit as a doublet |
| PARAMS1 | The fit parameters of the single spot                  |
| PARAMS2 | The fit parameters of the doublet                      |

## 9.15 Doublet Filter Analysis

Filters all the fit results produced by the DOUBLET ANALYSIS plugin using the specified filter settings and scores the results. This plugin requires the results generated by the DOUBLET ANALYSIS plugin. If these results are not available an error will be displayed.

This can be used to determine the best settings for optimum doublet fitting and filtering.

### 9.15.1 Selecting a two spot model

The DOUBLET ANALYSIS plugin fits a subset of the candidate spots in the benchmark image. These are fit as a single spot or as a double spot (doublet). The results are compared to the actual coordinates and scored. The DOUBLET FILTER ANALYSIS plugin applies filters to the fitting results to select if the single or doublet (or no fit) results should be accepted. The filtered results are then scored against the actual coordinates.

Unlike standard fitting and filtering, where the system must decide if the fit of a single spot is good, doublet fitting must decide if the fit is better with two spots rather than one. The system first requires that both results from the doublet are valid (i.e. using SNR, width, precision, etc). It then must decide if the fit is better using two spots rather than one. Given that the two spot model has more parameters it has an advantage over a simpler one spot model. The plugin provide various scores to determine if there is sufficient improvement in the fit to accept the more complex two spot model. The acceptance scores are described below.

#### 9.15.1.1 Adjusted Coefficient of Determination

$$R_{adj}^2 = 1 - \left( \frac{SS_{res}}{SS_{total}} \times \frac{n-1}{n-p-1} \right)$$

Where  $SS_{res}$  is the sum of the squared residuals (difference between the fitted model and the data),  $SS_{total}$  is the sum of the squared differences between the data and the mean of the data,  $n$  is the number of fitted points and  $p$  is the number of parameters. A higher score is better.

#### 9.15.1.2 Bias corrected Akaike Information Criterion

$$AIC = 2p - 2 \times \ln(\text{likelihood})$$

$$AICc = AIC + 2p \left( \frac{p+1}{n-p-1} \right)$$

Where  $AIC$  is Akaike Information Criterion,  $\text{likelihood}$  is the likelihood of the model given the data,  $n$  is the number of fitted points and  $p$  is the number of parameters.  $AICc$  is the bias corrected Akaike Information Criterion which corrects for an increase in the number of

parameters with respect to the number of fitted data points. A lower score is better.

### 9.15.1.3 Adjusted Coefficient of Determination

$$BIC = p \times \ln(n) - 2 \times \ln(\text{likelihood})$$

Where *BIC* is Bayesian Information Criterion, *likelihood* is the likelihood of the model given the data, *n* is the number of fitted points and *p* is the number of parameters. The BIC penalises additional parameters more than the AIC. A lower score is better.

### 9.15.1.4 Computing the likelihood

The likelihood score is output directly when using a Maximum Likelihood Estimator.

When using a Least Squares Estimator the variance of a model's residuals distributions,  $\sigma^2$ , is  $SS_{res}/n$ , where  $SS_{res}$  is the sum of the squared residuals (difference between the fitted model and the data) and *n* is the number of fitted points. Then, the maximum value of a model's log-likelihood function is:

$$\ln(\text{likelihood}) = -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\sigma^2) - \frac{1}{2\sigma^2} SS_{res}$$

## 9.15.2 Parameters

The following parameters can be configured:

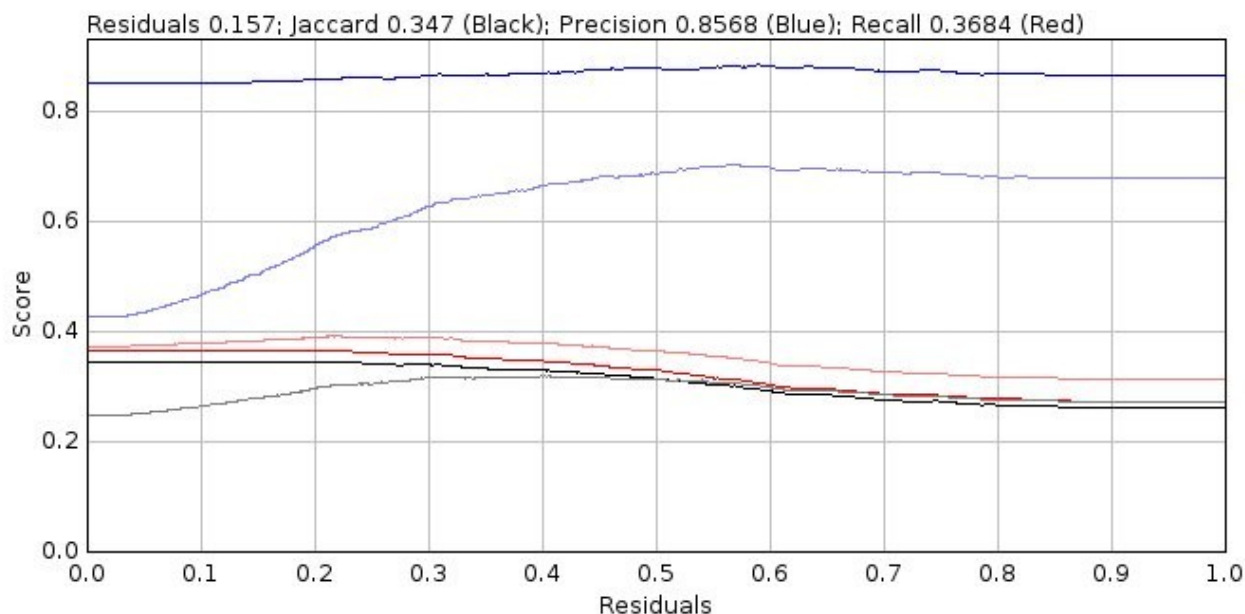
| Parameter          | Description   |
|--------------------|---|
| SELECTION CRITERIA | Specify the selection criteria for choosing a single or doublet fit.<br>AIC/BIC are computed using the likelihood taken from the sum of squared residuals. ML AIC/ML BIC are computed using the likelihood from the Maximum Likelihood Estimator (if available) |
| TEMPLATE           | Select a template to use to populate the filter settings.<br>Templates can be loaded using the TEMPLATE CONFIGURATION plugin  |
| BENCHMARK SETTINGS | Use the filter settings from the best filter identified by the BENCHMARK FILTER ANALYSIS plugin.<br>This allows testing the residuals threshold to use given the optimum filter for fitting single spots  |
| SHIFT FACTOR       | Any peak that shifts more than a factor of the initial peak standard deviation is discarded   |
| SIGNAL STRENGTH    | Any peak with a signal/noise below this level is discarded.<br>The signal is the calculated volume under the Gaussian. The image noise is calculated per frame  |
| MIN PHOTONS        | The minimum number of photons in a fitted peak. This requires a correctly calibrated gain to convert ADUs to photons.   |

| Parameter         | Description   |
|-------------------|---|
| MIN WIDTH FACTOR  | Any peak whose final fitted width is a factor smaller than the start width is discarded (e.g. 0.5x fold)  |
| MAX WIDTH FACTOR  | Any peak whose final fitted width is a factor larger than the start width is discarded (e.g. 2x fold)   |
| PRECISION         | Any peak with a precision above this level is discarded, i.e. not very good precision   |
| LOCAL BACKGROUND  | Use the local fitted background in the precision calculation. The default is to use the global background estimate for the frame  |
| DRIFT ANGLE       | <p>Set the maximum angle that a doublet centre can be away from the vector defining the asymmetry in the single fit residuals. The angle criteria is only applied if the doublet centre has shifted beyond the configured shift factor for filtering a single spot (i.e. if the doublet centre is far from the original fitted position it must be in the correct direction given the fit residuals).</p> <p>Note: This angle is displayed in the results table for each candidate fit result if the <code>SHOW RESULTS</code> option is chosen</p> |
| MIN GAP           | The minimum gap between the two centres of the doublet, i.e. prevent them being on top of each other  |
| SHOW RESULTS      | Show a results table with details of the fit results for each spot candidate  |
| SHOW JACCARD PLOT | Show a plot of the performance scores verses the residuals threshold  |
| USE MAX RESIDUALS | <p>Use the maximum residuals score. This is a measure of the maximum asymmetry in the fit residuals.</p> <p>The alternative is the average of the score for the two quadrants for which it is computed (i.e. the average asymmetry)</p>   |
| LOGGING           | Set to true to log detailed information about why each doublet fit was rejected   |
| TITLE             | A title to record in the results summary or the template file   |
| SAVE TEMPLATE     | <p>Set to true to save a template containing all the filter, fit and filtering settings from the benchmark plugins to file. The template will contain the best residuals threshold given the filter settings.</p> <p>This template can be loaded using the <code>TEMPLATE CONFIGURATION</code> plugin for use in the <code>PEAK FIT</code> plugin when fitting localisation data</p>  |

### 9.15.3 Results

After spot candidates have been filtered using the specified criteria the results are summarised. Each candidate that was fitted as a single spot has a residuals score. The results are sorted by their residuals score and the binary scoring statistics (Precision, Recall, Jaccard) are computed for each possible threshold for the residuals score; for a given threshold all spots below the threshold are assessed as single fits and all above are subjected to the selection criteria to choose the single or doublet fit.

If the **SHOW JACCARD PLOT** option was selected the scoring statistics can be plotted against the residuals threshold:



This plot is similar to the output from the **DOUBLET ANALYSIS** plugin where results were either accepted as a single or doublet based purely on the residuals score. The performance metrics for this simple selection are shown in light blue (precision), light red (recall) and grey (Jaccard).

However the plot now shows the results of applying the selection criteria on the fit of the model as a single or doublet and also the filtering of the spots using any spot selection filters. The performance metrics for the strict selection are shown in blue (precision), red (recall) and black (Jaccard). The recall will be lower (as strict filtering removes many spots) however the precision is higher as the system correctly picks good spots (true positives) more often than bad spots (false positives).

As before the plot usually shows a region where the Jaccard score is optimal. Measures of the size of this region are calculated for the summary table. An ideal result would be the maximum Jaccard score when the residuals threshold is zero, i.e. the system fits all spots as singles and doublets and can pick the best result from each fit. However this requires all spots to be fit twice which is computationally expensive. So the system computes the weighted average for the residuals threshold for the region of the curve within a fraction of the maximum performance improvement over no doublet fitting, i.e. the residuals score to achieve most of the performance benefit. This residuals score is often much higher than the residuals score for the best Jaccard score and so is a compromise between maximum performance and speed.



The filter analysis results are reported in a summary table:

| Field         | Description   |
|---------------|---|
| DENSITY       | The density of the molecules  |
| S             | The PSF width used for fitting  |
| WIDTH         | The size of the fitting region. The actual region is $2N+1$   |
| METHOD        | The fit method  |
| OPTIONS       | Options for the fit method  |
| BESTJ         | The best Jaccard score possible if the system is capable of accepting fit results that score more than 0.5 for true positive (fractional scores are possible due to the use of weighted scoring) and rejecting all others |
| TITLE         | The title from the input dialog   |
| USE RESIDUALS | Specify if the residuals score is the maximum or the average of residuals score 1 and 2 for the single spot   |
| SELECTION     | The selection criterion   |
| SHIFT         | The shift filter value  |
| SNR           | The SNR filter value  |
| PHOTONS       | The Signal (photons) filter value   |
| MIN WIDTH     | The minimum width filter value  |
| MAX WIDTH     | The maximum width filter value  |
| PRECISION     | The precision filter value  |
| LOCAL B       | True if using a local background for the precision filter   |
| ANGLE         | The angle limit   |
| GAP           | The minimum gap between doublets centres  |
| J ( $R=1$ )   | The Jaccard score for fitting everything as singles (residuals threshold = 1)   |
| MAXJ          | The maximum Jaccard score.<br>Note: The difference between MAXJ and J ( $R=1$ ) is the performance improvement for using doublet fitting  |

| Field        | Description   |
|--------------|---|
| RESIDUALS    | The residuals threshold used to obtain the maximum Jaccard score  |
| AREA +/- 15% | The average Jaccard score using the region around the maximum Jaccard score with a width of +/- 0.15  |
| AREA 98%     | The average Jaccard score using the region around the maximum Jaccard score with a Jaccard above 98% of the maximum   |
| MIN 98%      | The lower bounds of the region around the maximum Jaccard score with a Jaccard above 98% of the maximum   |
| MAX 98%      | The upper bounds of the region around the maximum Jaccard score with a Jaccard above 98% of the maximum   |
| RANGE 98%    | The range of the region around the maximum Jaccard score with a Jaccard above 98% of the maximum  |
| WMEAN 98%    | The weighted mean Jaccard score using the region around the maximum Jaccard score with a Jaccard above 98% of the maximum   |
| AREA >90%    | The average Jaccard score using the region around the maximum Jaccard score with a Jaccard above 90% of the performance improvement over no doublet fitting       |
| MIN >90%     | The lower bounds of the region around the maximum Jaccard score with a Jaccard above 90% of the performance improvement over no doublet fitting                   |
| MAX >90%     | The upper bounds of the region around the maximum Jaccard score with a Jaccard above 90% of the performance improvement over no doublet fitting                   |
| RANGE >90%   | The range of the region around the maximum Jaccard score with a Jaccard above 90% of the performance improvement over no doublet fitting                          |
| WMEAN >90%   | The weighted mean Jaccard score using the region around the maximum Jaccard score with a Jaccard above 90% of the performance improvement over no doublet fitting |

#### 9.15.3.1 Results Table

If the **SHOW RESULTS** option was selected the plugin outputs information about the fit of each spot candidate to a table. See section 9.15.3.1 for more details.

## 9.16 Image Background

Produces a background intensity image and a mask from a sample image.

The IMAGE BACKGROUND plugin is used to generate suitable input for the CREATE DATA plugin. The CREATE DATA plugin creates an image by simulating fluorophores using a distribution. One allowed distribution is the region defined by a mask. The fluorophores are created and then drawn on the background. The background can be an input image. Both the mask and background image can be created from a suitable *in vivo* image using the IMAGE BACKGROUND plugin. The purpose would be to simulate fluorophores in a distribution that matches that observed in super-resolution experiments.

The plugin requires that an image is open. The plugin dialogue is shown in Illustration 35. The plugin has two parameters the bias and the blur. These are described in the following section.

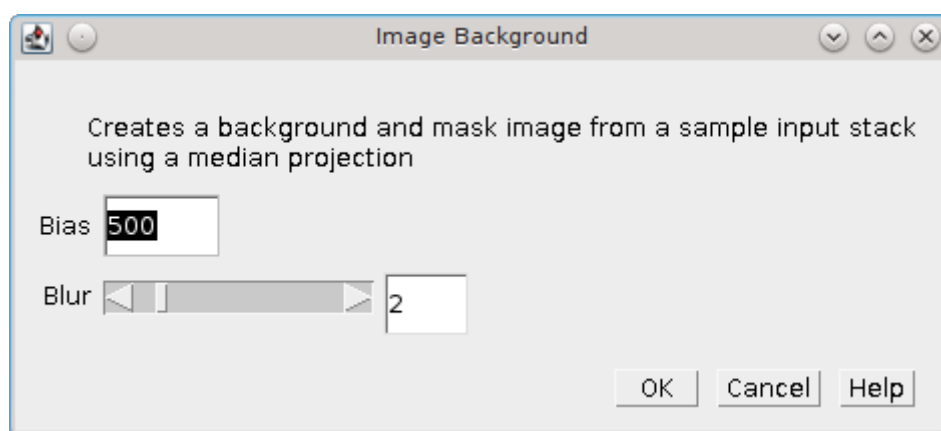


Illustration 35: Image background plugin dialogue

### 9.16.1 Image analysis

The IMAGE BACKGROUND plugin first computes a median intensity projection of the input image. A Gaussian blur is then applied to the projection to smooth the image. The blur parameter controls the size of the Gaussian kernel.

The bias is subtracted from the blurred image. The bias is an offset that may be added to the pixel values read by the camera so that negative noise values can be observed. It is a constant level that can be subtracted. What remains should be the background level. This can be ignored using a bias of zero.

Two output images are then displayed:

1. Background: the blurred projection
2. Mask: the blurred projection subjected to the ImageJ default thresholding method

## 9.17 Load Localisations

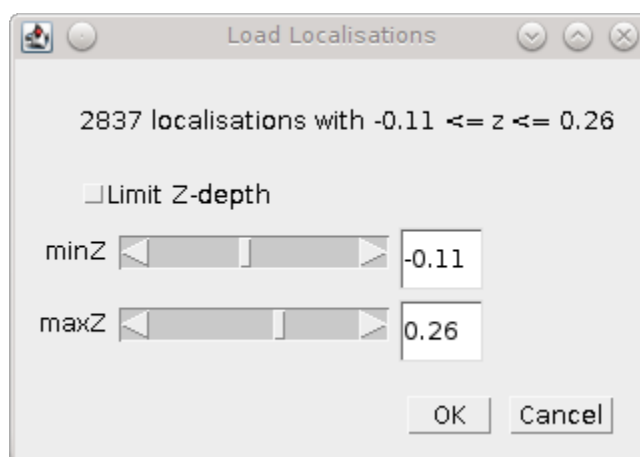
Loads the localisation file created by CREATE DATA into memory.

The CREATE DATA plugin simulates fluorophores in a 3D volume and then creates an image representation of the localisations. The localisations are saved to file using a simple tab delimited format. The records have the following fields:

| Field | Name      | Description          |
|-------|-----------|----------------------|
| 1     | T         | Time frame           |
| 2     | Id        | The identifier       |
| 3     | X         | The x-coordinate     |
| 4     | Y         | The y-coordinate     |
| 5     | Z         | The z-coordinate     |
| 6     | INTENSITY | The signal intensity |

The **LOAD LOCALISATIONS** plugin will ask the user to select a file. The file is then read assuming the **CREATE DATA** localisations data format. Any lines beginning with the # character are ignored. The units are assumed to be in pixels.

The plugin then presents a dialogue allowing the user to select a subset of localisations by z-depth (Illustration 36). The dialogue shows the minimum and maximum limits of the z coordinates and provides an option to limit the z-depth. The limits can then be adjusted using the sliders.



*Illustration 36: Load localisations plugin dialogue.*

The (subset of) localisations are saved to a results set in memory named 'Localisations'. The localisation results do not have a calibration or data on the peak-width and amplitude. Therefore the results cannot be used in many of the SMLM plugins. However the XY coordinates can be used in compatible plugins, for example within the **RESULTS MATCH CALCULATOR** plugin. Thus the **LOAD LOCALISATIONS** plugin can be used to load selected slices of data using the z-depth filter and compute the match statistics to fitted results. This allows benchmarking the performance of fitting spots that are both in and progressively out-of-focus. Since the plugin is fully supported by the ImageJ macro recorder this analysis can be automated in a macro script.

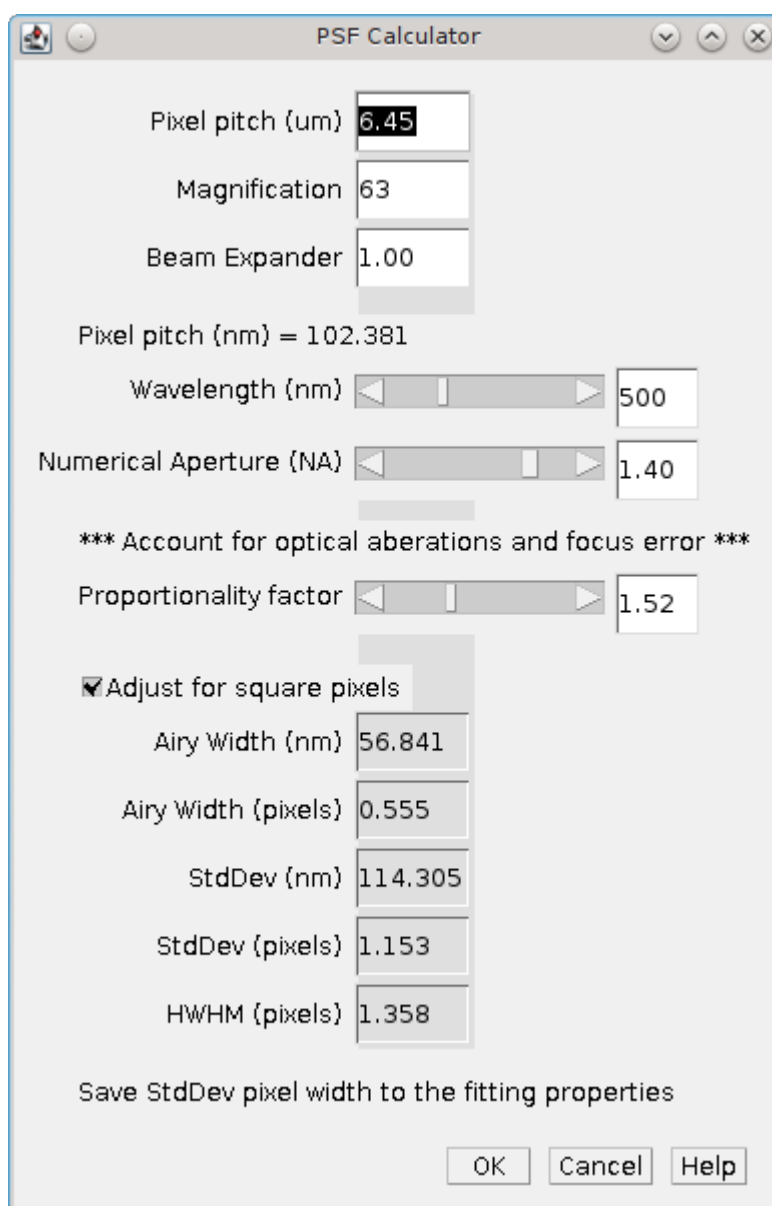
## 10 Calibration Plugins

The following plugins contain functionality to estimate the width of the Point Spread Function (PSF) for an imaging set-up and analyse the noise and gain of the imaging camera.

The plugins are described in the following sections using the order presented on the `PLUGINS > GDSC SMLM > CALIBRATION` menu.

### 10.1 PSF Calculator

A simple plugin that estimates your Gaussian approximation to the PSF using the microscope imaging parameters and the wavelength of light.



*Illustration 37: The PSF CALCULATOR dialogue.*

The calculator uses the following formula:

$$\text{Airy Width} = \frac{\lambda}{2\pi NA}$$

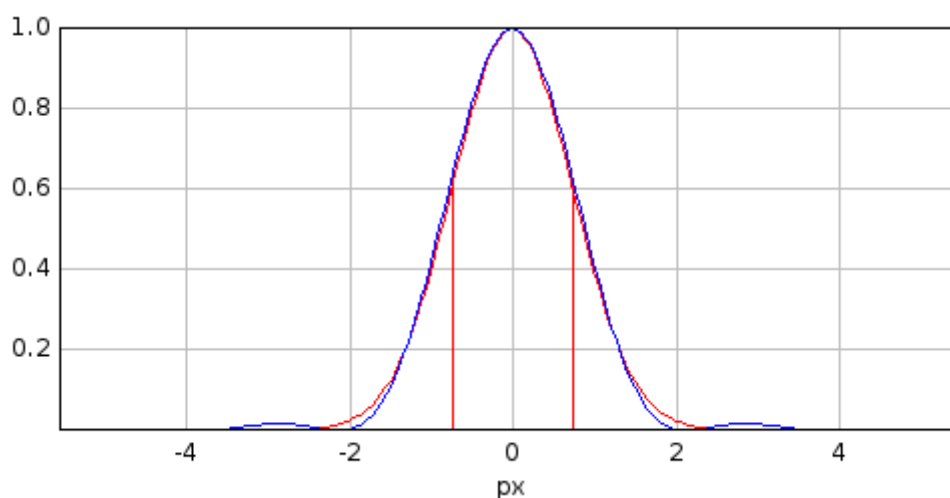
Where:

Airy Width     The width of the Airy pattern

$\lambda$                 The wavelength (in nm)

NA                The Numerical Aperture

The Airy profile can be approximated by a Gaussian profile. The equivalent Gaussian profile is created using a standard deviation of 1.323 times the Airy width. The PSF Calculator will show a plot of the Airy profile (blue) and the corresponding Gaussian profile (red). This is interactively updated when the parameters for the calculator are changed:



Note that the Gaussian is a good approximation until the tails of the Airy pattern. The Airy pattern contains waves of decreasing power out to infinity which are not modelled by the Gaussian.

The calculator allows for additional adjustments to be made to the calculated Gaussian standard deviation. To account for optical aberrations in the microscope the width is allowed to be wider by a set proportionality factor. The Gaussian standard deviation ( $s$ ) can then be adjusted ( $s_a$ ) for an accurate representation on square pixels using the following formula:

$$s_a = \sqrt{s^2 + a^2/12}$$

The Gaussian Half-Width and Half-Maxima (HWHM) is calculated from the standard deviation by multiplying by  $\sqrt{2 \cdot \ln(2)} = 1.177$ .

The following table describes the parameters of the plugin. The calculated properties (show in grey in the table below) are updated dynamically.

| Parameter                     | Description                 |
|-------------------------------|-----------------------------|
| PIXEL PITCH ( $\mu\text{M}$ ) | The camera pixel size       |
| MAGNIFICATION                 | The objective magnification |

|                          |  |
|--------------------------|--|
| BEAM EXPANDER            | Any addition magnification   |
| PIXEL PITCH (NM)         | The calculated image pixel size  |
| WAVELENGTH (NM)          | The wavelength of light used for the image ( $\lambda$ )                                 |
| NUMERICAL APERTURE       | The objective numerical aperture (NA)  |
| PROPORTIONALITY FACTOR   | The proportionality factor (set to 1 to match the Gaussian to the Airy profile)          |
| ADJUST FOR SQUARE PIXELS | Perform square pixel adjustment (set to false to match the Gaussian to the Airy profile) |
| AIRY WIDTH (NM)          | The calculated PSF Airy width in nanometres  |
| AIRY WIDTH (PIXELS)      | The calculated PSF Airy width in pixels  |
| STDDEV (NM)              | The calculated PSF Gaussian standard deviation in nanometres                             |
| STDDEV (PIXELS)          | The calculated PSF Gaussian standard deviation in pixels                                 |
| HWHM (PIXELS)            | The calculated PSF Gaussian HWHM in pixels   |

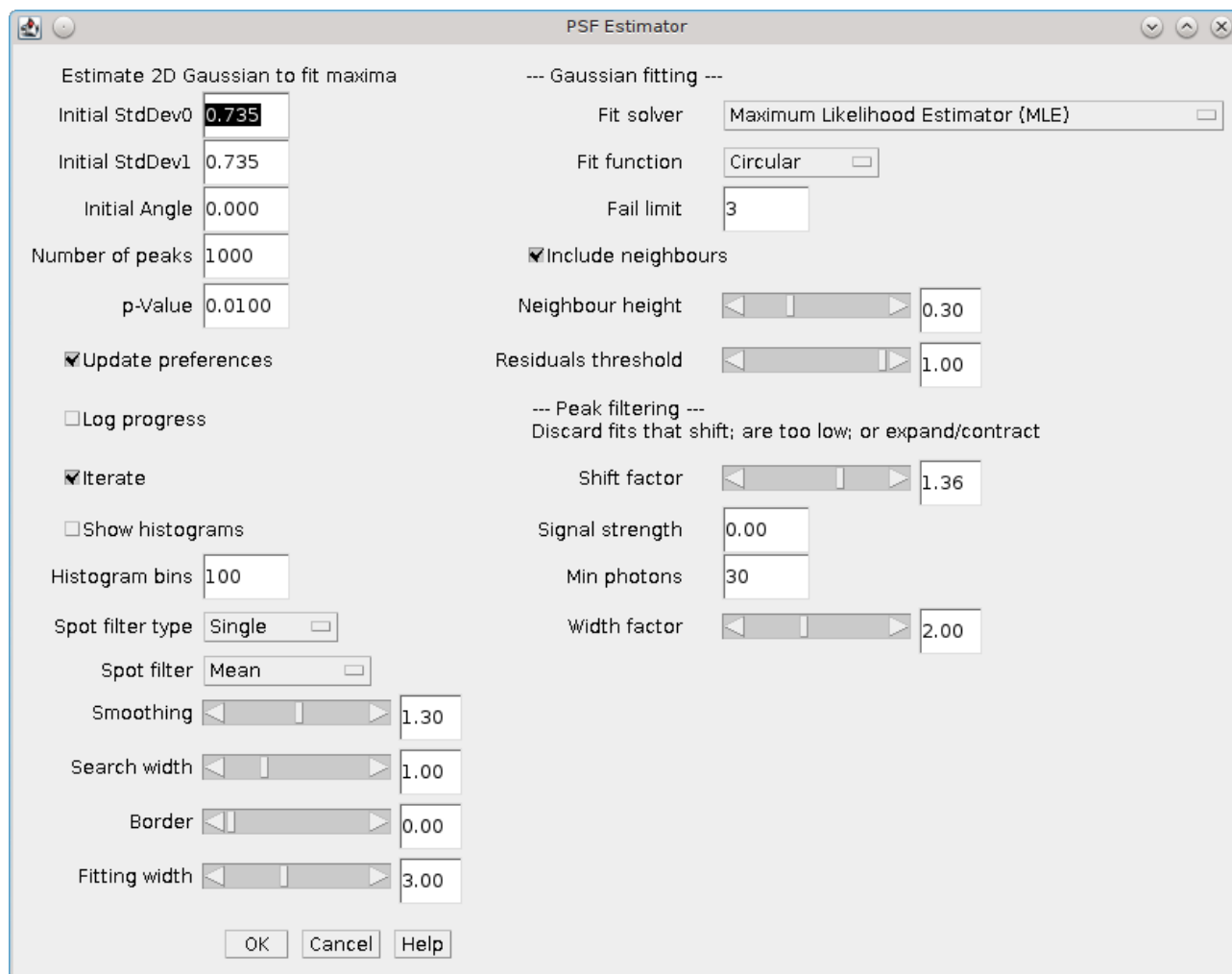
Note that the first three fields are only used to calculate the image pixel pitch. If this is already known then it can be entered into the `PIXEL PITCH (UM)` field (note the use of micrometres and not nanometres) and the `MAGNIFICATION` and `BEAM EXPANDER` can be set to 1. The `PIXEL PITCH` in nanometres is then used to convert the calculated widths in nanometres to pixel dimensions.

Clicking OK will save the PSF standard deviation in pixels to the global properties. This will be used in the `PEAK FIT` plugin.

Please contact [a.herbert@sussex.ac.uk](mailto:a.herbert@sussex.ac.uk) if you have feedback on the calculated width from the plugin versus your measured PSF using quantum dots (or other single-point light sources) on calibration images.

## 10.2 PSF Estimator

A plugin that estimates the PSF using a test image. The fit configuration is the same as in the `FIT CONFIGURATION` plugin with extra parameters provided to control the estimation. The `PSF ESTIMATOR` dialogue is shown in Illustration 38. Note that a second dialogue will collect parameters specific for the selected `FIT SOLVER`.



*Illustration 38: PSF ESTIMATOR dialogue.*

The estimator uses the starting configuration to fit N peaks taken from randomly selected frames in the image stack. The averages of the fitted parameters are then used as the start parameters to perform fitting again. This iterates until the Gaussian parameters do not significantly change. The parameters controlling the estimation are described below.

| Parameter           | Description   |
|---------------------|---|
| NUMBER OF PEAKS     | The number of fitted peaks to use to estimate the Gaussian parameters. The parameters are estimated by averaging across all the fitted peaks. |
| P-VALUE             | The p-value to use for significance testing, i.e. are the parameters the same using a Student's T-test at the given significance              |
| UPDATES PREFERENCES | If selected the plugin will update the global configuration with the calculated PSF values  |
| LOG PROGRESS        | Log progress of the estimator to the ImageJ log window  |



| Parameter       | Description  |
|-----------------|--|
| ITERATE         | <p>When the PSF parameters have converged and a <code>FREE</code> fitting option was chosen a test is done to determine if the angle or Y-width are significant. If not then the estimator will restart ignoring the insignificant parameter from the modelled PSF. The order of iterations is:</p> <p><code>FREE &gt; FREE CIRCULAR &gt; CIRCULAR</code></p> <p>Note these statistics often don't work so unless you expect astigmatism you can choose to start with a <code>CIRCULAR Gaussian</code> and just find the estimated widths.</p> |
| SHOW HISTOGRAMS | <p>Show a histogram of the estimated parameters from the final fitting run. A histogram is shown for each parameter. These can be used to verify the mean of the parameter distribution is a suitable estimate for the parameter.</p>  |
| HISTOGRAM BINS  | <p>The number of bins to use on the histograms</p>   |

Note that the estimator may not find any peaks if the fitting parameters are badly configured. The estimator can be reset to defaults by holding down the **CONTROL** key when running the plugin. The default values are shown below:

| Parameter                 | Value                      |
|---------------------------|----------------------------|
| INITIAL STDDEV0,1         | 1                          |
| INITIAL ANGLE             | 0                          |
| SPOT FILTER TYPE          | Single                     |
| SPOT FILTER               | Mean                       |
| SMOOTHING                 | 1.3                        |
| SEARCH WIDTH              | 1                          |
| BORDER                    | 1                          |
| FITTING WIDTH             | 3                          |
| FIT SOLVER                | Least Squares Estimator    |
| <i>FIT CRITERIA</i>       | <i>Least-squared error</i> |
| <i>SIGNIFICANT DIGITS</i> | <i>5</i>                   |
| <i>COORD DELTA</i>        | <i>0.0001</i>              |
| <i>LAMBDA</i>             | <i>10</i>                  |
| <i>MAX ITERATIONS</i>     | <i>20</i>                  |
| FIT FUNCTION              | Circular                   |
| FAIL LIMIT                | 3                          |
| INCLUDE NEIGHBOURS        | True                       |
| NEIGHBOUR HEIGHT          | 0.3                        |
| RESIDUALS THRESHOLD       | 1                          |
| SHIFT FACTOR              | 1                          |
| SIGNAL STRENGTH           | 0                          |
| MIN PHOTONS               | 30                         |
| WIDTH FACTOR              | 2                          |

### 10.3 Mean-Variance Test

The **MEAN-VARIANCE TEST** plugin can be used to calculate the gain and read noise of the microscope Charged Coupled Device (CCD) camera. The plugin requires a set of calibration images. A single-image mode is available but will provide less information on the camera.

#### 10.3.1 Multiple Input Images

When run the plugin will present a folder selection dialogue. The folder should contain a

set of calibration images. All the images should be taken of the same view with the camera in the same gain mode.

At least one image should be taken with no exposure time. This is the image the camera records when no light has been registered on the sensor and is called the bias image.

The remaining images should be a representative series of different exposures. The purpose is to analyse how the image noise varies with exposure time. In order for the analysis to be valid no images should saturate the camera bit-depth. E.g. for a 12-bit camera all images should have pixel values below  $2^{10} - 1 = 4095$ .

All the images in the folder are opened and processed by the plugin. Each image must contain at least 2 frames. If the filename contains a valid integer delimited by a space or a period character (.) then this will be taken as the exposure time. Otherwise an arbitrary exposure time is used, either zero for the first image (alphabetically sorted) or 9999 for the rest.

### 10.3.2 Analysis

If all the images are valid (contain at least 2 frames) then the plugin will perform the mean-variance test. The average value of the bias images is used as the bias. Each image is then analysed in turn. The mean of each frame is computed. Then a pairwise difference image (i.e. one frame subtracted from the other) is computed for all-vs-all frames. The variance of the difference image is recorded and used to approximate the camera gain:

$$gain = \frac{variance}{mean - bias}$$

This is recorded in a summary table. A graph is then produced of the mean verses the variance. This data is fitted with a straight line. The gradient of the line is the camera gain. The read noise of the camera is computed as:

$$read\ noise = \frac{\sqrt{bias\ variance}}{gain}$$

If the bias has multiple difference images then the average variance is used to calculate the read noise.

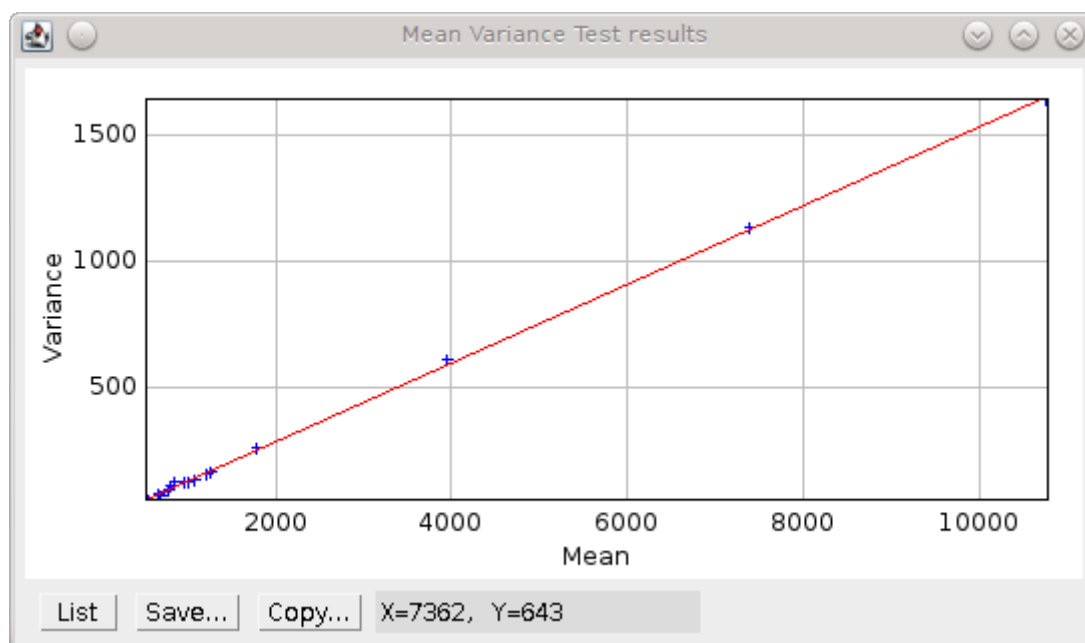
### 10.3.3 Output

The plugin produces a summary table of the analysis for each pair of frames. The table shows the following data:

| Column   | Description   |
|----------|---|
| IMAGE    | The source image  |
| EXPOSURE | The image exposure. This is the first integer number delimited by a space or period in the image title or, if no number can be found in the image title, zero for the first image and 9999 for the others |
| SLICE1   | The first frame (slice) used from the image   |
| SLICE2   | The second frame (slice) used from the image  |

|          |  |
|----------|--|
|          |  |
| MEAN1    | The mean of slice 1                                      |
| MEAN2    | The mean of slice 2                                      |
| MEAN     | The mean of both slices                                  |
| VARIANCE | The variance of the difference image                     |
| GAIN     | The gain estimate: $gain = \frac{variance}{mean - bias}$ |

The plugin will produce a plot of the mean-variance data as show in Illustration 39. The plot will show the best fit line in red. If the data points with the highest mean lie well under the line it is possible that these images had saturated pixel values and should be removed from the input data set.



*Illustration 39: Mean-variance plot produced by the Mean-Variance Test plugin. The best fit line is shown in red.*

The plugin reports the final calculated gain and read noise to the ImageJ log, e.g.

Mean Variance Test

Directory = /images/CameraCalibration/CameraGain-2-EmGain-0/

Bias = 515.4 +/- 7.4 (ADU)

Variance = -21.78 + 0.1557 \* mean

Read Noise = 47.53 (e-)

Gain = 1 / 6.422 (ADU/e-)

The parameters for the best fit line are shown as  $Variance = a + b * mean$ . The parameter  $b$  is the gain. The read noise is shown in electrons. The units for the gain are Analogue-to-Digital Unit (ADU) per electron.

Note that the gain can be expressed as electrons per ADU and so the output shows the gain using 1 over the reciprocal of the fit parameter to allow comparison with manufacturer gain values. E.g. in the example above  $1 / 6.422 = 1 / (1 / 0.1557)$  and the gain would be 6.422 e-/ADU.

### 10.3.4 Single Image Mode

The plugin can be run using a single image. Single image mode cannot compute the camera bias or read noise and the gain values are not as accurate as the full test using multiple images.

Hold the `SHIFT` key down when running the plugin and the analysis will be performed on the currently active image. The image must have more than one slice to allow difference images to be computed and be a white light image with a constant uniform exposure across the image field, i.e. no significant image features.

In single-image mode the plugin will compute the pairwise comparison of consecutive frames in the image and for each pair compute the approximate camera gain:

$$gain = \frac{variance}{mean - bias}$$

The bias must be provided since there is no input bias image; the plugin will ask the user to input the camera bias. The results will be displayed in a table as described above.

The plugin provides a plot of gain verses slice and a histogram of the values. These can be used to determine if the gain is constant throughout the image and so is a good estimate.

## 10.4 Mean-Variance Test (EM-CCD)

This plugin is similar to the `MEAN-VARIANCE TEST` plugin but is used on images taken using an Electron Multiplying (EM) CCD camera. An EM-CCD camera uses a multiplication device to increase the number of electrons that are extracted from the imaging sensor before the electrons are counted. The average number of electrons output from the multiplying device for each input electron is a constant known as the EM-gain. The plugin will compute the EM-gain of the camera using a set of calibration images. A single-image mode is available but will provide less information on the camera.

The analysis can only be performed if the gain for the camera in non-EM mode is already known. If the `MEAN-VARIANCE TEST` plugin has been used to calculate the gain in the same ImageJ session then the value will be stored in memory. If the camera gain is not known then using a value of 1 will allow the plugin to run and the output EM-gain will be the total gain of the system.

### 10.4.1 Multiple Input Images

Input images requirements are the same as the `MEAN-VARIANCE TEST` plugin: images should be taken of the same view using different exposure times. Each image must have at least two frames. All images must be taken with the camera in the same gain mode and EM-gain mode. A bias image (zero exposure) must be provided.

If all the images are valid the plugin will show a dialogue asking for the camera gain (Illustration 40). This will be the last entered value or the value computed by the `MEAN-`

VARIANCE TEST plugin.

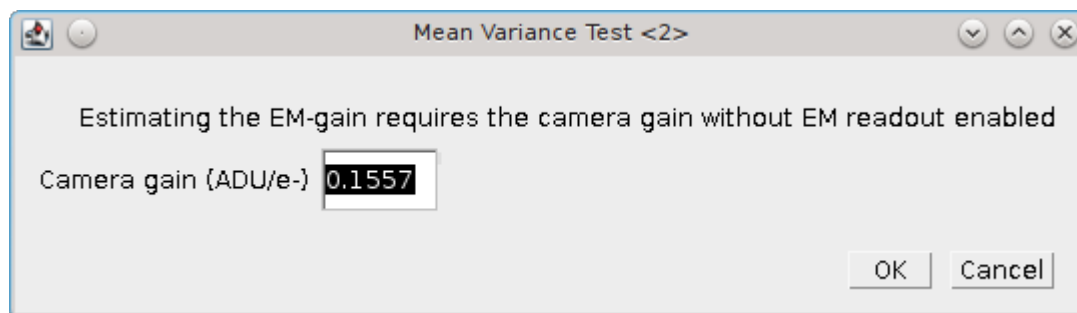


Illustration 40: EM-gain dialogue of the Mean-Variance Test (EM-CCD) plugin

### 10.4.2 Analysis

The images are analysed as per the MEAN-VARIANCE TEST plugin. However the analysis of the difference image is used to approximate the camera EM-gain:

$$EM\ gain = \frac{variance}{(mean - bias)(2 \times gain)}$$

This is recorded in a summary table. A graph is then produced of the mean verses the variance. This data is fitted with a straight line. The gradient of the line is the EM-gain multiplied by twice the camera gain therefore the EM-gain can be computed as:

$$EM\ gain = \frac{gradient}{2 \times gain}$$

### 10.4.3 Output

The plugin summary table and mean-variance plot are the same as the MEAN-VARIANCE TEST plugin. The final calculated EM-gain and total gain is reported to the ImageJ log, e.g.

Mean Variance Test

```
Directory = /images/CameraCalibration/CameraGain-2-EmGain-250/
Bias = 512.3 +/- 13.15 (ADU)
Variance = -36550.0 + 79.66 * mean
Read Noise = 0.3301 (e-)
Gain = 1 / 6.422 (ADU/e-)
EM-Gain = 255.8
Total Gain = 39.83 (ADU/e-)
```

The total gain is the EM-gain multiplied by the camera gain. As can be seen from comparison of the analysis results with and without the EM-mode the use of EM amplification dramatically reduces the camera read noise and greatly enhances the pixel values (ADUs) produced per electron. This allows images of weak photon signals to be made, for example in single-molecule light microscopy.

The total gain can be used to convert the ADUs into photons if the camera Quantum Efficiency (QE) is known. The QE states how many photons are converted into an electron charge when they hit the camera sensor; the QE units are electrons per photon

(e-/photon). This can be provided by the camera manufacturer and is dependent on the wavelength of light. The photon signal is therefore:

$$\text{Photons} = \frac{\text{ADUs}}{\text{total gain} \times \text{QE}}$$

The total gain multiplied by the QE is known as the system gain. The system gain is used as an input parameter in the **PEAK FIT** plugin to convert the pixel values into photons.

### 10.4.4 Single Image Mode

The plugin can be run using a single image. Single image mode cannot compute the camera bias or read noise and the gain values are not as accurate as the full test using multiple images.

Hold the **SHIFT** key down when running the plugin and the analysis will be performed on the currently active image. The image must have more than one slice to allow difference images to be computed and be a white light image with a constant uniform exposure across the image field, i.e. no significant image features.

In single-image mode the plugin will compute the pairwise comparison of consecutive frames in the image and for each pair compute the approximate camera gain:

$$\text{EM gain} = \frac{\text{variance}}{(\text{mean} - \text{bias})(2 \times \text{gain})}$$

The bias must be provided since there is no input bias image; the plugin will ask the user to input the camera bias and camera gain. Using a camera gain of 1 will calculate the total gain of the system. The results will be displayed in a table as described above.

The plugin provides a plot of gain verses slice and a histogram of the values. These can be used to determine if the gain is constant throughout the image and so is a good estimate.

## 10.5 EM-Gain Analysis

Analysis a white light image from an EM-CCD camera, construct a histogram of pixel intensity and fit the histogram to obtain the bias, EM-gain, read noise and photons per pixel (see Ulbrich & Isacoff (2007) Supplementary Information).

### 10.5.1 EM-CCD Probability Model

The **EM-GAIN ANALYSIS** plugin uses an analysis that assumes that the EM-CCD camera has three main sources of noise:

1. Photon shot noise occurs when light is emitted from an object. Although the average rate of light from an object is constant for a given time, e.g. 30 photons/second, each photon will arrive at a different time and the gaps between them will vary. This will lead to changes in the number of photons counted each second. This noise follows a Poisson distribution with a mean of the average photon emission rate.
2. The photons are converted to electrons on the camera sensor. These electrons are then multiplied in the Electron Multiplication (EM) gain register. This multiplication increases the number of electrons to be read and reduces the relative size of any

error introduced when reading the value. However the EM-gain process is random and introduces noise that is modelled using a Gamma distribution with a shape parameter equal to the number of input electrons and the scale parameter equal to the gain.

3. Read noise occurs when the values stored on the camera chip for each pixel are read and converted to numbers. This noise follows a Gaussian distribution with mean zero and variable standard deviation.

The probability of observing a pixel value given an input number of photons is therefore a convolution of a Poisson, Gamma and Gaussian distribution. The convolution of the Poisson and Gamma distribution can be expressed as:

$$G_{p,m}(c) = e^{-p} \delta(c) + \sqrt{\frac{p}{cm}} e^{-\frac{c}{m}-p} \text{Bessel}I_1\left(2\sqrt{\frac{cp}{m}}\right)$$

where:

|              |   |
|--------------|---|
| p            | The average number of photons                         |
| m            | The EM-gain multiplication factor                     |
| c            | The observed pixel count                              |
| $\delta(c)$  | The Dirac delta function (1 when $c=0$ , 0 otherwise) |
| BesselI1     | Modified Bessel function of the 1 <sup>st</sup> kind  |
| $G_{p,m}(c)$ | The probability of observing the pixel count c        |

The output of this is subsequently convolved numerically (no algebraic solution exists) with a Gaussian function with standard deviation equal to the camera read noise and mean equal to the camera bias.

### 10.5.1.1 Camera Bias

Note that in order to observe the read noise of the camera a bias (offset) is added to the camera pixel values. This allows a pixel to record negative read noise on very low counts which would not be possible using unsigned integer values as no value below zero is allowed. The bias for the camera is set by the manufacturer and is set at a value far greater than the expected read noise of the system, e.g. 100, 400, 500 or 1000 for a read noise of 3-30 ADUs (Analogue to Digital Units, or pixel values).

### 10.5.2 Input image

The plugin requires a white light image where each pixel has been exposed to the same number of photons. This can be produced by imaging without a sample instead using white paper in front of the objective so that images are evenly illuminated. The light can be adjusted by varying the exposure time and different calibration performed by using different camera gain settings.

The input image is used to construct a histogram of the pixel values that are observed for the given camera settings and background number of photons. This is then fit using the Poisson-Gamma-Gaussian probability mass function.

Ideally the input image should provide a minimum of 1000000 pixels, for example 16



frames of a 256x256 pixel image. This level of pixels is required to construct an even histogram that adequately samples the probability mass function. The pixels should have the same mean, i.e. a constant mean across the field of view. If it is not possible to achieve a constant mean across the field, for example in the instance of a gradient in the illumination, then the plugin will support rectangular ROI crops of the image. However the number of pixels should reach the minimum limit to construct a good histogram.

If the minimum pixel limit is not reached the plugin will log a warning but will continue to analyse the image.

### 10.5.3 Parameters

The following parameters can be configured:

| Parameter          | Description   |
|--------------------|---|
| BIAS ESTIMATE      | The initial estimate for the camera bias. The bias may be obtained from the camera manufacturer's specifications. A guess can be made by selecting the darkest part of the image, taking the mean and rounding (usually down) to the nearest hundred  |
| GAIN ESTIMATE      | The initial estimate for the total gain of the camera. The total gain may be obtained from the camera manufacturer's specifications. A good guess would be 25-50.   |
| NOISE ESTIMATE     | The initial estimate for the camera read noise. The read noise in electrons may be obtained from the camera manufacturer's specifications. This will have to be converted to ADUs by applying the camera gain (not the total gain). A good guess would be 3-10                                    |
| SHOW APPROXIMATION | Show on the final output plot a function that approximates the convolution of the Poisson-Gamma distribution with a Gaussian distribution.<br><br>This approximate PMF is used to model the EM-Gain when performing Maximum Likelihood Estimation fitting within the P <sub>PEAK</sub> FIT plugin |

Note that the plugin will remember the last values that were fitted for the bias, gain and noise estimates. Thus an initial guess can be used, the image analysed and then the plugin repeated with updates to the estimates if appropriate to refine the fit.

### 10.5.4 Simulation Mode

Instead of using an input image to create a histogram of pixel values, it is possible to simulate pixel values by generating a Poisson-Gamma-Gaussian random variable. To run the plugin in simulation mode hold down the SHIFT key when running the plugin. The following additional parameters will be available:

| Parameter | Description  |
|-----------|--|
| SIMULATE  | Check this box to simulate the histogram of pixel values |

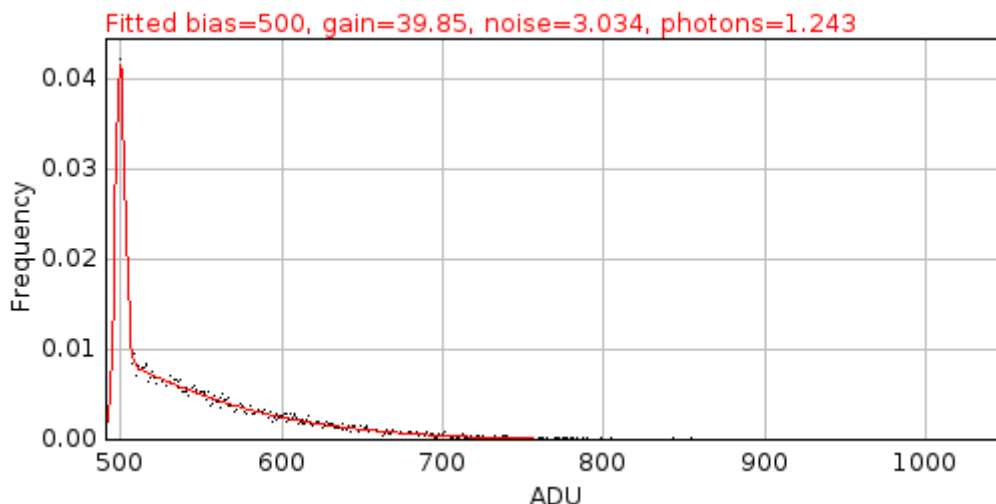
| Parameter  | Description   |
|------------|---|
| BIAS       | The camera bias for the simulation  |
| GAIN       | The total gain for the simulation   |
| NOISE      | The read noise for the simulation   |
| PHOTONS    | The average number of photons per pixel for the simulation  |
| SAMPLES    | The number of samples for the simulation  |
| SAMPLE PDF | <p>Check this to generate the Probability Mass Function (PMF) using the provided parameters. Then sample randomly from within this PMF.</p> <p>The default is to generate a random Poisson sample using the average photon number, then use this to generate a Gamma sample from the photon count and then generate a Gaussian sample from the amplified photon count</p> |

Simulation mode can be used to see if the fitting process is working given the expected parameters for bias, gain, noise and photons.

### 10.5.5 Results

The plugin will create a histogram of the pixel values and attempt to fit it using the Poisson-Gamma-Gaussian PMF. The estimated and fitted parameters are written to the ImageJ log.

The histogram of pixel values, fitted PMF and the fit parameters are shown on a plot as show below:



The values for the gain, bias and noise should be constant for different background photon levels. This can be evaluated using different input calibration images. The parameters can be used within the P<sub>EAK</sub> FIT plugin to perform Maximum Likelihood Estimation modelling the camera noise of the EM-CCD camera.

## 10.6 EM-Gain PMF

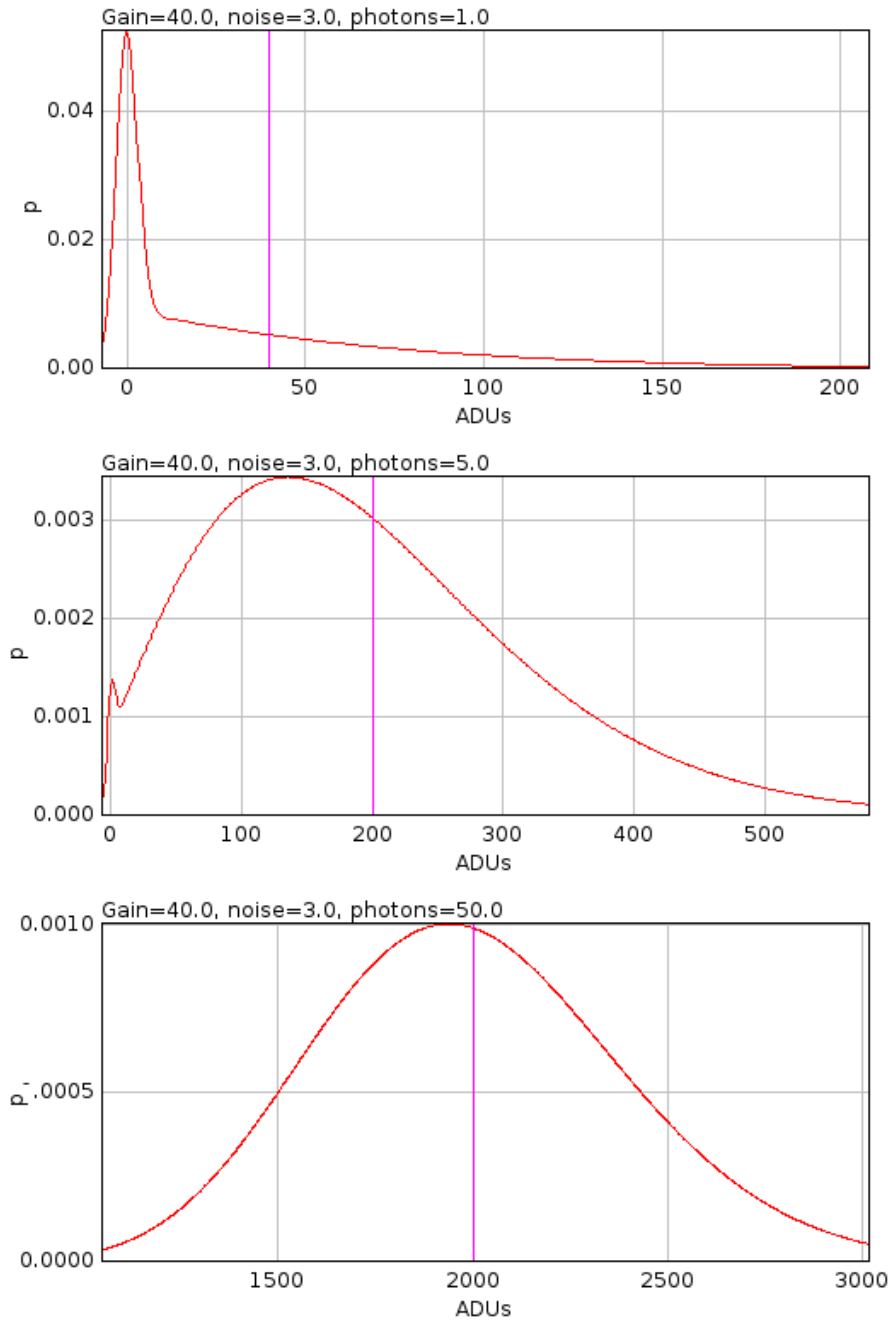
Displays a plot of the probability mass function (PMF) of the expected value of a pixel on an EM-CCD camera given an average number of photons. The form of the PMF is a convolution of a Poisson, Gamma and Gaussian distribution. See 10.5.1: EM-CCD Probability Model for more details.

The PMF is then approximated by analytically calculating the PMF of a Poisson-Gamma distribution and then approximating a convolution with a Gaussian distribution. The method for this approximation is taken from the supplementary Python software provided by Mortensen, et al (2010). They used this approximation when fitting the images of single fluorophores in TIRF (Total Internal Reflection Fluorescence) images taken with an EM-CCD camera. A second plot showing the difference between the real PMF and the approximation is displayed. This allows investigation of any situation where the approximation is not appropriate for modelling the PMF. It is rare for the approximation to differ by more than 1%.

The plugin has the following parameters:

| Parameter          | Description   |
|--------------------|---|
| GAIN               | The total gain for EM-CCD camera  |
| NOISE              | The camera read noise   |
| PHOTONS            | The average number of photons per pixel for the simulation  |
| SHOW APPROXIMATION | Show on the PMF plot the approximation function.<br><br>Note: This approximate PMF is used to model the EM-Gain when performing Maximum Likelihood Estimation fitting within the P <sub>EAK</sub> F <sub>IT</sub> plugin  |
| REMOVE HEAD        | Set a limit on the initial cumulative probability to remove from the plot. This allows removing the start of the curve where the convolution of the Poisson-Gamma distribution with the Gaussian is incomplete  |
| REMOVE TAIL        | Set a limit on the final cumulative probability to remove from the plot. This allows removing the tail of the curve where the convolution of the Poisson-Gamma distribution with the Gaussian is incomplete. It also allows removing the long tail which can take up a large amount of the plot width |
| RELATIVE DELTA     | Check this to show the difference between the actual PMF and the approximate PMF as a relative score. The default is absolute   |

Examples of the PMF are shown below. The magenta line on the plot shows the position of the average number of photons after the gain has been applied.



The PMF is skewed for low photons with a spike at  $c=0$  blurred by the Gaussian read noise. Increasing photon counts return a shape more characteristic of a Poisson distribution. For this reason it is possible to use a simple Poisson model for the camera noise when performing Maximum Likelihood Estimation, i.e. ignoring the effect of EM-gain and read noise, if the number of photons within the localisation is large. This is an option available within the `PEAK FIT` plugin and allows much faster fitting since the Poisson PMF (a) can be evaluated much faster than the Poisson-Gamma-Gaussian PMF; and (b) has an analytical derivative allowing derivative based fitting methods.

## 10.7 Diffusion Rate Test

The DIFFUSION RATE TEST plugin will simulate molecule diffusion and fit a graph of mean-squared displacement to determine the diffusion coefficient. This is a test plugin to show that the simulated diffusion in the CREATE DATA plugin generates correct moving particles.

When a molecule is diffusing it can move in any direction. The total distance it moves and the track it took may not be visible due to the speed of movement. However the diffusion of particles in a single dimension can be modelled as a population. If the squared distances from the origin after a set time are plotted as a histogram they can be modelled using a Gaussian curve. The average distance the particles will move is zero and the variance of the Gaussian curve will be the mean-squared displacement (MSD). This can be expressed by unit time. The MSD is proportional to the diffusion coefficient ( $D$ ). The relationship for a single-dimension is  $MSD = 2D$ . This increases to  $4D$  and  $6D$  for two and three dimensional distances (since the diffusion in each dimension is independent).

### 10.7.1 Grid Walk simulation

Since the MSD in a single dimension is equal to  $2D$ , the mean-distance a particle moves will be  $\sqrt{2D}$ . This step size can be used to simulate diffusion using a grid walk. At each step a particle can move forward or backwards by the step size. If the direction is random then the population of particles will have an average displacement of zero, a mean displacement of the step size and a MSD of step size squared =  $2D$ . Multi-dimension diffusion is done by simulating the movement in each dimension separately.

### 10.7.2 Random Move simulation

Diffusion can also be simulated by moving particles on a random vector. The distance moved should be sampled from a Gaussian distribution with a mean of zero and a standard deviation of  $\sqrt{MSD}$ . This is  $2D$ ,  $4D$  or  $6D$  for 1, 2 or 3 dimensions respectively.

However the unit vector must be directed in a random orientation. For one dimension this is either forward or backward. For higher dimensions a random vector can be produced by sampling the movement in each dimension from a Gaussian distribution with mean zero and standard deviation 1. This vector is normalised to unit length.

The generation of the unit vector and the movement distance can be combined into a single stage. The random displacement is produced by sampling each dimension from a Gaussian distribution with mean zero and standard deviation of  $\sqrt{2D}$ . This is the equivalent of 1D diffusion in 3 independent dimensions.

### 10.7.3 Confined Diffusion

Particles may not be able to freely move in any direction, for example when they collide with a barrier. The DIFFUSION RATE TEST plugin allows particles to be confined in a sphere. In this case the diffusion step is calculated and if the step would move the particle outside the sphere the move is rejected. Attempts are made to move the particle a set number of times until successful otherwise the particle coordinates are not updated. This simulation produces good results when the average step size is at least an order of magnitude less than the sphere radius (so allowing many steps inside the sphere to be valid) and the RANDOM MOVE simulation is used.

### 10.7.4 Analysis

The DIFFUSION RATE TEST plugin simulates the random diffusion of many particles over a period of time. Each diffusion path is then analysed. The plugin has the following parameters:

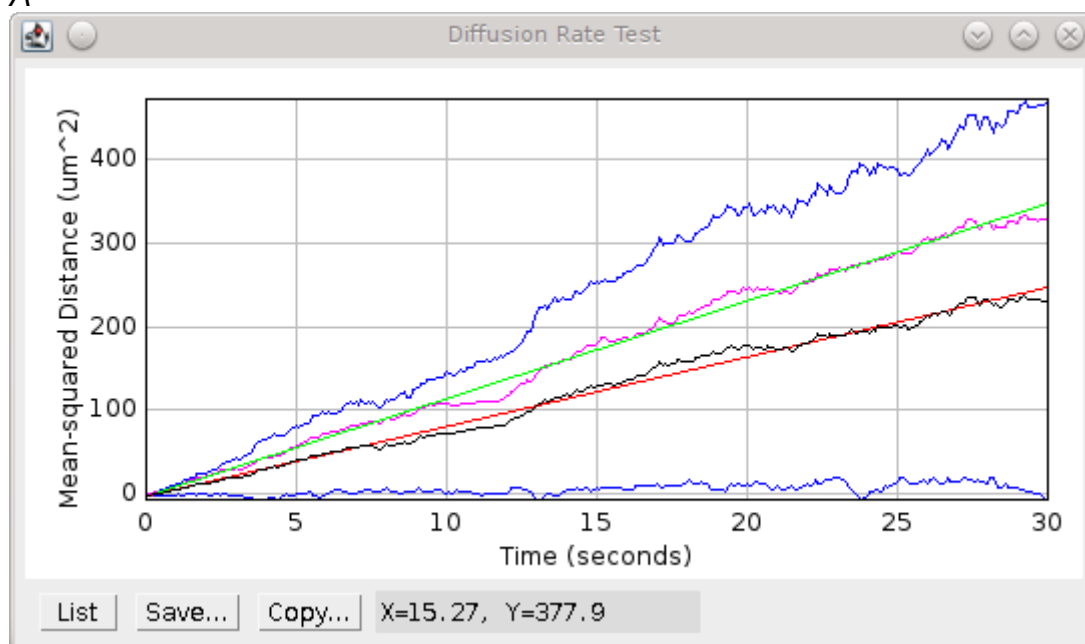
| Parameters           | Description  |
|----------------------|--|
| PIXEL PITCH (NM)     | The pixel size for the simulation  |
| SECONDS              | The duration of the simulation   |
| STEPS PER SECOND     | The number of diffusion steps the particle makes per second  |
| PARTICLES            | The number of particles to simulate  |
| DIFFUSION RATE       | The diffusion coefficient (D)  |
| USE GRID WALK        | If true then simulate diffusion using a grid walk, otherwise use a random move. The grid walk simulation is approximately 3 times faster |
| USE CONFINEMENT      | If true then use a sphere to confine the particle movement   |
| CONFINEMENT ATTEMPTS | The number of times to attempt a confined move   |
| CONFINEMENT RADIUS   | The radius of the confinement sphere   |
| FIT N                | When using confined diffusion only fit the first N points of the MSD plot  |
| SHOW EXAMPLE         | Show an example image of a diffusion path  |
| MAGNIFICATION        | The magnification of the example image. The pixels will represent (pixel pitch) / magnification nanometres                               |

### 10.7.5 Output

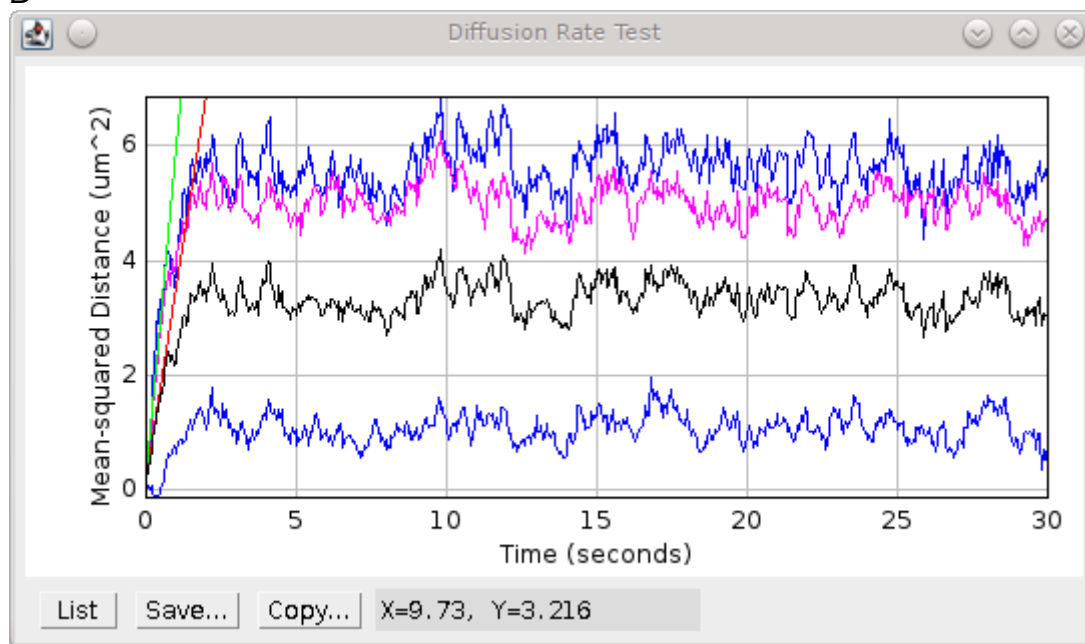
#### 10.7.5.1 MSD plot

For each particle the plugin will compute the squared displacement from the origin over the time course of the simulation. Distance are computed in 2D and 3D. The plugin will plot the mean-squared distance against time for the population as shown in Illustration 41A. The 2D and 3D MSD data are then fit using a linear regression. The gradient of the fit can be used to calculate the diffusion coefficient by dividing by 4 or 6 respectively.

A



B



*Illustration 41: Mean-squared displacement chart produced by the Diffusion Rate Test plugin for (A) random diffusion and (B) confined diffusion. Black: 2D MSD; Red: 2D MSD fitted; Blue: 2D MSD  $\pm$  Standard Deviation; Magenta: 3D MSD; Green: 3D MSD fitted.*

If confined diffusion is performed the MSD will reach a natural upper limit. This will result in a plateau of the MSD plot as shown in Illustration 41B. In this case only the initial diffusion of the particles will be unconstrained. The analysis should therefore fit the initial linear section of the MSD plot. If the confinement radius is too small there may be no linear section to the MSD curve.

**Note:**

The asymptote of the curve for confined diffusion should be defined by the average distance to the centre of a random distribution of particles within a sphere. This can be computed using the distance from the centre of all the points in a sphere divided by the number of points in a sphere. The surface area (SA) of a sphere is equal to the number of points at distance  $r$  from the centre. So  $SA \times r$  is the sum of the distances of points at distance  $r$  from the centre. If this is integrated from zero to  $R$  it produces the sum of all distances from any point within a sphere of radius  $R$ . The number of points is the volume ( $V$ ) of the sphere. Therefore:

$$\frac{\int_0^R SA \times r}{V} = \frac{\int_0^R 4\pi r^2 \times r}{4\pi R^3/3} = \frac{\int_0^R 4\pi r^3}{4\pi R^3/3} = \frac{\pi R^4}{4\pi R^3/3} = \frac{3R}{4}$$

Thus the mean-distance to the centre for particles in a sphere is  $3/4 R$ . This can be used to check that the confined simulation is performing as a true random diffusion within a sphere.

**10.7.5.2 Diffusion example**

If the `SHOW EXAMPLE` option was selected the plugin will show an image of the track of a single particle. The track is shown on a black background. The track is initialised at a value of 32 and ends with a value of 255. The movement can thus be followed using a color Lookup table (LUT), e.g. `IMAGE > LOOKUP TABLES > FIRE`.

The plugin will also show a plot of the displacement of the particle over time. The red line shows the X displacement and the blue shows the Y displacement.

**10.7.5.3 Analysis results**

The fitting analysis results are output to the ImageJ log window, e.g.

```
Diffusion Rate Test : D = 1.0 um^2/sec, Precision = 0.0 nm
```

```
Mean-displacement per dimension = 1414.0 nm/sec
```

```
Simulation step-size = 44.72 nm
```

```
Raw data D=1.0 um^2/s, Precision = 0.0 nm, N=22000, step=0.001 s, mean=0.004034 um^2, MSD = 4.034 um^2/s
```

```
2D Diffusion rate = 1.013 um^2 / sec (50.22 ms)
```

```
3D Diffusion rate = 1.175 um^2 / sec (50.22 ms)
```

The input diffusion coefficient is shown for reference, the units are  $\mu\text{m}^2/\text{sec}$ . This is converted to the expected mean-displacement in nm per second and the simulation step size (in nm). This will allow the user to experiment with the radius of the confinement sphere and the number of simulation steps. Remember that the step size should be less than the sphere radius when using confinement. The fitted diffusion coefficients from the 2D and 3D fitting are then shown. These should be close to the input diffusion rate.

If the simulation was performed using confinement then the final distance to the origin for each particle will be saved. The average distance will be shown along with the expected asymptote distance, i.e. the mean-distance to the centre of a sphere, which is calculated as  $3/4$  of the confinement radius, e.g.

```
3D asymptote distance = 702.7 nm (expected 750.00)
```



#### 10.7.5.4 Memory Results

The coordinates of each diffusing particle, starting at the origin (0,0), are saved to a results dataset in memory. Each consecutive step of the same particle is given a new frame and particles are allocated a unique ID. The current frame is incremented between particles so that each particle track is separated in time. This allows the results set to be used within the `TRACE DIFFUSION` and `DRAW CLUSTERS` plugins to verify their functionality.

#### 10.7.6 Extra options

Hold the Shift key down when running the plugin to activate extra options. The following options are available and are described in the following sections:

| Parameters      | Description   |
|-----------------|---|
| AGGREGATE STEPS | Create a second dataset by averaging N consecutive positions into a single location   |
| MSD ANALYSIS    | Specify the maximum number of steps used to perform MSD analysis. This is only relevant when the AGGREGATE STEPS parameter is above 1 |
| PRECISION       | Specify the localisation precision of positions   |

#### 10.7.7 AGGREGATE STEPS parameter

The standard plugin simulates diffusion in small steps. These can be aggregated together to simulate the position of the particle in a frame taken on a camera. In this case the average position of a set of consecutive steps is calculated to aggregated the position into a frame. The mean-squared distance between frames is then reported to the ImageJ log:

Raw data  $D=1.0 \text{ um}^2/\text{s}$ , Precision = 0.0 nm,  $N=22000$ , step=0.001 s, mean=0.004016  $\text{um}^2$ , MSD = 4.016  $\text{um}^2/\text{s}$

Aggregated data  $D=1.0 \text{ um}^2/\text{s}$ , Precision=0.0 nm,  $N=200$ , step=0.1 s, mean=0.2713  $\text{um}^2$ , MSD = 2.713  $\text{um}^2/\text{s}$

Note that the aggregation has the effect of reducing the mean-squared displacement for the dataset.

The aggregated data is saved into a dataset in memory.

#### 10.7.8 MSD ANALYSIS parameter

The MSD ANALYSIS option is available when data aggregation has been performed using the AGGREGATE STEPS parameter. When multiple small steps are aggregated into single coordinates this causes the observed MSD to be lower than the expected MSD given the diffusion coefficient. Effectively the averaging of the position of a particle within a frame has caused loss of information about the diffusion distance covered within that frame. MSD analysis allows the effect of aggregation to be analysed.

For each simulated track the position of the particle is computed as a rolling average of the coordinates using the configured AGGREGATED STEPS (N). For example the first position is the average of the first N steps. The next position is computed by adding the N+1 coordinate

and subtracting the 1<sup>st</sup> coordinate from the sum to create a new average.

Using the rolling average position the squared distance between each position and the  $j^{\text{th}}$  position along the sequence is computed for all  $j$  up to the limit set by the `MSD_ANALYSIS` parameter. The mean squared distance is then reported for each separation  $j$  to a summary table:

| Column                 | Description   |
|------------------------|---|
| D                      | The input diffusion coefficient   |
| P <sub>PRECISION</sub> | The precision error of each average position  |
| D <sub>SIM</sub>       | The simulated diffusion coefficient (calculated using the mean-squared displacement between non-aggregated coordinates) |
| STEP                   | The size of a single simulation step in seconds   |
| RESOLUTION             | The number of steps that are aggregated into a frame  |
| FRAME                  | The frame length in seconds   |
| T                      | The separation $j$ between the two rolling average positions in seconds   |
| N                      | The separation $j$ between the two rolling average positions in frames  |
| N                      | The number of samples to compute the MSD  |
| MSD                    | The mean-squared displacement (MSD)   |
| D                      | The observed diffusion coefficient (calculated as $\text{MSD} / 4t$ )   |

Note that if the `PPRECISION` parameter is used the positions will be adjusted with a random error before the distances are computed. This is computationally expensive and the simulation takes longer.

### 10.7.9 `PPRECISION` parameter

By default the exact coordinates of a particle are used in the analysis and to create the output datasets. To simulate the results generated by a super-resolution image reconstruction method the coordinates can be reported with a random error added to each position. The error simulates the fitting precision of the super-resolution localisation method. Error is added independently to the X and Y coordinates using a Gaussian random variable with the given standard deviation.

The precision error has the effect of increasing the mean-squared displacement for the dataset, e.g.

```
Raw data D=1.0 um^2/s, Precision = 0.0 nm, N=22000, step=0.001 s,
mean=0.004016 um^2, MSD = 4.016 um^2/s
```

...

Raw data  $D=1.0 \text{ } \mu\text{m}^2/\text{s}$ , Precision = **30.0 nm**,  $N=22000$ , step= $0.001 \text{ s}$ , mean= $0.007614 \text{ } \mu\text{m}^2$ , MSD = **7.614  $\mu\text{m}^2/\text{s}$**

## 10.8 Trace Diffusion

The TRACE DIFFUSION plugin will trace molecules through consecutive frames and then perform mean-squared displacement analysis to calculate a diffusion coefficient.

The plugin is similar to the DIFFUSION RATE TEST plugin however instead of simulating particle diffusion the plugin will use an existing results set. This allows the analysis to be applied to results from fitting single-molecule images using the PEAK FIT plugin.

### 10.8.1 Analysis

The plugin runs a tracing algorithm on the results to find localisations that occur in consecutive frames. Details of the tracing algorithm can be found in section 7.2: Trace Molecules. The distance threshold for the tracing algorithm can be specified but the time threshold is set to 1 frame, i.e. only continuous tracks will be extracted. Thus a pair of localisations within adjacent frames will be connected if they are within the distance threshold. In addition the plugin allows the track to be excluded if a second localisation occurs within an exclusion threshold of the first localisation. This effectively removes traces of particles that could overlap with another moving particle.

Once the tracks have been identified the tracks are filtered using a length criteria and shorter tracks discarded. Optionally the tracks can be truncated to the minimum length which ensures even sampling of particles with different track lengths. The plugin computes the mean-squared distance of each point from the origin. Optionally the plugin computes the mean-squared distance of each point from every other point in the track. These internal distances increase the number of points in the analysis. Therefore if the track is not truncated the number of internal distances at a given time separation is proportional to the track length. To prevent bias in the data towards the longer tracks the average distance for each time separation is computed per track and these are used in the population statistics. Thus each track contributes only once to the mean-displacement for a set time separation.

The Mean-squared distance (MSD) per molecule is calculated using two methods. The ALL-VS-ALL method uses the sum of squared distances divided by the sum of time separation between points. The value includes the all-vs-all internal distances (if selected). The ADJACENT method uses the average of the squared distances between adjacent frames divided by the time delta ( $\Delta t$ ) between frames. The MSD values are expressed in  $\mu\text{m}^2/\text{second}$  and can be saved to file or shown in a histogram.

The average mean-squared distances for all the traces are plotted against the time separation and a best fit line is calculated. The Mean-squared distances are proportional to the diffusion coefficient (D):

$$MSD(n \Delta t) = 4 D n \Delta t + 4 \sigma^2$$

where  $n$  is the number of separating frames,  $\Delta t$  is the time lag between frames, and  $\sigma$  is the localisation precision. Thus the gradient of the best fit line can be used to obtain the diffusion coefficient. Note that the plugin will compute a fit with and without an explicit intercept and pick the solution with the best fit to the data (see 10.8.6 Selecting the best fit). Note that an additional best fit line can be computed using a MSD correction factor

(see 10.8.3 MSD Correction).

### 10.8.1.1 Apparent Diffusion Coefficient

Given that the localisations within each trace are subject to a fitting error, or precision ( $\sigma$ ), the apparent diffusion coefficient ( $D^*$ ) can be calculated accounting for precision [Uphoff *et al*, 2013]:

$$D = \max\left(0, \frac{MSD}{4n\Delta t} - \frac{\sigma_{loc}^2}{n\Delta t}\right)$$

The plugin thus computes the average precision for the localisations included in the analysis can optionally report the apparent diffusion coefficient ( $D^*$ ). If the average precision is above 100nm then the plugin prompts the user to confirm the precision value.

## 10.8.2 Jump Distance analysis

The jump distance is how far a particle moves in given time period. Analysis of a population of jump distances can be used to determine if the population contains molecules diffusing with one or more diffusion coefficients [Weimann *et al*, 2013]. For two dimensional Brownian motion the probability that a particle starting at the origin will be encountered within a shell of radius  $r$  and a width  $dr$  at time  $\Delta t$  is given by:

$$p(r^2, \Delta t) dr^2 = \frac{1}{4D\Delta t} e^{-\frac{r^2}{4D\Delta t}} dr^2$$

This can be expanded to a mixed population of  $m$  species where each fraction ( $f_j$ ) has a diffusion coefficient  $D_j$ :

$$p(r^2, \Delta t) dr^2 = \sum_{j=1}^m \frac{f_j}{4D_j\Delta t} e^{-\frac{r^2}{4D_j\Delta t}} dr^2$$

For the purposes of fitting the integrated distribution can be used. For a single population this is given by:

$$P(r^2, \Delta t) = \int_0^{r^2} p(r^2) dr^2 = 1 - e^{-\frac{r^2}{4D\Delta t}}$$

The advantage of the integrated distribution is that specific histogram bin sizes are not required to construct the cumulative histogram from the raw data. Note that the integration holds for a mixed population of  $m$  species where each fraction ( $f_j$ ) has a diffusion coefficient  $D_j$ :

$$P(r^2, \Delta t) = 1 - \sum_{j=1}^m f_j e^{-\frac{r^2}{4D_j\Delta t}}$$

Weimann *et al* (2013) show that fitting of the cumulative histogram of jump distances can accurately reproduce the diffusion coefficient in single molecule simulations. The performance is measured relative to the average distance a particle travels in the chosen time ( $d$ ) and the average localisation precision ( $\sigma$ ) expressed as a ratio  $\beta = d/\sigma$ . When  $\beta$  is above 6 then jump distance analysis reproduces the diffusion coefficient as accurately as MSD analysis for single populations. For mixed populations of moving and stationary

particles the MSD analysis fails and the jump distance analysis yields accurate values when  $\beta$  is above 6.

The `TRACE DIFFUSION` plugin performs jump distance analysis using the jumps between frames that are  $n$  frames apart. The distances may be from the origin to the  $n$ th frame or may use all the available internal distances  $n$  frames apart. A cumulative histogram is produced of the jump distance. This is then fitted using a single population and then for mixed populations of  $j$  species by minimising the sum-of-squared residuals (SS) between the observed and expected curves. Alternatively the plugin can fit the jump distances directly without using a cumulative histogram. In this case the probability of each jump distance is computed using the formula for  $p(r^2, \Delta t)$  and the combined probability (likelihood) of the data given the model is computed. The best model fit is achieved by maximising the likelihood (MLE – Maximum Likelihood Estimation).

When fitting multiple species the fit is rejected if: (a) the relative difference between coefficients is smaller than a given factor; or (b) the minimum fraction,  $f_j$ , is less than a configured level. The result must then be compared to the previous result to determine if increasing the number of parameters has improved the fit (see 10.8.6 Selecting the best fit).

Optimisation is performed using a fast search to maximise the score by varying each parameter in turn (Powell optimiser). In most cases this achieves convergence. However in the case that the default algorithm fails then a second algorithm is used that uses a directed random walk (CMAES optimiser). This algorithm depends on randomness and so can benefit from restarts. The plugin allows the number of restarts to be varied. For the optimisation of the sum-of-squares against the cumulative histogram a least-squares fitting algorithm (Levenberg-Marquardt or LVM optimiser) is used to improve the initial score where possible. The plugin will log messages on the success of the optimisers to the `IMAGEJ` log window. Extra information will be logged if using the `DEBUG FITTING` option.

### 10.8.3 MSD Correction

This corrects for the diffusion distance lost in the first and last frames of the track due to the representation of diffusion over the entire frame as an average coordinate. A full explanation of the correction is provided in Appendix C : MSD Correction.

The observed MSD can be converted to the true MSD by dividing by a correction factor ( $F$ ):

$$F = \frac{n-1/3}{n}$$

Where  $n$  is the number of frames over which the jump distance is measured (i.e. end - start).

When performing jump distance analysis it is not necessary to correct each observed squared distance before fitting. Since the correction is a single scaling factor instead the computed diffusion coefficient can be adjusted by applying the correction factor after fitting. This allows the plugin to save the raw data to file and use for display on results plots.

If the `MSD CORRECTION` option is selected the plugin will compute the corrected diffusion coefficient as:

$$D_{corr} = D \cdot \frac{n}{n-1/3}$$

### 10.8.3.1 Fitting the plot of MSD verses N frames

When fitting the linear plot of MSD verses the number of frames the correction factor can be included. The observed MSD is composed of the actual MSD multiplied by the correction factor before being adjusted for the precision error:

$$oMSD(n\Delta t) = 4D(n\Delta t) - \frac{4D(\Delta t)}{3} + 4\sigma^2$$

This is still a linear fit with a new representation for the intercept that allows the intercept to be negative. To ensure the intercept is correctly bounded it should be represented using the fit parameters and not just fit using a single constant C.

When performing the linear fit of the MSD verses jump distance plot, 3 equations are fitted and the results with the best information criterion is selected. The results of each fit are written to the `IMAGEJ` log. The following equations are fit:

Linear fit:

$$oMSD(n\Delta t) = 4D(n\Delta t)$$

Linear fit with intercept:

$$oMSD(n\Delta t) = 4D(n\Delta t) + 4\sigma^2$$

Linear fit with MSD corrected intercept:

$$oMSD(n\Delta t) = 4D(n\Delta t) - \frac{4D(\Delta t)}{3} + 4\sigma^2$$

Note: In each model the linear gradient is proportional to the diffusion coefficient.

### 10.8.4 Precision Correction

Given that the localisations within each trace are subject to a fitting error, or precision ( $\sigma$ ), the apparent diffusion coefficient ( $D^*$ ) can be calculated accounting for precision [Uphoff *et al*, 2013]:

$$D_{app} = \max\left(0, \frac{MSD}{4n\Delta t} - \frac{\sigma_{loc}^2}{n\Delta t}\right)$$

If the `PRECISION CORRECTION` option is selected the plugin will subtract the precision and report the apparent diffusion coefficient ( $D^*$ ) from the jump distance analysis.

### 10.8.5 MSD and Precision correction

Both the `MSD CORRECTION` and `PRECISION CORRECTION` can be applied to the fitted MSD to compute the corrected diffusion coefficient:

$$D = \frac{n}{n-1/3} \cdot \max\left(0, \frac{MSD}{4n\Delta t} - \frac{\sigma_{loc}^2}{n\Delta t}\right)$$

### 10.8.6 Selecting the best fit

The Bias Corrected Akaike Information Criterion (cAIC) [Hurvich & Tsai, 1989] is calculated for the fit using the log likelihood ( $L$ ), the number of data points ( $n$ ) and the number of parameters ( $p$ ):

$$AIC = 2p - 2L$$

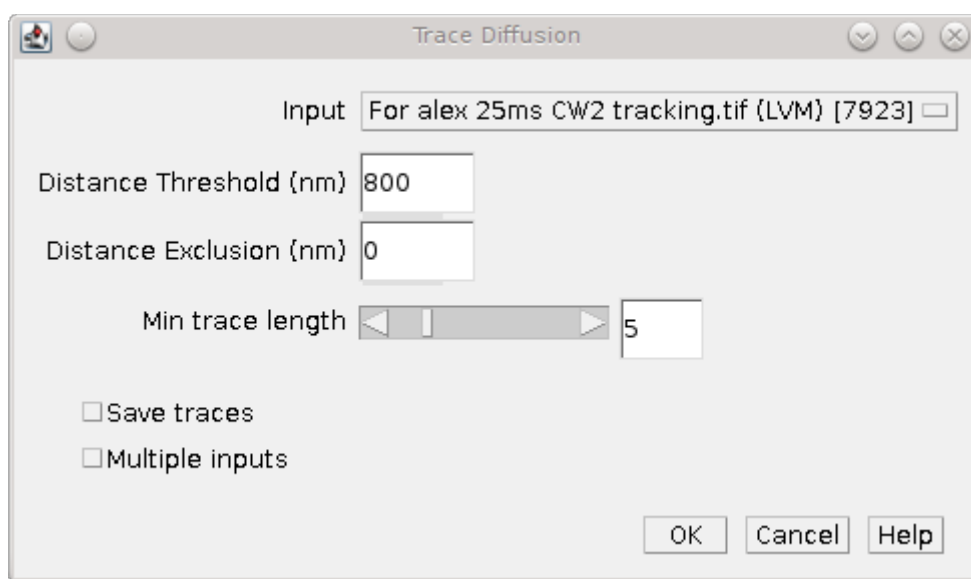
$$cAIC = AIC + (2p(p+1)/(n-p-1))$$

The corrected AIC penalises additional parameters. The model with the lowest AIC is preferred. If a higher AIC is obtained then increasing the number of fitted species in the mixed population has not improved the fit and so fitting is stopped. Note that when performing Maximum Likelihood Estimation the Log Likelihood ( $L$ ) is already known and is used directly to calculate the corrected AIC. When fitting the sum-of-squared residuals (SS) the Log Likelihood can be computed as:

$$L = 0.5 * (-n(\ln(2\pi) + 1 - \ln(n) + \ln(SS)))$$

### 10.8.7 Parameters

The plugin dialogue allowing the data to be selected is shown in below:



The plugin has the following parameters:

| Parameters         | Description   |
|--------------------|---|
| INPUT              | Specify the input results set   |
| DISTANCE THRESHOLD | The distance threshold for tracing  |
| DISTANCE EXCLUSION | The exclusion distance. If a particle is within the distance threshold but a second particle is within the exclusion distance then the trace is discarded (due to overlapping tracks) |
| MIN TRACE LENGTH   | The minimum length for a track (in time frames)   |

|             |  |
|-------------|--|
| IGNORE ENDS | <p>Ignore the end jumps in the track.</p> <p>If a fluorophore activated only part way through the first frame and bleaches only part way through the last frame the end jumps represent a shorter time-span than the frame interval. These jumps can optionally be ignored.</p> <p>This option requires tracks to be 2 frames longer than the <math>M_{IN}</math> TRACE LENGTH parameter</p> |
| SAVE TRACES | Save the traces to file in the P <sub>EAK</sub> FIT results format   |

When all the datasets have been traced the plugin presents a second dialogue to configure the diffusion analysis. The following parameters can be configured:

| Parameters           | Description  |
|----------------------|--|
| TRUNCATE TRACES      | Set to true to only use the first N points specified by the $M_{IN}$ TRACE LENGTH parameter  |
| INTERNAL DISTANCES   | Compute the all-vs-all distances. Otherwise only compute distance from the origin  |
| FIT LENGTH           | Fit the first N points with a linear regression  |
| MSD CORRECTION       | <p>Perform Mean Square Distance (MSD) correction.</p> <p>This corrects for the diffusion distance lost in the first and last frames of the track due to the representation of diffusion over the entire frame as an average coordinate</p>   |
| PRECISION CORRECTION | <p>Correct the fitted diffusion coefficient using the average precision of the localisations.</p> <p>Note that uncertainty in the position of localisations (fit precision) will contribute to the displacement between localisations. This can be corrected for by subtracting <math>4s^2</math> from the measured squared distances with <math>s</math> the average precision of the localisations</p> |
| MAXIMUM LIKELIHOOD   | Perform jump distance fitting using Maximum Likelihood Estimation (MLE). The default is sum-of-squared residuals (SS) fitting of the cumulative histogram of jump distances  |
| FIT RESTARTS         | <p>The number of restarts to attempt when fitting using the CMAES optimiser. A higher number produces a more robust fit solution since the best fit of all the restarts is selected.</p> <p>Note that the CMAES optimiser is only used when the default Powell optimiser fails to converge</p>   |
| JUMP DISTANCE        | The distance between frames to use for jump analysis   |

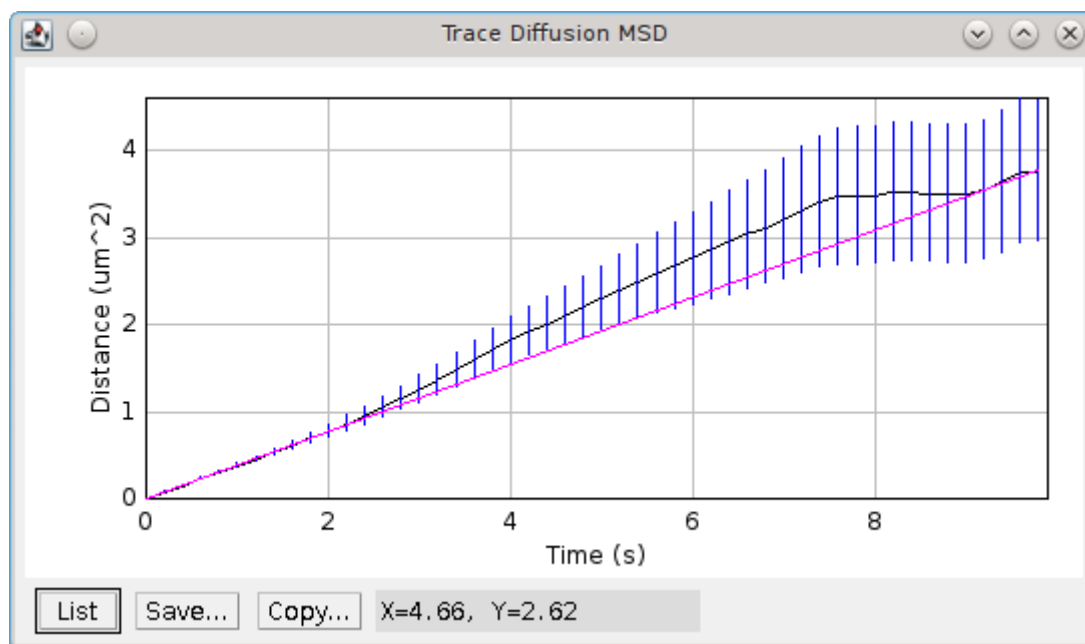


|                      |   |
|----------------------|---|
| MINIMUM DIFFERENCE   | The minimum relative difference (ratio) between fitted diffusion coefficients to accept the model. The difference is calculated by ranking the coefficient in descending order and then expressing successive pairs as a ratio. Models with coefficients too similar are rejected |
| MINIMUM FRACTION     | The minimum fraction of the population that each species must satisfy. Models with species fractions below this are rejected  |
| MINIMUM <i>N</i>     | The minimum number of species to fit. This can be used to force fitting with a set number of species.<br><br>This extra option is only available if the plugin is run with the <b>SHIFT</b> key held down, otherwise the default is 1   |
| MAXIMUM <i>N</i>     | The maximum number of species to fit. In practice this number may not be achieved if adding more species does not improve the fit   |
| DEBUG FITTING        | Output extra information to the <b>IMAGEJ</b> log window about the fitting process  |
| SAVE TRACE DISTANCES | Save the traces to file. The file contains the per-molecule MSD and $D^*$ and the squared distance to the origin for each trace   |
| SAVE RAW DATA        | Select this to select a results directory where the raw data will be saved. This is the data that is used to produce all the histograms and output plots  |
| SHOW HISTOGRAMS      | Show histograms of the trace data. If selected a second dialogue is presented allowing the histograms to be chosen and the number of histogram bins to be configured  |
| TITLE                | A title to add to the results table   |

## 10.8.8 Output

### 10.8.8.1 MSD verses time

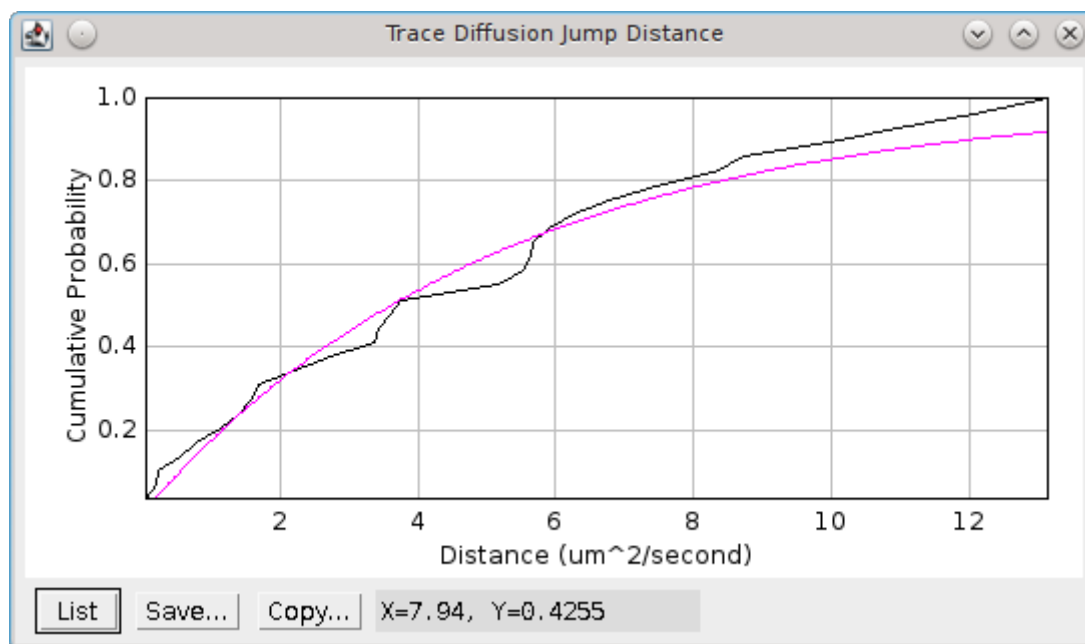
The plugin will plot the mean-squared distances against the time as show in Illustration 42. The plot shows the best fit line. If the data is not linear then the diffusion of particles may be confined, for example by cellular structures when using *in vivo* image data. In this case the diffusion coefficient will be underestimated.



*Illustration 42: Plot of mean-squared distance verses time produced by the Trace Diffusion plugin. The mean of the raw data is plotted with bars representing standard error of the mean. The best fit line is shown in magenta.*

#### 10.8.8.2 Jump distance histogram

The plugin produces a cumulative probability histogram of the jump distance (see Illustration 43). The best fit for a single species model will be shown in magenta. Any significant deviations of the histogram line from the single species fit are indicative of a multi-species population. If a multiple species model has a better fit than the single species model then it will be plotted in yellow.



*Illustration 43: Jump distance cumulative probability histogram. The best fit for the single species model is shown in magenta.*

### 10.8.8.3 Histograms

If the **SHOW HISTOGRAMS** option is selected the plugin presents a second dialog where the histograms can be configured. The number of bins in the histogram can be specified and outliers can optionally be removed. Outliers are any point more than 1.5 times the interquartile range above or below the upper and lower quartile boundaries. The following histograms can be chosen:

| Parameters       | Description  |
|------------------|--|
| TOTAL SIGNAL     | The total signal of each trace   |
| SIGNAL-PER-FRAME | The signal-per-frame of the localisations in a trace   |
| T-ON             | The on-time of a trace. This excludes the traces too short to be analysed  |
| MSD/MOLECULE     | The average mean-squared distance per molecule. Plots of the all-vs-all and adjacent MSD are shown.<br><br>If the particles contain molecules moving with different diffusion rates or a fixed fraction of molecules then the histogram may be multi-modal |
| D*/MOLECULE      | The apparent diffusion coefficient per molecule. Plots of the all-vs-all and adjacent D* are shown.  |

### 10.8.8.4 Summary table

The plugin shows a summary table of the analysis results. This allows the plugin to be run

with many different settings to view the effect on the calculated diffusion coefficient. The following columns are reported:

| Column        | Description   |
|---------------|---|
| TITLE         | The title (specified by the TITLE parameter)  |
| DATASET       | The input dataset   |
| EXPOSURE TIME | The dataset exposure time per frame   |
| D-THRESHOLD   | The distance threshold  |
| EX-THRESHOLD  | The exclusion distance  |
| MIN-LENGTH    | The minimum track length that was analysed  |
| IGNORE ENDS   | True if the end jumps of tracks were ignored  |
| TRUNCATE      | True if tracks were truncated to the min length   |
| INTERNAL      | True if internal distance were used   |
| FIT LENGTH    | The number of points fitted in the linear regression  |
| MSD CORR      | True if MSD correction was applied  |
| S CORR        | True if precision correction was applied  |
| MLE           | True if Maximum Likelihood fitting was used   |
| TRACES        | The number of traces analysed   |
| S             | The average precision of the localisations in the traces  |
| D             | The diffusion coefficient from MSD linear fitting   |
| FIT S         | The fitted precision when fitting an intercept in the MSD linear fit  |
| JUMP DISTANCE | The time distance used for jump analysis  |
| N             | The number of jumps for jump distance analysis  |
| BETA          | The beta parameter which is the ratio between the mean squared distance the localisation precision: $MSD / s^2$<br>A beta above 6 indicates that jump distance analysis will produce reliable results [Weimann <i>et al</i> , 2013] |
| JUMP D        | The diffusion coefficient(s) from jump analysis   |
| FRACTIONS     | The fractions of each population from jump analysis   |

|              |   |
|--------------|---|
| IC           | The information criterion (IC) for the best model fit.<br><br>Note that the IC is not comparable between the MLE or LSQ methods for fitting. It is also not comparable when the number of jumps is different. It can only be used to compare fitting the same jump distances with a different number of mobile species. This can be controlled using the MINIMUM and MAXIMUM N parameters |
| TOTAL SIGNAL | The average total signal of each trace  |
| SIGNAL/FRAME | The average signal-per-frame of the localisations in a trace  |
| T-ON         | The average on-time of a trace. This excludes the traces too short to be analysed   |

The plugin will report the number of traces that were excluded using the length criteria and the fitting results to the IMAGEJ log. This includes details of the jump analysis with the fitting results for each model and the information criterion used to assess the best model, e.g.

783 Traces filtered to 117 using minimum length 5

Linear fit (5 points) : Gradient = 2.096, D = 0.5239  $\mu\text{m}^2/\text{s}$ , SS = 0.047595 (2 evaluations)

Jump Distance analysis : N = 151, Time = 6 frames (0.6 seconds). Mean Distance = 1371.0 nm, Precision = 38.55 nm, Beta = 35.57

Estimated D = 0.4698  $\mu\text{m}^2/\text{s}$

Fit Jump distance (N=1) : D = 0.0498  $\mu\text{m}^2/\text{s}$ , SS = 0.433899, IC = -453.1 (12 evaluations)

Fit Jump distance (N=2) : D = 1.655, 0.0346  $\mu\text{m}^2/\text{s}$  (0.1832, 0.8168), SS = 0.014680, IC = -960.3 (342 evaluations)

Fit Jump distance (N=3) : D = 1.655, 0.0346, 0.0346  $\mu\text{m}^2/\text{s}$  (0.1832, 0.1204, 0.6964), SS = 0.014680, IC = -956.1 (407 evaluations)

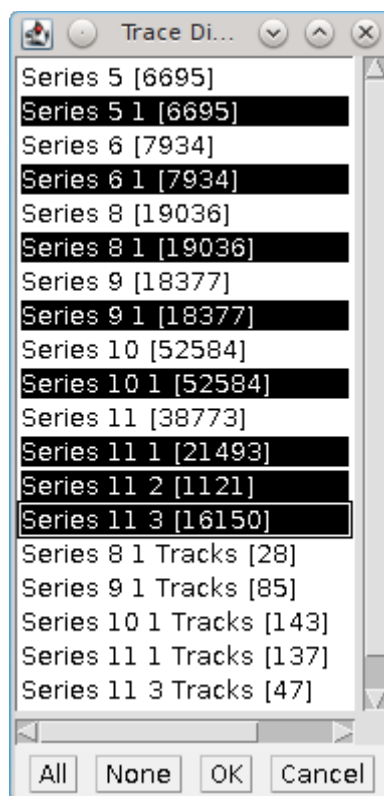
Coefficients are not different: 0.0346 / 0.0346 = 1.0

Best fit achieved using 2 populations: D = 1.655, 0.0346  $\mu\text{m}^2/\text{s}$ , Fractions = 0.1832, 0.8168

## 10.9 Trace Diffusion (Multi)

This plugin allows the TRACE DIFFUSION plugin to be run with multiple input datasets. Each dataset will be traced separately. The results are then combined for analysis. This allows analysis of multiple repeat experiments as if one single dataset.

When the plugin runs a dialog is presented that allows the datasets to be selected:



Click a single result set to select or deselect.

Hold the **SHIFT** key to select or deselect a range of results starting from the last clicked result set.

Use the **ALL** or **NONE** buttons to select or deselect all the results.

Click the **CANCEL** button to end the plugin.

Click the **OK** button to run the **TRACE DIFFUSION** plugin with the selected results. The **TRACE DIFFUSION** plugin will not have the **INPUT** option as the results have already been selected. If multiple datasets are chosen the dataset name in the results table will be named using the first dataset plus the number of additional datasets, e.g. Dataset 1 + 6 others.

Note that the plugin supports the **IMAGEJ** recorder to allow running within an **IMAGEJ** macro.

## 11 Tools Plugins

The following plugins contain utilities for image manipulation.

The plugins are described in the following sections using the order presented on the **PLUGINS > GDSC SMLM > TOOLS** menu.

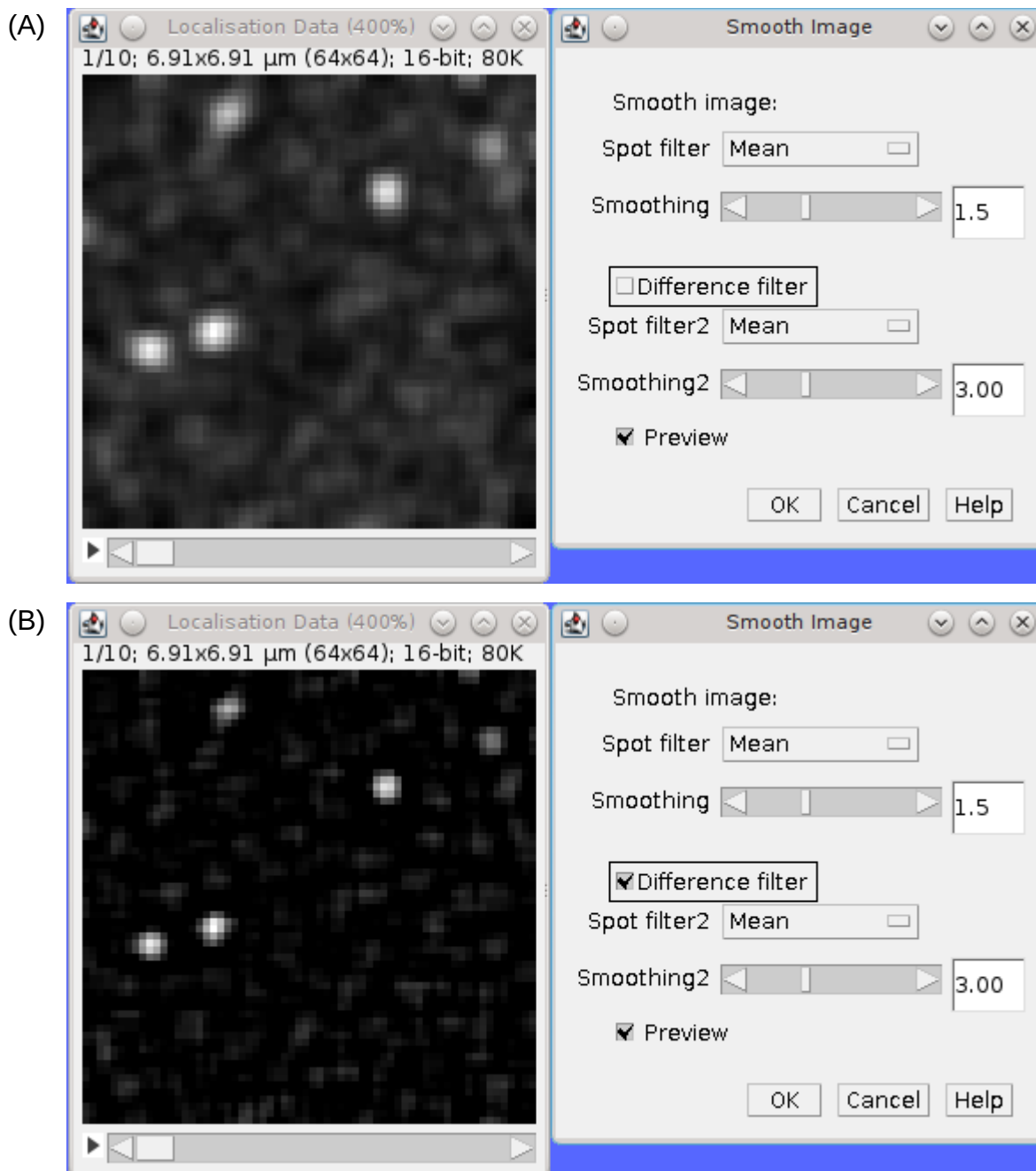
### 11.1 Smooth Image

Provides a filter plugin for smoothing an image (see Illustration 44).

The filter uses the same methods as the **P<sub>EA</sub>K F<sub>IT</sub>** plugin for identifying local maxima. However only a single or difference filter is available (no Jury filter) so that a single dialog can display all the options. A single filter applies a single smoothing operation to the image. A difference filter applies two smoothing operations and the second smoothed image is subtracted from the first. This produces a difference-of-smoothing image. As can be seen by comparing Illustration 47(A) and (B), the difference filter is beneficial when there is a variable background across the image. The filter reduces the contribution the background has to the brightness of the spots by performing local contrast enhancement.

The following algorithms are available:

| Algorithm     | Description   |
|---------------|---|
| MEAN          | Compute the mean in a square region. The region can be any size as the final edge pixels are given a weight using the region width  |
| BLOCK MEAN    | Compute the mean in a square region. The region is rounded to integer pixels  |
| CIRCULAR MEAN | Compute the mean in an approximate circular region. The circle is drawn using square pixels. To see the circle mask use <b>P<sub>RO</sub>CES</b> S > <b>F<sub>IL</sub>T</b> ERS > <b>S</b> HOW CIRCULAR MASKS...  |
| GAUSSIAN      | Perform Gaussian convolution. The convolution kernel standard deviation is set to the <b>S<sub>MOOTH</sub></b> parameter.<br><br>The total region width around each point that is used will be $2n+1$ with $n=\lceil 2.8\sigma \rceil$ where $\lceil x \rceil$ is the ceiling function. |
| MEDIAN        | Compute the median in a square region. The region is rounded to integer pixels  |



*Illustration 44: Smooth image dialogue with the preview applied to the current image. (A) Standard smoothing for a 16-bit image. (B) Difference-of-smoothing for a 32-bit image.*

Use the **P<sub>REVIEW</sub>** button to see the effect of smoothing. If you click **OK** the plugin will perform smoothing on the entire stack or optionally just the current frame.

### 11.1.1 Smoothing within the **P<sub>EAK</sub> F<sub>IT</sub>** plugin

Note that the **P<sub>EAK</sub> F<sub>IT</sub>** plugin calculates the smoothing window size using a factor of the PSF width. This can result in a non-integer value. Algorithms that require integer window sizes (e.g. Block mean, median) have the window size rounded down to the nearest integer to avoid over smoothing.



Most images analysed within the `PEAK FIT` plugin will use a filter size less than 4 due to the small size of the PSF from single molecule microscopy.

## 11.2 Binary Display

Switches an image to binary (white/black) to allow quick visualisation of localisations.

The SMLM plugins contain several methods for generating an image. Often images are created with a large difference in value for pixels that contain localisations. The difference can be so great that some localisations are not visible. The Binary Display plugin converts all non-zero pixels to the value 1. This allows the user to see any pixel that contains any level of localisation. This mode may be useful for drawing regions of interest (ROIs) around dense sections of localisations.

Note that the data for the image are directly updated. The data can be reset using the `RESET DISPLAY` plugin.

## 11.3 Reset Display

Resets a binary image generated by `BINARY DISPLAY` back to the standard display.

This will only work for stack images if the user remains at the same slice position. Moving to a new slice and back will delete the information used to reset the image.

## 11.4 Pixel Filter

Perform filtering to replace hot pixels from an image.

The `PIXEL FILTER` is a simple plugin that will replace pixels with the mean of the surrounding region if they are more than N standard deviations from the mean. The filter is designed to remove outlier (hot) pixels that are much brighter than their neighbour pixels. These pixels will be identified as candidate maxima by the `PEAK FIT` plugin although they are not suitable for Gaussian fitting.

The filter operates on the currently selected image. The preview option allows the results of the filter to be viewed before running the filter on the current frame or optionally the entire image stack.

The filter uses a cumulative sum and sum-of-squares lookup table to compute the region mean and standard deviation. This allows fast computation in constant time regardless of the size of the neighbourhood region.

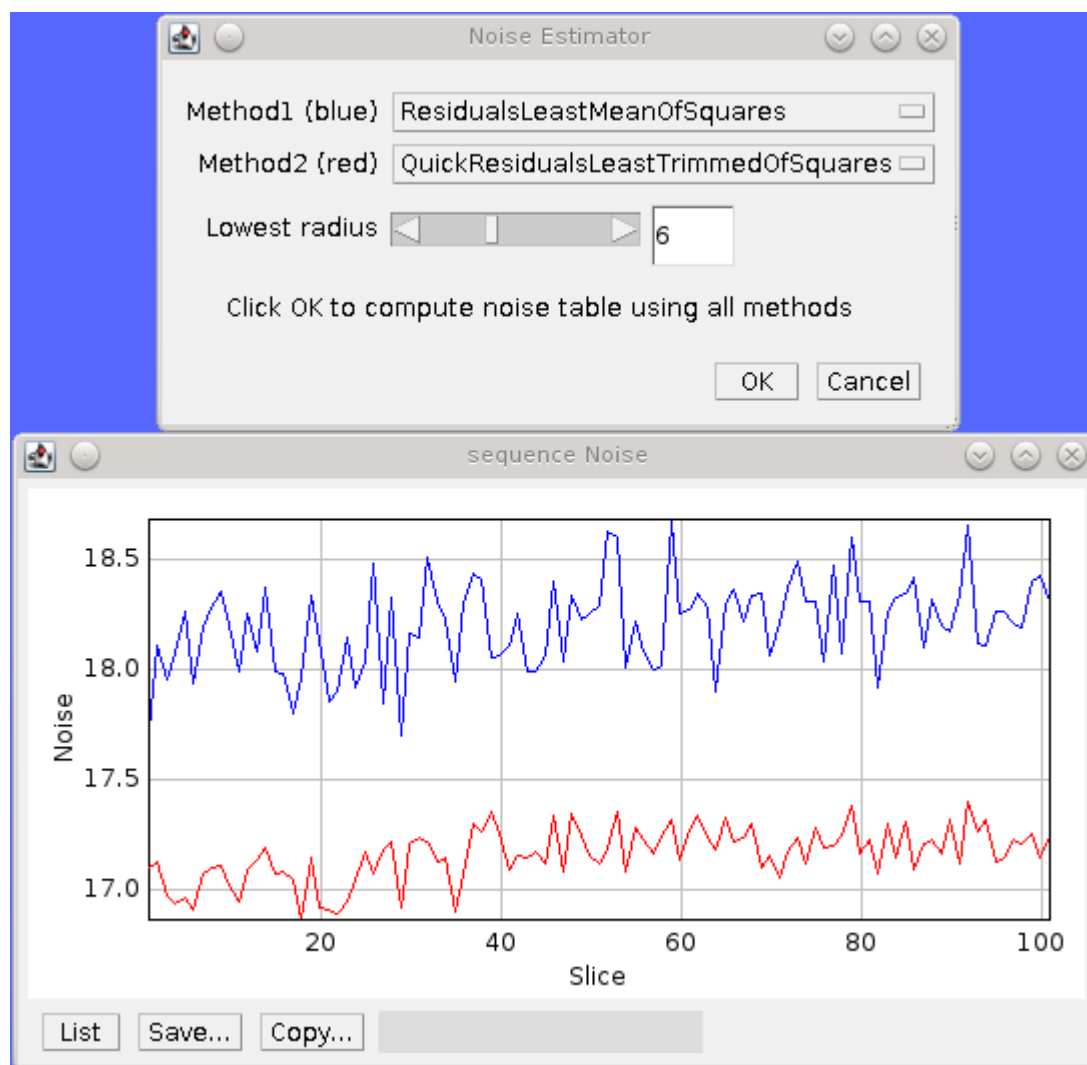
The following parameters can be set:

| Parameters | Description  |
|------------|--|
| RADIUS     | The radius of the square neighbourhood region  |
| ERROR      | The number of standard deviations above the mean that identifies a hot pixel                 |
| PREVIEW    | Preview the filter on the image. The number of pixels replaced will be shown in the dialogue |

## 11.5 Noise Estimator

Estimates noise in an image. This plugin can be used to compare noise estimation methods. Note that estimating the noise in an image is important when setting the signal-to-noise ratio (SNR) for use in filtering localisation fitting results.

When loaded the plugin provides a plot of the noise estimate of the next 100 frames of a stack (from the current frame) as shown in Illustration 47. Two noise estimation methods can be chosen (see table below). Changing an estimation method will dynamically update the noise plot.



*Illustration 45: Noise Estimator plugin showing the noise plot for the next 100 frames in the image. METHOD1 is shown in blue, METHOD2 is shown in red.*

If you click OK the plugin will compute all the estimation methods for the entire stack (or optionally just the current frame) and display the results in a table.

The following noise methods are available:

| Method     | Description                          |
|------------|--------------------------------------|
| ALL PIXELS | The standard deviation of the pixels |

|  |   |
|--|---|
|  |   |
| LOWEST PIXELS                                  | The standard deviation of a box region around the lowest intensity pixel in the image. The box region can be adjusted using the LOWEST RADIUS parameter. (This is the method used within QuickPALM [Henriques <i>et al</i> , 2010] and can produce inconsistent noise levels between frames due to the small sample size) |
| RESIDUALS LEAST MEDIAN OF SQUARES              | Calculate the median of the residuals. Then use this to estimate the standard deviation of the residuals  |
| RESIDUALS LEAST TRIMMED SQUARE ESTIMATOR       | Square the residuals. Sum the smallest half of the squared residuals. Then use this to estimate the standard deviation of the residuals. (This is insensitive to high intensity pixels)   |
| RESIDUALS LEAST MEAN SQUARE ESTIMATOR          | Calculate the standard deviation of the residuals   |
| QUICK RESIDUALS LEAST MEDIAN OF SQUARES        | As before but ignore pixels on the image boundary   |
| QUICK RESIDUALS LEAST TRIMMED SQUARE ESTIMATOR | As before but ignore pixels on the image boundary   |
| QUICK RESIDUALS LEAST MEAN SQUARE ESTIMATOR    | As before but ignore pixels on the image boundary   |

### 11.5.1 Image Residuals

The residuals of an image are calculated for each pixel using the total difference to the 4n connected pixels. These are normalised so that the sum of the residuals squared is the same as the sum of the image pixels squared. Comparing each pixel to its neighbours provides a robust method of estimating noise if the underlying signal is adequately sampled. Variations between neighbour pixels are expected to be small, consequently large variations indicate high noise.

All the image residuals methods are based on the “Least trimmed square” robust estimator described in:

P. Rousseeuw and A. Leroy, Robust Regression and Outlier Detection. New York: Wiley, 1987

### 11.5.2 Noise estimation within the Peak Fit plugin

The fitting code currently uses the QUICKRESIDUALSLEASTMEANOF SQUARES. This method is more stable than using the standard deviation of the image pixels since large variations around the high intensity localisations are smoothed by using the image residuals.

The noise estimation method can be changed in the PEAK FIT plugin by holding the SHIFT or ALT key down when running the plugin to see the extra options.

## 11.6 Median Filter

Compute the median of an image, on a per-pixel basis, using a rolling window at set intervals.

Super-resolution image data can contain a low amount of background which affects the performance of fitting routines if it is not constant, for example cell walls may be visible as a change in low level fluorescence over a distance of a few pixels. This uneven background will not be modelled by a fitting routine which assumes the background is constant. Any local gradients in the background can be eliminated by assuming that all real fluorescence over a short time frame will be much higher than the other values for the pixels in the same location. Using the median value for the pixel will approximate the background. This can be subtract from the image data prior to fitting so that only fluorescent bursts are left for fitting.

The **MEDIAN FILTER** plugin will compute the median for each pixel column through the image (i.e. all z positions of the pixel) using a rolling window. The median can be calculated at every pixel or at intervals. In the case of interval calculation then the intermediate points have linearly interpolated medians.

The median image either replaces the input image, or is subtracted from the input image to produce an image with only localisations. A bias offset is added to this image to allow noise to be modelled (i.e. values below zero).

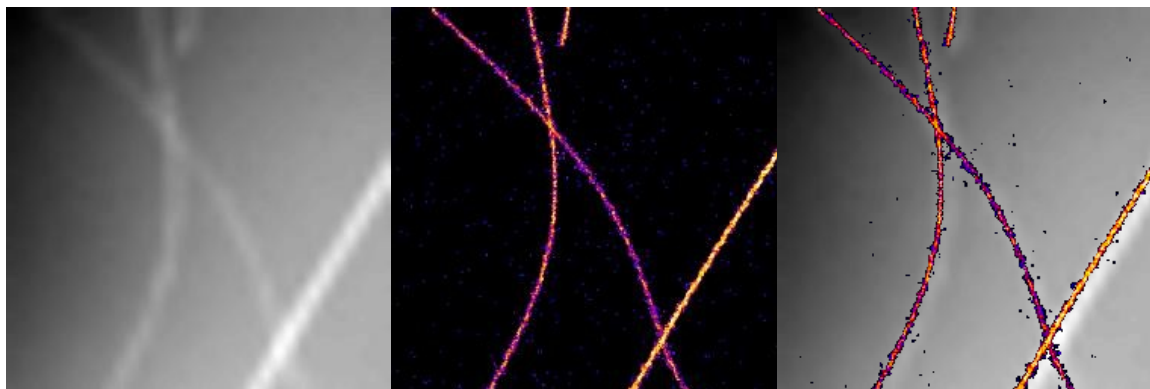
The following parameters can be specified:

| Parameters | Description  |
|------------|--|
| RADIUS     | The number of pixels to use for the median. The median is calculated using a window of $(2 \times \text{radius} + 1)$  |
| INTERVAL   | The interval between slices to calculate the median. An interval of 1 will produce a true rolling median. Larger intervals will require interpolation for some pixels  |
| BLOCK SIZE | The algorithm is multi-threaded and processes a block of pixels on each thread in turn. Specify the number of pixels to use in a block. Larger blocks will require more memory due to the algorithm implementation for calculating rolling medians.<br><br>The number of threads is set in <b>EDIT &gt; OPTIONS &gt; MEMORY &amp; THREADS...</b> |
| SUBTRACT   | Subtract the median image from the original image  |
| BIAS       | If subtracting the median, add a bias to the result image so that negative numbers can be modelled (i.e. when the original image data is lower than the median)  |

## 11.7 Overlay Image

Allow an image to be added as an overlay with a transparent background. Using a transparent background is not possible with the standard ImageJ 'IMAGE > OVERLAY > ADD IMAGE...' command.

For example the super-resolution image created from fitting localisations can be overlaid on the average z-projection of the original image to show where the localisations occur:



The OVERLAY IMAGE plugin must be run after selecting the image to overlay. The following parameters can be specified:

| Parameters             | Description   |
|------------------------|---|
| IMAGE TO ADD           | Select the image to use as the overlay.<br>The list only shows the images that are valid. Overlay images must be equal or smaller in width and height than the target image |
| X LOCATION             | The x location to insert the overlay (measured from the top-left corner)  |
| Y LOCATION             | The y location to insert the overlay (measured from the top-left corner)  |
| OPACITY                | The opacity of the overlay. 100% will totally obscure the underlying image  |
| TRANSPARENT BACKGROUND | Select this to use a transparent background for any pixels with a value of zero. This allows the underlying image to be seen even when the opacity is set to 100%           |
| REPLACE OVERLAY        | Select this to replace the current overlay. Uncheck this to add to the current overlay (i.e. combine overlays)  |

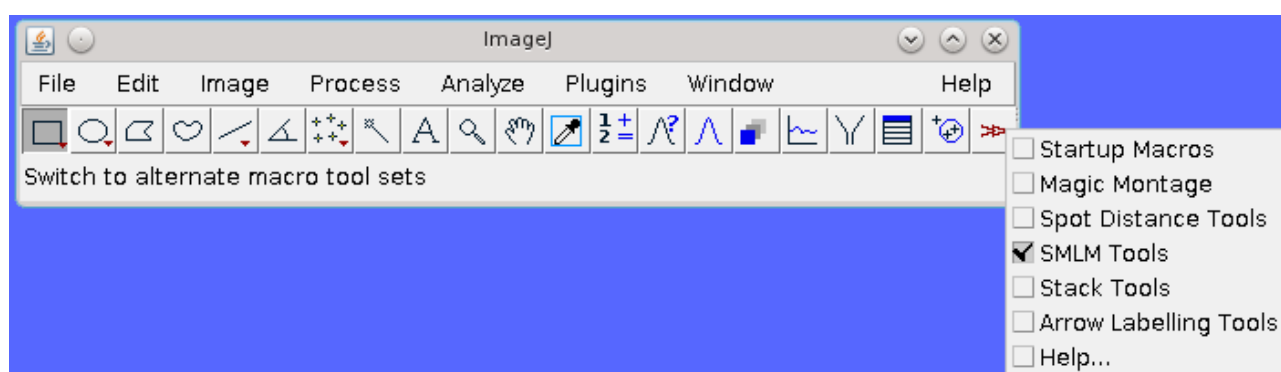
## 12 Toolset Plugins

The following plugins add functionality to ImageJ to allow the SMLM plugins to be run using the ImageJ toolbar or a dedicated window panel of buttons.

The plugins are described in the following sections using the order presented on the **PLUGINS > GDSC SMLM > TOOLSET** menu.

### 12.1 Install SMLM Toolset

You can install a toolset for the SMLM plugins using the 'GDSC SMLM > INSTALL SMLM TOOLSET' plugin. This will create a text file in the ImageJ/macros/toolsets directory. The toolset can be activated using the 'More Tools' menu button (the last button on the ImageJ toolbar) as shown in Illustration 46.



*Illustration 46: Activate the SMLM toolset using the ImageJ 'More Tools' menu.*

The toolset defines 8 buttons for the ImageJ menu bar that allow various plugins to be run. When the mouse is over the button the name of the plugin will be shown in the ImageJ window. Click the button to run the plugin.

Note: The latest version of the 'SMLM Tools.txt' file is packaged within the SMLM Jar file. This can be manually extracted using an archiving utility. The **INSTALL SMLM TOOLSET** plugin simply extracts this file and writes it to the ImageJ macros/toolsets directory if the user has the correct access permissions.

### 12.2 Show SMLM Tools

Displays a window with buttons to run all the SMLM plugins.

The window is constructed using a configuration file. The plugins listed in the configuration file are used to build a window with buttons to allow them to run. The file is searched for in the following locations:

- [ImageJ Path]/plugins/smlm.config
- [SMLM Jar File]/gdsc/smlm/plugins.config

Noted that the plugins.config file within the jar file is used by ImageJ to build the entries for the plugins menu. To customise the ImageJ menus the plugins.config file can be extracted from the Jar file, modified and replaced. This will also change the SMLM Tools window unless a separate smlm.config file is placed in the ImageJ plugins directory.

## GDSC SMLM ImageJ Plugins

The plugins.config file has entries consisting of 3 fields separated by commas. Lines starting with # are ignored:

e.g.

```
#ImageJ menu, "Plugin name", full.java.plugin.ClassName("arguments")
```

```
Plugins>GDSC SMLM, "Peak Fit", gdsc.smlm.ij.plugins.PeakFit
```

```
Plugins>GDSC SMLM, "Peak Fit", gdsc.smlm.ij.plugins.PeakFit("spot")
```

```
Plugins>GDSC SMLM, "-"
```

This example shows that the SPOT FINDER plugin is running the PEAK FIT plugin with an argument indicating that it should only find the spot candidates.

The SMLM TOOLS window is a series of buttons arranged in columns. Each button on the window has the name of the plugin. The buttons are ordered using the order of the configuration file. Any separator entries in the file using the plugin name "-" will result in padding being applied between buttons (a separator row). Separator rows are also added for empty lines. 2 or more successive separator entries will result in a new column. Illustration 47 shows an example of the SMLM TOOLS window.

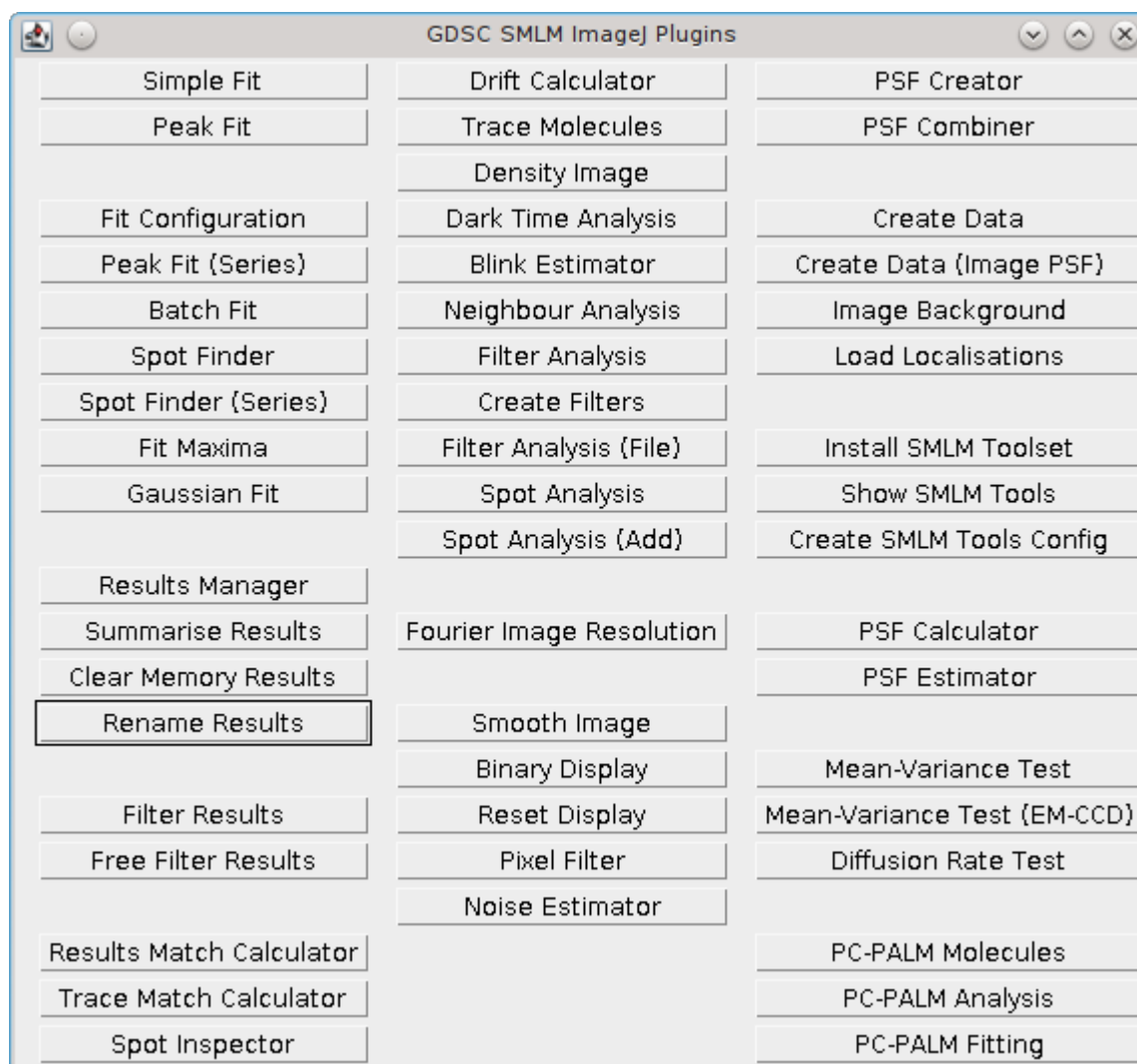


Illustration 47: SMLM Tools window

### **12.3 Create SMLM Tools Config**

Create a configuration file allowing the SMLM TOOLS window to be customised.

This plugin extracts the plugins.config file from the SMLM jar file and writes it to [ImageJ Path]/plugins/smlm.config. Optionally the file can be edited before installation to customise the plugins that appear on the SMLM TOOLS window (see 12.2: Show SMLM Tools).

If the file cannot be written the plugin will report an error. If the file already exists the user will be informed that they will overwrite the file and the option to remove the existing file is also provided. If removed, the SMLM TOOLS window will revert to showing the default plugins.

If the plugin is run and the SMLM TOOLS window is currently open then it will be closed and re-opened to update to the new configuration.



## Appendix A Localisation Precision

The theoretical limit (precision) for fitting the signal (number of photons) and the XY coordinates (localisation) can be computed using the formulas of Thompson *et al* (2002) for the signal and Mortensen *et al* (2010) for the localisation.

Note that these formulas are derived from modelling the point spread function (PSF) as a 2D Gaussian for both the simulation and the fitting. Given that the true data will have a PSF defined by the microscope parameters these formulas only approximate the precision that can be obtained on image data.

The photon count  $N$  is computed using the volume of the fitted 2D Gaussian function (the signal) divided by the camera gain:

$$N = \frac{\text{Signal}}{\text{Gain}}$$

The background noise  $b^2$  is estimated by the P<sub>PEAK FIT</sub> plugin during the fitting process either using a global noise estimate per frame or the local background level. This is also adjusted by the camera gain to provide the noise in photons.

Note that the precision of the localisation is the square root of the variance.

**Warning: Without a correctly calibrated camera gain and bias the precision estimates in the SMLM code will be inaccurate.**

### A.1 Signal Precision

From Thompson *et al* (2002).

$$\text{Var}_N = F \times \left( N + \frac{4\pi s^2 b^2}{a^2} \right)$$

where

|                |   |
|----------------|---|
| $\text{Var}_N$ | The variance of the signal a Gaussian 2D function to a Gaussian 2D PSF  |
| $F$            | The noise scaling factor. 2 for an EM-CCD camera; 1 otherwise   |
| $s$            | The standard deviation of the Gaussian function   |
| $N$            | The number of photons in the localisation   |
| $b^2$          | The expected number of photons per pixel from a background with spatially constant expectation value across the image |
| $a$            | The pixel size in nm  |

From Thompson *et al* (2002).

### A.2 Localisation Precision for Least Squares Fitting

From Mortensen *et al* (2010).

$$Var_x = F \times \frac{s_a^2}{N} \times \left( \frac{16}{9} + \frac{8\pi s_a^2 b^2}{Na^2} \right)$$

where

|         |   |
|---------|---|
| $Var_x$ | The variance of the localisation position in the X dimension when fitting a Gaussian 2D function to a Gaussian 2D PSF                         |
| $F$     | The noise scaling factor. 2 for an EM-CCD camera; 1 otherwise   |
| $s_a$   | The standard deviation of the Gaussian function (s) adjusted for square pixels. The adjustment is computed as:<br>$S_a = \sqrt{s^2 + a^2/12}$ |
| $N$     | The number of photons in the localisation   |
| $b^2$   | The expected number of photons per pixel from a background with spatially constant expectation value across the image                         |
| $a$     | The pixel size in nm  |

### A.3 Localisation Precision for Maximum Likelihood Fitting

From Mortensen *et al* (2010).

$$Var_x = F \times \frac{s_a^2}{N} \times \frac{1}{-\int_0^1 \frac{t \ln(t)}{t+\rho} dt}$$

where

|         |   |
|---------|---|
| $Var_x$ | The variance of the localisation position in the X dimension when fitting a Gaussian 2D function to a Gaussian 2D PSF                         |
| $F$     | The noise scaling factor. 2 for an EM-CCD camera; 1 otherwise   |
| $s_a$   | The standard deviation of the Gaussian function (s) adjusted for square pixels. The adjustment is computed as:<br>$S_a = \sqrt{s^2 + a^2/12}$ |

where  $a$  is the pixel size in nm

|        |   |
|--------|---|
| $N$    | The number of photons in the localisation   |
| $\rho$ | The integration factor: $\frac{2\pi s_a^2 b^2}{Na^2}$   |
| $b^2$  | The expected number of photons per pixel from a background with spatially constant expectation value across the image |

Note that since the formula for Maximum Likelihood fitting involves an integral with no analytic solution the formula is evaluated using numerical integration. This is slow to compute.

The formula can be used to demonstrate that for any given set of parameters the precision

of Maximum Likelihood fitting is lower (i.e. better) than Least Squares fitting.

## Appendix B Comparison Metrics

Several plugins within the SMLM package compute matches between points. One set of points can be labelled as the actual result, the second can be labelled as the predicted results. When comparing actual and predicted points the following combinations are possible:

| Actual Point        | Predicted Point     | Classification | Label     |
|---------------------|---------------------|----------------|-----------|
| P <sub>RESENT</sub> | P <sub>RESENT</sub> | True Positive  | <i>tp</i> |
| P <sub>RESENT</sub> | A <sub>BSENT</sub>  | False Negative | <i>fn</i> |
| A <sub>BSENT</sub>  | P <sub>RESENT</sub> | False Positive | <i>fp</i> |
| A <sub>BSENT</sub>  | A <sub>BSENT</sub>  | True Negative  | <i>tn</i> |

The classification counts can be used to compute binary scoring statistics as described below.

### B.1 Recall

Recall measures the number of actual points that are correctly predicted. It is also known as the True Positive Rate (TPR) or sensitivity.

$$recall = \frac{tp}{tp + fn}$$

A score of 1 indicates that all the points were predicted, lower scores indicate that some points were missed.

Recall can be interpreted probabilistically as the chance that a randomly selected actual point will be predicted.

### B.2 Precision

Precision measures the confidence of the predicted points. It is also known as the Positive Predicted Value (PPV).

$$precision = \frac{tp}{tp + fp}$$

A score of 1 indicates that all the predicted points were correct, lower scores indicate that some points are not correct.

Precision can be interpreted probabilistically as the chance that a randomly selected prediction is correct.

### B.3 Jaccard

The Jaccard measures the similarity between two sets and is defined as the size of the intersection divided by the size of the union:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{tp}{tp + fp + fn}$$

A score of 1 indicates that the overlap is perfect. Zero indicates no overlap.

## **B.4 F-score**

The precision and recall can be combined in a weighted score, the  $F_\beta$ -measure or F-score.

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

A weight of 1 produces the balanced F-score where precision and recall are weighted equally.  $F_{0.5}$  puts more emphasis on precision and  $F_2$  puts more emphasis on recall. However the weight  $\beta$  can be any non-negative real value. The score was derived so that it measures the effectiveness of retrieval with respect to a user who associates  $\beta$  times as much importance to recall as precision.

## **B.5 FNR**

False-negative rate:

$$FNR = \frac{fn}{fn + tp}$$

## **B.6 FDR**

False discovery rate:

$$FDR = 1 - \text{Precision} = \frac{fp}{tp + fp}$$

## **B.7 Defining the Actual or Predicted Points**

Note that the categorisation of the actual and predicted points can be arbitrary. If the categorisation is reversed then the precision and recall scores will be reversed. However the  $F_1$ -score will remain identical because it is the harmonic mean of the two scores. The  $F_1$ -score is consequently a good measure of the similarity of two sets of points that have been aligned for matches.

## **B.8 Metrics requiring the True Negative count**

Note that when comparing point coordinates that can only ever exist the definition of the true negative is invalid. However there are cases where a set of results have two classifications (either absent or present), and a set of predictions aim to predict those values. In this case the TN count can be obtained and used to compute scores:

## **TNR**

True-negative rate:

$$TNR = \frac{tn}{fp + tn}$$

## NPV

Negative predictive value:

$$NPV = \frac{tn}{tn + fn}$$

## FPR

False-positive rate:

$$FPR = \frac{fp}{fp + tn}$$

## ACC

Accuracy:

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn}$$

## MCC

The Matthews Correlation Coefficient is used in machine learning as a measure of the quality of binary (two-class) classifications, introduced by biochemist Brian W. Matthews in 1975. It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC is in essence a correlation coefficient between the observed and predicted binary classifications; it returns a value between  $-1$  and  $+1$ . A coefficient of  $+1$  represents a perfect prediction,  $0$  no better than random prediction and  $-1$  indicates total disagreement between prediction and observation. The statistic is also known as the phi coefficient.

$$MCC = \frac{tp * tn - fp * fn}{\sqrt{tp + fp * tp + fn * tn + fp * tn * fn}}$$

## Informedness

$$Informedness = TPR + TNR - 1$$

## Markedness

$$Markedness = PPV + NPV - 1$$

## Appendix C MSD Correction

The mean squared distance (MSD) is classically computed using the distance between a diffusing molecule after time  $t$  and the origin. However the situation is different for localisations that have been determined using imaging of moving fluorescent objects. The position is not the exact position of the object but an average over the exposure time used to acquire the image frame. The distance between two frames thus represents the distance between an average position in the start and end frames, optionally with extra frames inserted in-between. The extra frames only effect the time between the start and end frame.

The effect of averaging the position in the start and end frames can be examined using a simulation. Diffusion can be simulated by sampling from a Gaussian distribution to generate random movements in a single dimension. The width of the Gaussian ( $s$ ) is related to the diffusion coefficient ( $D$ ) and time of diffusion ( $t$ ):

$$s = \sqrt{2Dt}$$

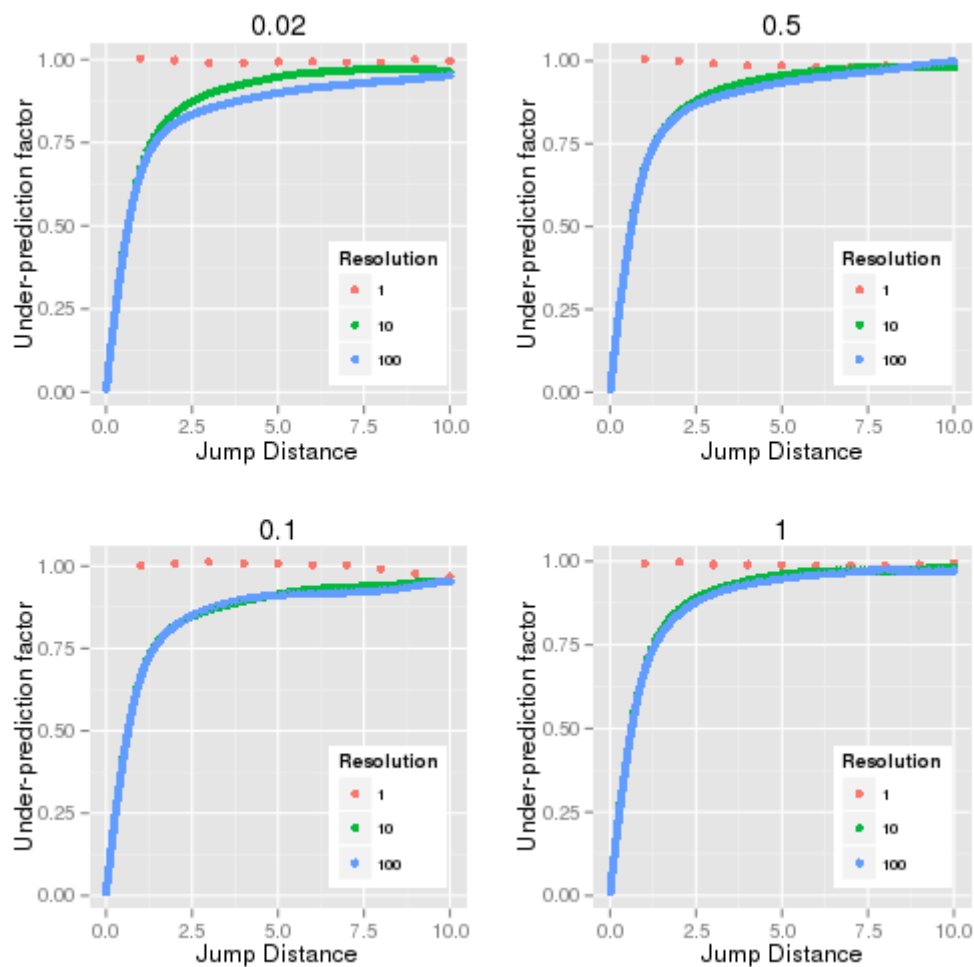
Samples can be taken for X and Y dimensions independently to simulate a diffusion track for a molecule. The location of a fluorophore within a frame can be determined using the average of a number of  $N$  consecutive positions from the simulated diffusion track. The location of the fluorophore in the next frame is the average of the next  $N$  consecutive positions. The distance between these two points can then be calculated.

Since this is a simulation is possible to calculate distances between overlapping frames of  $N$  simulation steps that are offset by  $j$  simulation steps. This is performed by taking the distance:

$$d = \sum_i^{i+N} \frac{x_n}{N} - \sum_{i+j}^{i+j+N} \frac{x_n}{N}$$

for each dimension and computing the mean squared distance in two dimensions.

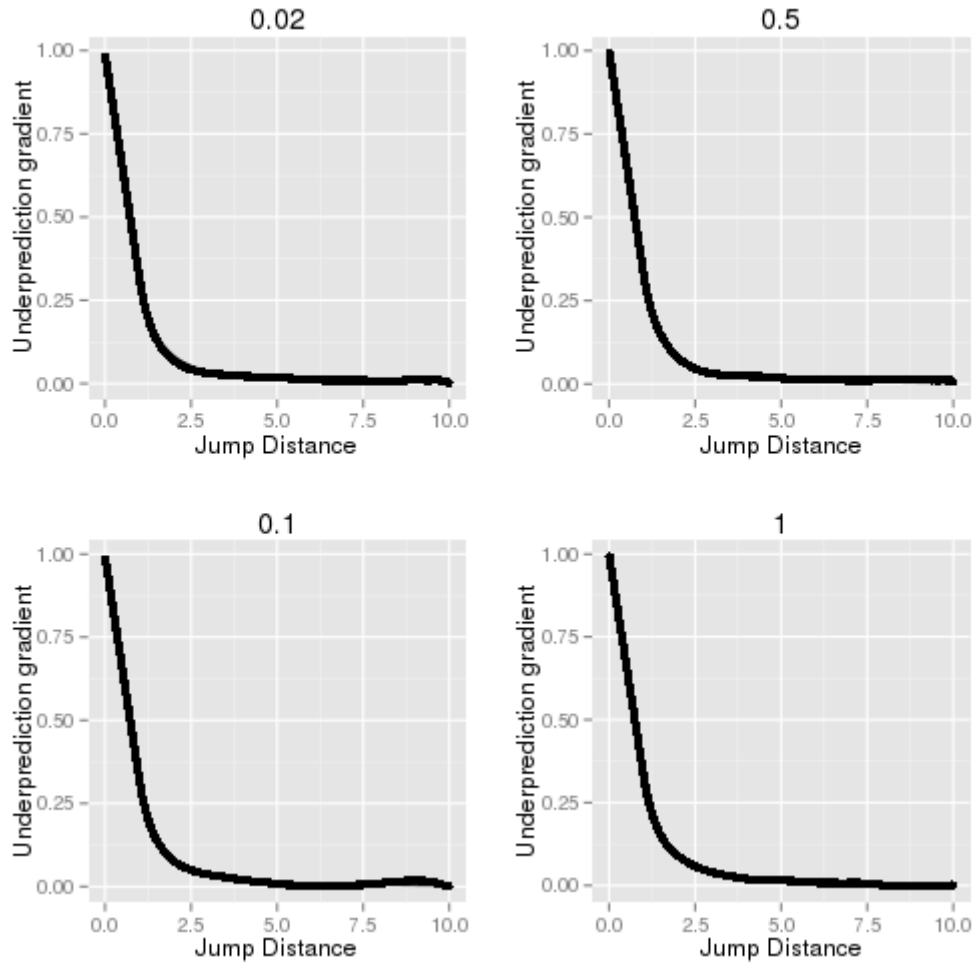
The following graph shows the ratio (under-prediction factor) between the observed MSD and the expected MSD against the frame overlap (Jump distance):



The simulation was performed using different number of steps per frame (resolution) and different diffusion coefficients. There is no under-prediction when the resolution is 1. At higher resolutions the under-prediction converges to a curve that is constant irrespective of the diffusion coefficient.



The curve gradient can be computed by taking the difference of adjacent values in Y divided by the difference in X. For the high resolution curve this results in:



### C.1 $n < 1$

The curve is linear for jump distances under 1 frame. For  $n < 1$  the gradient of the gradient is  $-2/3$  and the curve is described by:

$$\frac{du}{dn} = 1 - \frac{2n}{3}$$

$$\int 1 - \frac{2n}{3} dn = n - \frac{n^2}{3} + C$$

C equates to zero to fit the under-prediction line.

### C.2 $n > 1$

The gradient follows a power law for  $n > 1$ .

$$\frac{du}{dn} = \frac{1}{3n^2}$$

$$\int \frac{1}{3n^2} dn = C - \frac{1}{3n}$$

C equates to 1 to fit the under-prediction line.

Given that the under-prediction is only relevant to experimental data when the number of frames is 1 or more the observed MSD can be converted to the true MSD by dividing by a correction factor ( $F$ ):

$$F = 1 - \frac{1}{3n} = \frac{3n}{3n} - \frac{1}{3n} = \frac{3n-1}{3n} = \frac{n-1/3}{n}$$

This correction factor effectively states that the diffusion of the objects within the start and end frames has reduced the time interval by 1/3. This is logical since any more frames added in between the start and end frames count entirely. Only the start and end frames are composed of an average location of a diffusing particle.

When performing jump distance analysis it is not necessary to correct each observed squared distance before fitting. Since the correction is a single scaling factor instead the computed diffusion coefficient can be adjusted by applying the correction factor.

### C.3 Fitting the plot of MSD verses $N$ frames

When fitting the linear plot of MSD verses the number of frames we can account for the correction factor. The observed MSD is composed of the actual MSD multiplied by the correction factor before being adjusted for the precision error:

$$oMSD(n\Delta t) = MSD(n\Delta t) \times \left(\frac{n-1/3}{n}\right) + 4\sigma^2$$

$$oMSD(n\Delta t) = 4D(n\Delta t) \times \left(\frac{n-1/3}{n}\right) + 4\sigma^2$$

$$oMSD(n\Delta t) = 4D(\Delta t) \times n \left(\frac{n-1/3}{n}\right) + 4\sigma^2$$

$$oMSD(n\Delta t) = 4D(\Delta t) \times (n-1/3) + 4\sigma^2$$

$$oMSD(n\Delta t) = 4D(n\Delta t) - \frac{4D(\Delta t)}{3} + 4\sigma^2$$

This is still a linear fit. The intercept has a new representation and allows the intercept to be negative. To ensure the intercept is correctly bounded it should be represented using the fit parameters and not just fit using a single constant  $C$ .

Note: The final fitted equation is the 2-dimension equivalent of the equation provided in Backlund, *et al* (2015), equation (1):

The effect of static and dynamic errors on the expected value of the estimated 1D MSD for the special case of pure Brownian motion is to produce a constant offset of the linear dependence according to:

$$E[\hat{M}(n)] = 2Dnt_E + 2\sigma^2 - \frac{2}{3}Dt_E$$

where  $t_E$  is the exposure time of the acquisition,  $n$  is the number of frames spanning the lag,  $D$  is the diffusion coefficient,  $\sigma$  is the localisation error including an additional correction due to the spreading of photons over a greater area for a moving particle,  $E$  is the expectation value and the hat denotes the estimated quantity.

## 13 References

- Abraham, A. V., Ram, S., Chao, J. Ward, E.S., Ober, R. J. (2009) Quantitative study of single molecule location techniques. *Optical Express* 17, 23352.
- Annibale, P., Vanni, S., Scarselli, M., Rothlisberger, U., Radenovic, A. (2011). Quantitative Photo Activated Localization Microscopy: Unravelling the Effect of Photoblinking. *PLoS ONE* 6(7), e22678.
- Backlund, M.P., Joyner, R., Moerner, W.E. (2015) Chromosomal locus tracking with proper accounting of static and dynamic errors. *Physical Review E* 91, 062716
- Coltharp, C, Kessler, RP, Xiao, J (2012), Accurate Construction of Photoactivated Localization Microscopy (PALM) Images for Quantitative Measurements. *PLoS One*. 7(12).
- Edelstein, A., Amodaj, N., Hoover, K., Vale, R., and Stuurman, N. (2010), Computer Control of Microscopes Using  $\mu$ Manager. *Current Protocols in Molecular Biology* 14.20.1-14.20.17.
- Heel, M and Schatz, M (2005), Fourier shell correlation threshold criteria. *Journal of Structural Biology*, Volume 151, Issue 3, Pages 250–262.
- Henriques, R., Lelek, M., Fornasiero, E.F., Valtorta, F., Zimmer, C. & Mhlanga, M.M. (2010), QuickPALM: 3D real-time photoactivation nanoscopy image processing in ImageJ. *Nature Methods* 7:339–340.
- Hurvich, C. M.; Tsai, C.-L. (1989), Regression and time series model selection in small samples. *Biometrika* 76: 297–307.
- Mortensen KI, Churchman LS, Spudich JA, Flyvbjerg H (2010) Optimized localization analysis for single-molecule tracking and super-resolution microscopy. *Nat Methods* 7(5):377-81.
- Neubeck, A.; Van Gool, L. (2006), Efficient Non-Maximum Suppression. *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol.3, pp.850,855
- Nieuwenhuizen, R.P.J., Lidke, K.A., Bates, M., Puig, D.L., Grunwald, D., Stallinga, S., Rieger, B. (2013). Measuring image resolution in optical nanoscopy. *Nature Methods*, 10, 557.
- Puchnar, E.M., Walter, J.M., Kasper, R., Huang, B., Lim, W.A. (2013). Counting molecules in single organelles with superresolution microscopy allows tracking of the endosome maturation trajectory. *PNAS*, 110: 16015-16020.
- Schneider, C.A., Rasband, W.S., Eliceiri, K.W. (2012), NIH Image to ImageJ: 25 years of image analysis. *Nature Methods* 9, 671-675.
- Sengupta, P., Jovanovic-Talisman, T., Lippincott-Schwartz, J. (2013). Quantifying spatial resolution in point-localisation super-resolution images using pair correlation analysis. *Nature Protocols* 8: 345-354.
- Sengupta, P., Jovanovic-Talisman, T., Skoko, D., Renz, M., Veatch, S.L., Lippincott-Schwartz, J. (2011) Probing protein heterogeneity in the plasma membrane using PALM and pair correlation analysis. *Nature Methods* 8, 969-975.
- Snyder, D. L., Helstrom, C. W., Lanterman, A.D., Faisal, M., White, R.L. (1995)

Compensation for readout noise in CCD images. *Journal of the Optical Society of America* 12: 272-283.

Tubbs RN (2004) Lucky exposures: Diffraction limited astronomical imaging through the atmosphere. *Observatory* 124.

Ulbrich & Isacoff (2007) Subunit counting in membrane-bound proteins. *Nature Methods* 4, 319-321.

Uphoff, S., Reyes-Lamothe, R., Garza de Leon, F., Sherratt, D.J., Kapanidis, A.N. (2013), Single-molecule DNA repair in live bacteria. *PNAS*, 110: 8063-8068.

Veatch, S.L., Machta, B.B., Shelby, S.A., Chiang, E.N., Holowka, D.A., Baird, B.A. (2012). Correlation Functions Quantify Super-Resolution Images and Estimate Apparent Clustering Due to Over-Counting. *PLoS One* 7, Issue 2, e31457.

Weimann, L., Ganzinger, K.A., McColl, J., Irvine, K.L., Davis, S.J., Gay, N.J., Bryant, C.E., Klenerman, D. (2013) A Quantitative Comparison of Single-Dye Tracking Analysis Tools Using Monte Carlo Simulations. *PLoS One* 8, Issue 5, e64287.

Wolter, S., Schüttelpelz, M., Tscherepanow, M., Van De Linde, S., Heilemann, M. And Sauer, M. (2010), Real-time computation of subdiffraction-resolution fluorescence images. *Journal of Microscopy*, 237: 12–22.