ImageJ FindFoci 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

Table of Contents

Introduction

The FindFoci ImageJ plugins allow the identification of peak intensity regions within 2D and 3D images. The algorithm is highly configurable and parameters can be optimised using reference images and then applied to multiple images using the batch mode.

There are six different plugins that can be run from the ImageJ plugins menu:

1. FindFoci
2. FindFoci GUI
3. FindFoci Batch
4. FindFoci Optimiser
5. FindFoci Optimiser GUI
6. FindFoci Optimiser Multi-Image

The main algorithm is contained within the FindFoci plugin. This is a standard ImageJ plugin filter that operates on an opened image. It shows a dialogue box to request parameters and then performs the image processing. Results are presented as a mask image, shown in tabular format and/or saved to file.

The FindFoci GUI opens a permanent window frame within ImageJ. It has the same parameters as the FindFoci plugin. It runs the FindFoci plugin to perform the processing work. A preview option allows the parameters to be interactively updated.

The FindFoci Batch plugin allows a set of parameters for the FindFoci algorithm to be applied to all the images in a directory. The algorithm is run on each image in turn and the results are saved to an output directory.

The FindFoci Optimiser finds the best parameters for the FindFoci algorithm using the maxima marked on a representative image. It iterates through thousands of combinations and identifies the best parameters to find the desired peak regions. These parameters can then be easily applied to other images of the same type using the ImageJ scripting tools or the FindFoci Batch plugin.

The FindFoci Optimiser GUI opens a permanent window within ImageJ. It allows the user to select an image and run the FindFoci Optimiser. This is a convenience plugin to allow the optimiser to be run repeatedly on one or many images.

The FindFoci Optimiser Multi-Image plugin allows the optimiser to be run on a directory of reference images to find the best parameters across the dataset. The results for each image are saved to an output directory and can be reloaded allowing the parameters to be ranked using different scoring methods.

In addition to the plugins that run the FindFoci algorithms there are several plugins that provide useful functionality for peak analysis:

1. FindFoci Helper
2. FindFoci Settings
3. Match Calculator
4. Point Aligner
5. Assign Foci to Objects
6. Assign Foci to Clusters

The FindFoci Helper provides semi-automated assistance to aid identification of peaks in

an image. The plugin uses the FindFoci algorithm to identify all possible peaks in an image. Then when a user clicks on the image to label a peak using the ImageJ ROI tools the plugin updates the point, moving it to the nearest peak position if one is available. The plugin selects peaks within a search radius using either the nearest-neighbour or the highest peak. The plugin supports dragging points and removing false assignments. Results can be saved to a table.

The FindFoci Settings plugin sets options for the Find Foci algorithm, e.g. the limit on the number of potential maxima for the FindFoci algorithm. This prevents the algorithm running with too many maxima and prompts the user to adjust the FindFoci parameters to be more suitable.

The Match Calculator compares the ROI points marked on two images and calculates the number of points that match. Points are aligned using a nearest-neighbour ranking within a specified distance threshold and the results are displayed as a statistics table. Using the image intensity for each point allows further analysis of the data including a scatter plot of intensity for matched points and analysis of matches within each quartile of the data. The matched and unmatched points can be shown as a colour-coded overlay on the input image.

The Point Aligner can move the current ROI points to align with the true image peaks found using the FindFoci algorithm. Points are aligned using a nearest-neighbour ranking. Alignment results are output to a result table and can be used to update the ROI positions or shown as a colour-coded overlay on the image.

The Assign Foci to Objects plugin can align the most recent foci identified by the algorithm to objects on a mask image. Summary statistics are provided of the number of foci per object.

The Assign Foci to Clusters plugin can perform 2D clustering using the most recent foci identified by the algorithm. Summary statistics are provided of the number of foci per cluster and the clusters can be displayed on the FindFoci output mask image.

Further information

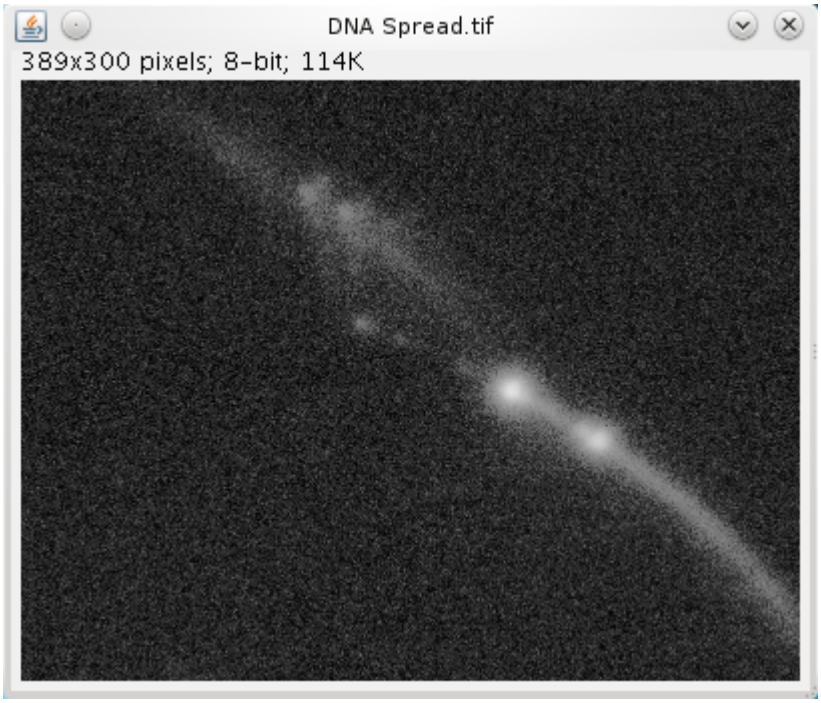
The FindFoci algorithm and the advantages of training software on multiple reference images for focus detection is described in the paper:

Herbert AD, Carr AM, Hoffmann E (2014) FindFoci: A Focus Detection Algorithm with Automated Parameter Training That Closely Matches Human Assignments, Reduces Human Inconsistencies and Increases Speed of Analysis. PLoS ONE 9(12): e114749. doi: 10.1371/journal.pone.0114749.

FindFoci

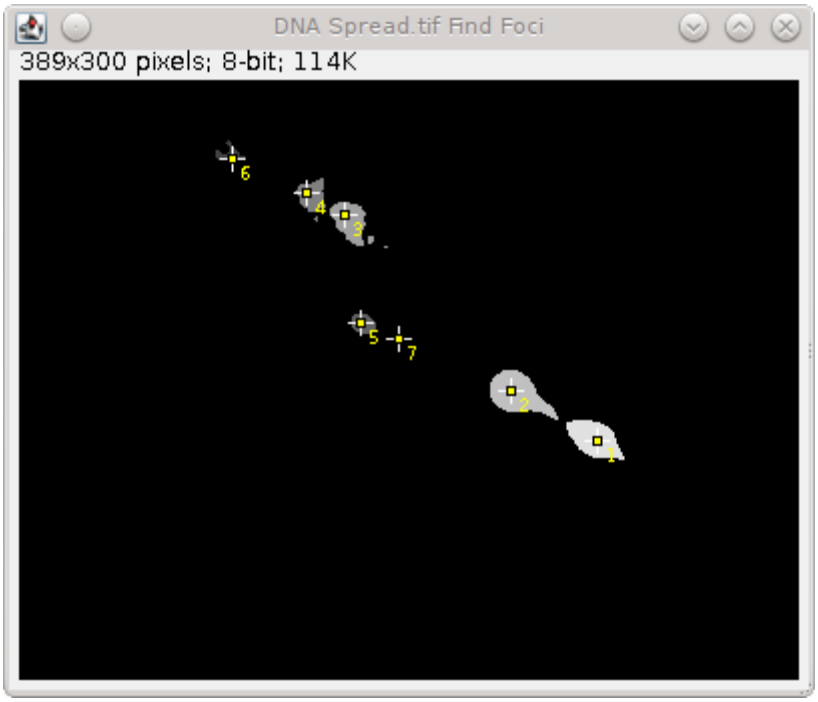
ImageJ plugin that finds areas of maximum intensity in 2D and 3D images.

Example Input



Input image with regions of high intensity for peak analysis.

Example Output



Output image showing the identified peak regions. Each peak centre is labelled with a point ROI marker.

Features

- Processes 8-bit, 16-bit and 32-bit greyscale images
- Processes 2D and 3D images (i.e. image stacks)
- Allows hyperstacks but only processes the current channel and time frame
- Finds peak regions above a configurable background
- Searches within the region of interest (ROI) if available
- Merges sub-peaks into their parent using merging criteria
- Ranks peaks using configurable criteria and outputs the top N
- Outputs the peaks regions as a new image
- Generates a results table
- Marks peak centres using ImageJ point ROIs
- Overlay the peak regions on the original image in 2D/3D
- Allows processing within a mask, labelling maxima within each identified mask object
- Saves results text table and images to a directory
- All functionality is scriptable
- Efficient two-stage algorithm is fast

Overview

The FindFoci plugin finds all the points of maximum intensity and expands them into regions. The initial regions are processed using merging criteria to produce a cleaner final result.

The following diagram shows an intensity profile to which the peak finding algorithm will be applied.

Stage 1: Assign Background

Assign the background level (all pixels below this level are ignored).

Stage 2: Assign Maxima

Find the points with an intensity higher than all the surrounding points. These local maxima are potential peaks.

Stage 3: Expand Peaks

Expansion is then performed by progressively assigning points of lower intensity to the peak area above it. This allows all points to be processed in a single pass from the highest to the lowest. The result is an initial assignment of all the points above the background to a peak.

Stage 4: Merge Peaks

The final stage is to merge insignificant peaks into their parent peak. This can be done if the sub-peak is not high enough or covers insufficient area. This stage is optional but can improve results on noisy images.

Image Input

When the FindFoci plugin is run the target image is the currently active image window within ImageJ. The image can be an 8-, 16- or 32-bit greyscale image. If the image has z-slices then the image is processed in 3D. If the image is a hyperstack the current time frame and channel are processed.

32-bit Support

FindFoci supports integer (8- or 16-bit) images and true floating point 32-bit images. 32-bit processing was added in August 2016, 5 years after the original FindFoci plugin was created. Testing ensured that the same functionality and speed when using integer images was preserved. Float processing can handle negative values and floating-point images that are output from numerical analysis methods, e.g. deconvolution of microscope images.

Integer images are processed using integer arithmetic. Float images are processed using floating point arithmetic. A key part of the algorithm is the processing of image pixels in descending height order using the image histogram values. The process of creating and working with a histogram (for example in automated thresholding methods) and during pixel value comparisons is faster for integer images. So when an integer input image is converted to a float image before processing the speed will be slower. However the results will be the same under many conditions. The system recognises the 32-bit image is an integer image and uses an integer histogram. True 32-bit images use an interpolated histogram for auto-thresholding methods and a full sorted histogram for height processing. Note that the threshold heights are computed in floating-point for 32-bit images (not rounded to the nearest integer) so results may differ when using a search method or peak height method that are fractions of the maximum value. If you use absolute values for peak heights the results will be identical.

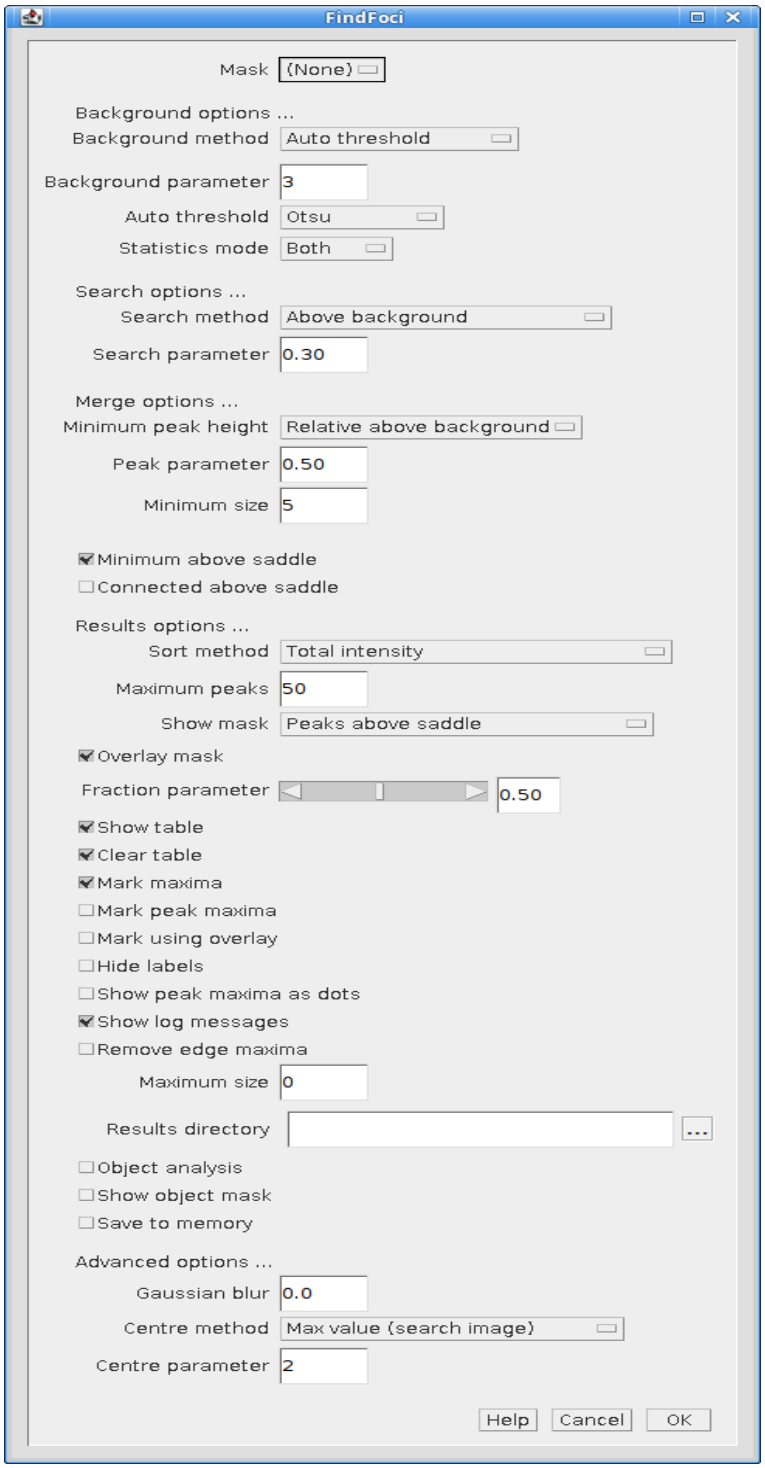
When using a blur on an integer image the result is stored as an integer image after rounding the blurred values. Only when using a 32-bit input image is the blur stored as a 32-bit image, i.e. the blur will be smoother with 32-bit processing. This will lead to different results when using a 16-bit image with a blur and the same image converted to 32-bit before running the algorithm. The user is recommended to try both 16- and 32-bit images when using a blur to determine if there is a benefit in their case.

In practice the speed difference between the integer and floating point processing will not

impact analysis as the algorithm is fast. If you notice a significant slowdown when using 32-bit processing then please contact Alex Herbert (details are on the front page).

Plugin Interface

The FindFoci interface uses the standard ImageJ Generic Dialog. The plugin has many options but these can be divided into sections controlling parts of the algorithm. The different sections are shown in the following image and are described below.



Mask

The Mask parameter allows the user to select a mask specifying which pixels to process. Any non-zero pixel will be included in the analysis. The selection box is populated with

images that have the same width and height dimensions as the image to be processed. The mask can either have 1 slice or can be a stack with the same number of z-slices as the image to be processed.

Note that the mask can have objects defined using unique pixel values. The plugin provides the option to identify objects within the mask and then label any maxima as belonging to an object (see the Object analysis option). This allows the results to be used to count maxima in pre-labelled objects such as cell nuclei.

Background Parameters

The background parameters set the lowest intensity level used in the search for peak areas. Pixels that are below this level will not be included in the peak areas.

A simple approach would set the background to 0 (zero). However this means that all pixels in the image must be processed resulting in a slower runtime and the identification of many false peaks with insignificant heights. The plugin therefore provides different options for setting a background above zero.

The background parameters section contains four input fields: the main field (Background method) allows the selection of the background method and the other fields pass parameters to the selected method if applicable.

Background Method	Description
Absolute	The background intensity is set using the input value in the Background parameter text field
Mean	The background intensity is set using the mean of the image
Std.Dev above mean	The background intensity is set using the mean of the image plus the Background parameter field multiplied by the standard deviation of the image
Auto threshold	The background intensity is set using the Auto-threshold method. The Auto threshold option uses the Auto Threshold plugin developed by Gabriel Landini (see http://pacific.mpi-cbg.de/wiki/index.php/Auto_Threshold). The plugin uses various methods to partition an image into foreground and background. Only the foreground pixels are then used in the peak finding algorithm. In most cases the Otsu method provides a fast and robust partitioning.
None	The background intensity is set as 0. Equivalent to using Absolute with a value of zero

The Statistics mode option is only relevant when using a selected region of interest (area ROI) on the image. In this case the algorithm will only search within the ROI for maxima. However the automatic background methods can use the pixel values from inside, outside or both inside and outside the ROI to set the background level. This is controlled using the Statistics mode option.

Search Parameters

The search parameters control how far the algorithm expands local maxima into peak regions. This can be used to reduce the area of peaks on an image to only part of the peak

above the background. In an image that has hundreds of peaks this can make the peaks easier to view distinctly in the output image.

The section contains two input fields: the main field (Search method) allows the selection of the search method and the other field (Search parameter) passes a parameter to the selected method if applicable.

Search Method	Description
Above background	A region is grown until the intensity drops below the background (default method)
Half peak value	A region is grown until the intensity drops to halfway between the value at the peak (the seed for the region) and the background level. This is equivalent to using the Fraction of peak - background option with the threshold value set to 0.5
Fraction of peak - background	A region is grown until the intensity drops to: background + (parameter value) * (peak height) E.g. a Search parameter of 0.2 would grow the peak to 80% of its height above the background.

Peak Merging Parameters

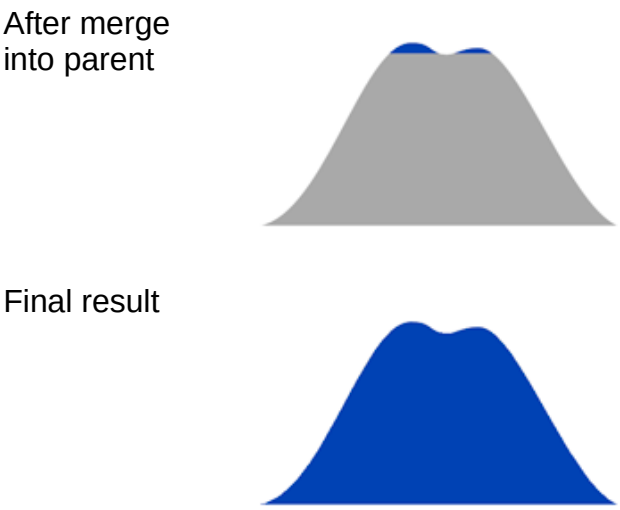
The peak merging parameters are applied after the initial pass of the algorithm has identified all the peak regions. The parameters are used to control removal of insignificant peaks from the results.

Insignificant peaks are only removed if they have no neighbour peaks. If they have a neighbour then the peak's pixels are added to the highest neighbour peak (the parent peak). This is effectively merging a sub-peak into the larger peak.

The following diagram shows the merging process:

Initial peak regions

Peaks above saddle



Peaks can be merged using two criteria: the peak height and the peak size. Merging is performed in three steps: Minimum height; Minimim size; Minimum size above saddle. Each step can be disabled.

Peak Height

The peak height sets a minimum height that the peak must be above the highest saddle point, i.e. how far the peak stand out from any neighbours.

The peak height contains two input fields: the main field (Minimum peak height) allows the selection of the method and the other field (Peak parameter) passes a parameter to the selected method if applicable.

Minimum Peak Height	**Description**
Absolute Height	The peak must be an absolute height above the highest saddle point. The height is specified by the Peak parameter
Relative Height	The peak must be a relative height above the highest saddle point. The height is calculated as peak intensity * parameter value. E.g. a value of 0.2 indicates that the peak must contains at least 20% of its total height above the saddle point
Relative above background	The peak must be a relative height above the highest saddle point. The height is calculated as (peak intensity – background) * parameter value. E.g. a value of 0.2 indicates that the peak must contains at least 20% of its height over the background above the saddle point

Note that the if the peak has no neighbours, i.e. they have no saddle points, then the peak height limit is applied relative to the background.

To disable the peak height criteria set the Peak parameter to zero.

Peak size

The Minimum size parameter sets the minimum size of a peak. If the peak is below this size then the peak is merged/removed. Set to zero to show all peaks including isolated local maxima.

The Minimum size above saddle optional restricts the peak size criteria to the pixels above its highest contact with a neighbour peak (the saddle point). This can be useful for eliminating sub-peaks which may only have a few pixels above the saddle to the parent peak but have been allocated a large number of pixels below the saddle height. Set to False to disable this filtering step.

When peaks are merged the default behaviour is to count all the pixels above the highest saddle point. Note that when merging peaks it is possible to merge a peak into another without changing the highest saddle point of the new peak, i.e. the old peak saddle was below the new peak's highest saddle point. However additional pixels added to the peak may be above the highest saddle point. This may lead to counting pixels above the highest saddle point that are not connected. Use the Connected above saddle option to ensure that only connected pixels from the peak maximum down to the highest saddle point are used to determine the size above the saddle.

Note: The original version of the FindFoci algorithm did not support the Connected above saddle option so the default is False for backwards compatibility.

Results Parameters

The results parameters control how the results will be displayed within ImageJ.

The Sort method options correspond to different columns of the results table (see the results table section). This allows the peaks to be sorted by any column in the table.

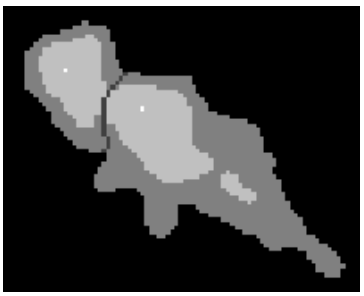
Maximum peaks specifies a limit (N) on the number of peaks that will be reported. Peaks are ranked using the specified sort method and then the top N are selected.

The Show mask parameter sets the type of output image to display. The image will be displayed in a image window with the original image title plus 'FindFoci'. If this window exists (i.e. the plugin has already been run on the input image) then the plugin will update the existing window. Therefore you should save your result mask image if you do not wish it to be overwritten.

The mask is shown as a greyscale image with value 0 corresponding to no peak region. ImageJ's max display settings are adjusted so that the highest value is white. In all output mask images the peak maxima (X,Y,Z coordinates) are shown as white pixels.

The mask options are shown below:

Result	Description
None	No mask image is shown

Peaks	The mask image contains all the pixels in the peak regions. Peak regions are assigned a value that corresponds to their ranking. The highest ranked peaks are shown as the lightest colours. The mask value for each peak is shown in the results table 
Threshold	The mask image contains all the pixels in the peak regions. Each peak region is thresholded using the auto-threshold method. This shows the high parts of the peak in a light shade and the low parts in a dark shade. The saddle pixels between peaks are shown in a dark grey 
Peaks above saddle	The mask image contains all the pixels in the peak region above the highest saddle point. Peaks are coloured as the Peaks output above 
Threshold above saddle	The mask image contains all the pixels in the peak region above the highest saddle point. Peaks are coloured as the Threshold output above 

Fraction of intensity	For each peak, this method cumulatively sums the pixels in descending intensity until a fraction of the total intensity is reached. All remaining pixels are not displayed. Requires setting the Fraction parameter
Fraction height above saddle	Show only the pixels within the peak that are above a fraction of the height above the saddle. If the fraction parameter is zero this will output the same mask as the Peaks above saddle option. Requires setting the Fraction parameter

There are several flags for the results parameters as shown below:

Overlay mask	Draw the calculated mask on the original image using an overlay. The overlay is yellow at 50% opacity. This allows the user to view the pixels that have been used to create the mask. The overlay can be remove using Image>Overlay>Remove Overlay
Fraction parameter	Define the parameter for the Fraction... mask options
Show table	Show the results table. This contains all of the information on the peak regions including height, centre, area and intensity. Note that if a results table is already present the results will be appended
Clear table	Select this option to remove any results already in the results table
Mark maxima	Label the maxima on the input image using ImageJ point ROIs. Existing ROIs will be deleted. Note: The point ROI will be associated with the correct slice when using input images with more than 2 dimensions. Double clicking the point ROI button on the ImageJ toolbar will open the Point Tool options. Use Show on all slices to display the points on every slice in the stack
Mark peak maxima	Label the maxima on the result mask image using ImageJ point ROIs
Mark using overlay	Label the maxima using an overlay. The default uses an ImageJ ROI. Using an overlay prevents destroying any ROI on the image, for example an area ROI used to restrict the analysis region. This only applies if Mark maxima/Mark peak maxima are selected
Hide labels	Remove the number labels from the ImageJ point ROIs that are added by Mark maxima and Mark peak maxima. This is useful when the labels overlap other closely spaced foci.
Show peak maxima as dots	If this option is selected then the pixel location at the centre of the maxima will be given the maximum value displayed in the mask (this can be used instead of Mark peak maxima). Disable this if the mask will be used for region labelling based on pixel value

Show log messages	Allow the plugin to write messages to the ImageJ log window. This provides additional information on the peak finding algorithm
Remove edge maxima	Removes any maxima from the results that contain pixels at the image border. This leave only peaks that are entirely within the image
Max Size	Removes any maxima from the results above a maximum size. Note: This is applied to the final results and does not affect the peak merge algorithm
Results directory	Specify a valid directory to save the results
Object analysis	If a mask was used then this option will search for objects within the mask. All non-zero pixels are potential objects. Objects are formed by joining touching pixels with the same value. Each distinct collection is assigned an object ID. The original mask value for the object is stored. The results table will contain the parent containing object and the original mask value for each identified maximum
Show object mask	Select this option to display an image of the mask objects calculated by the Object analysis. Pixels will have the object ID as their pixel value
Save to memory	Save the results to memory. Results are named using the image title and the channel and frame used for the analysis. This option allows other ImageJ plugins to use the results

The Results directory parameter allows the user to specify a directory to use to save the results. The directory must exist otherwise the plugin ignores the parameter. If the directory exists then the plugin will create 3 files named with the prefix FindFoci-[Image Short Title]-YYYYMMDD_HHmmss (the datestamp ensures that files will not be over-written). The 3 files are:

1. .xls: A tab-delimited text file containing the results as written to the result table (the Show table option is not required)
2. .roi: An ImageJ ROI file. This uses ImageJ's ROI format. It can be loaded into ImageJ using the ROI manager
3. .params: A text file containing all the parameters used to run the plugin including the original input image filename. This can be used as input for the FindFoci Batch plugin

Advanced Parameters

Gaussian Blur

The Gaussian blur parameter is used to apply a Gaussian blur to the input image before running the peak finding algorithm. This is very useful for noisy images since it can eliminate peaks that are single pixel local maxima by smoothing the image content. This dramatically improves the algorithm speed due to the smaller number of merging steps required to eliminate all the sub-peak regions. Note that if a Gaussian blur is applied the

final results are still calculated using the original image (e.g. the total intensity values).

FindFoci supports integer (8- or 16-bit) images and true floating point 32-bit images. When using a blur on an integer image the result is stored as an integer image after rounding the blurred values. Only when using a 32-bit input image is the blur stored as a 32-bit image. This will lead to different results when using a 16-bit image with a blur and the same image converted to 32-bit before running the algorithm. The user is recommended to try both 16- and 32-bit images when using a blur to determine if there is a benefit in their case.

Centre Method

The Centre method contains two input fields: the main field allows the selection of the method used to calculate the centre of the peak and the other field passes a parameter to the centre method if applicable.

Centre Method	**Description**
Max value (search image)	Define the peak centre using the highest pixel value of the search image (default). In the case of multiple highest value pixels, the closest pixel to the geometric mean of their coordinates is used
Max value (original image)	Re-map peak centre using the highest pixel value of the original image. If no blur is applied then this matches the result using the search image
Centre of mass	Re-map peak centre using the peak centre of mass (COM). The COM is computed within a given volume of the highest pixel value. Only pixels above the saddle height are used to compute the fit. The volume dimensions are specified using 2xN+1 where N is the Centre parameter
Gaussian	Re-map peak centre using a Gaussian fit. Only pixels above the saddle height are used to compute the fit. The fit is performed in 2D using a projection along the z-axis. If the centre parameter is 1 a maximum intensity projection is used; else an average intensity project is used. The z-coordinate is computed using the centre of mass along the projection axis located at the xy centre. Note: Gaussian fitting functionality is experimental. This method rarely produces better results than the simpler methods above since a large peak is required for a fit. In addition the current method does not fit the correlation between axes and so poorly models elliptical peaks. If no fit is possible then the peak is not remapped from the 'Max value (search image)' result

Results Table

The results table contains details about each peak region. The table contains the following information:

Result	**Description**
Peak #	The number of peak

Mask Value	The value used for the peak in the mask image (applies to the mask Peaks option)
X, Y, Z	The peak maxima location in X, Y and Z coordinates
Size	The number of pixels in the peak
Max	The maximum intensity value of the peak
Total	The total intensity of pixels in the peak
Saddle value	The intensity value of the highest saddle point
Saddle Id	The Id of the neighbour peak sharing the saddle
Abs.Height	The height of the peak above the highest saddle
Rel.Height (>Bg)	The relative height above the background: (absolute height) / (max value – background) i.e. The amount of the peak height that is above the saddle point
Size > saddle	The number of pixels above the highest saddle
Total > saddle	The total intensity above the highest saddle
Av	The average intensity of the peak
Total (>Bg)	The total intensity in the peak above the background level
Av (>Bg)	The average intensity in the peak above the background level
Total (>Min)	The total intensity in the peak above the minimum image value
Av (>Min)	The average intensity in the peak above the minimum image value
% Signal	The total intensity divided by the total image intensity
% Signal (>Bg)	The total intensity above background divided by the total image intensity above the background
% Signal (>Min)	The total intensity above minimum image value divided by the total image intensity above the minimum image value
Signal / Noise	The peak maximum value divided by the background
Object	The parent containing object identified by the mask object analysis. If Object analysis was not selected this will be zero
State	The original mask value of the parent containing object identified by the mask object analysis. If Object analysis was not selected this will be zero

Sorting

Note 1:

Although not all the columns are listed in the Sort method parameter, some values are derived from others and it should be possible to order the results using any column describing the maximum properties.

Note 2:

It is not possible to sort using the Object or State columns since the object analysis is performed after selection of the top maxima.

Note 3:

Certain values based on intensity are sensitive to either the background or, in the case of 32-bit processing, if the image contains negative values:

Result	Sensitive to Background	Sensitive to Negative Values
Total		Y
Total (>Bg)	Y	Y
Av		Y
Av (>Bg)	Y	Y
Total > saddle		Y

If the result is senstive to the background then the user should note that changing the background method and/or parameter may reorder the results. This occurs when peaks cover a very different number of pixels, thus changing the background (Bg) will alter the total added to each peak by $Change = Count \times \Delta Bg$. Thus a large peak may be re-ranked above a smaller peak, or vice versa.

Negative Values

If the sort result is sensitive to the image having negative values and the image contains negative values (this is only possible when performing 32-bit processing) then the sort will be incorrect. For example the total intensity of a large peak may sum to a large negative value, and a small peak a small negative value. The small peak will be sorted before the large peak. This is unwanted and so the plugin will write a warning to the ImageJ log:

```
WARNING: Image minimim is below zero and the chosen sort index is sensitive to
negative values
```

In this case the user is recommended to choose another result for the sort, for example Total (>Min) or Max.

FindFoci GUI

ImageJ plugin that finds areas of maximum intensity in 2D and 3D images. The Graphical User Interface (GUI) passes the selected parameters to the FindFoci plugin for processing.

Features

- Permanent GUI within ImageJ
- Allows different images to be selected
- Runs the FindFoci algorithm to identify peak regions
- Support live preview of the results
- Supports the ImageJ macro recorder

Note that the plugin supports the ImageJ macro recorder for the FindFoci command. This means that a user can record their actions with the FindFoci GUI within a macro and it will successfully record the execution of the FindFoci plugin.

Plugin Interface

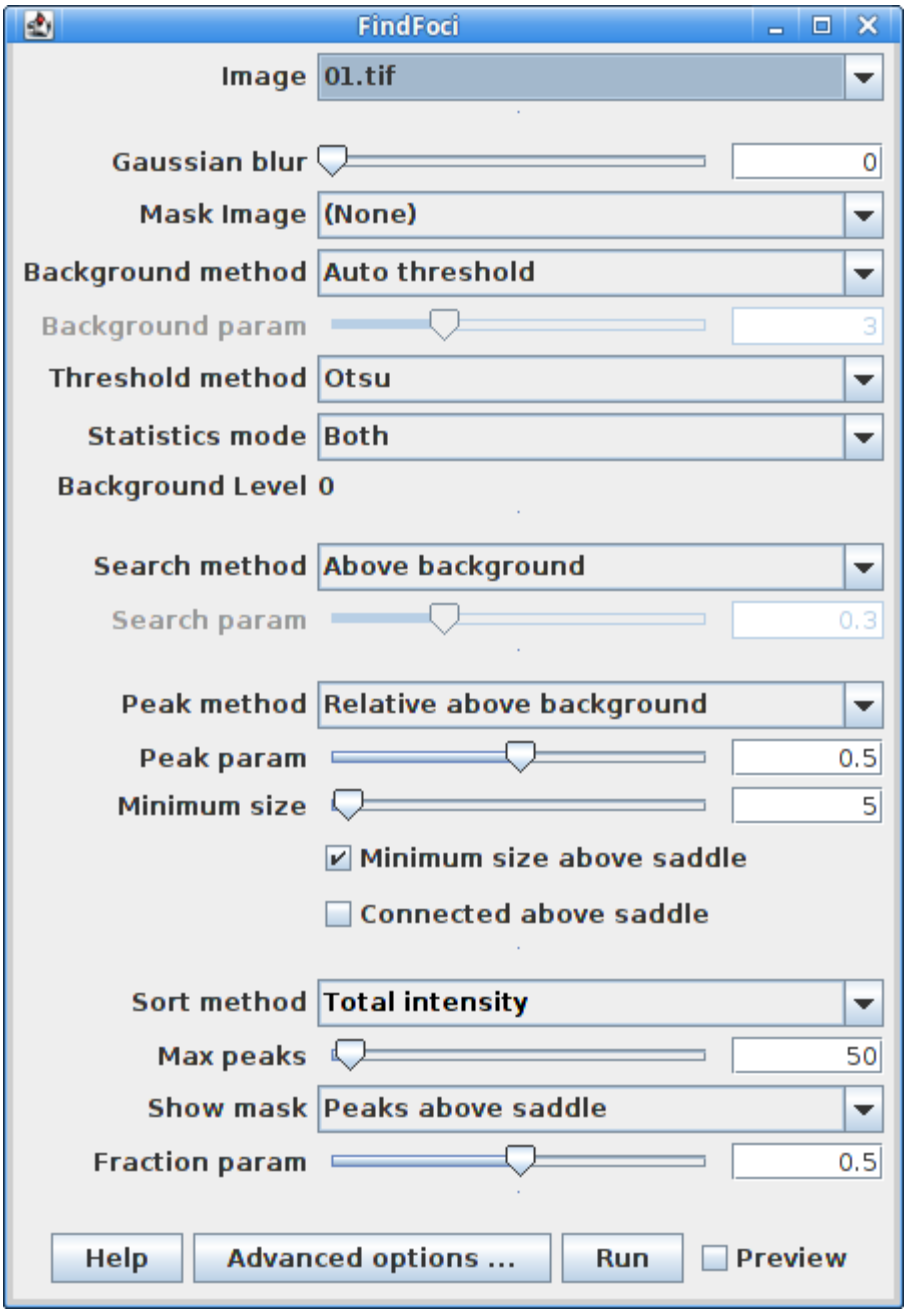
The FindFoci GUI uses a Java frame within the ImageJ application. The plugin supports all the options of the FindFoci plugin. Full details of the parameters can be found in the FindFoci Plugin Interface section.

Since the interface is not limited by the layout of the standard ImageJ dialogue the design has been altered. The window is smaller and several less used options are accessed by clicking the 'Advanced options …' button. The smaller size makes the plugin a better option to use on monitors with a low resolution.

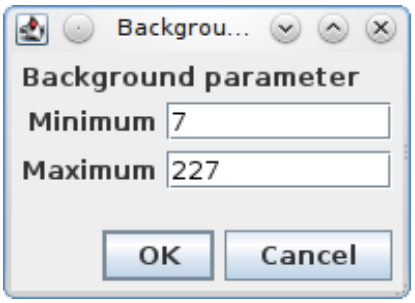
The interface has a drop-down menu that is used to select the image. This allows the user to quickly try the same parameters on different images that are open. The image list is populated with the currently open 8-bit, 16-bit and 32-bit greyscale images. Any images with the title ending in 'FindFoci' (i.e. previous results) are also ignored.

The interface has been designed to only allow the user to set the options that are relevant. For example it is only possible to set the Threshold method when the Background method is set to Auto threshold. This makes it easier for a user to set the parameters for the algorithm.

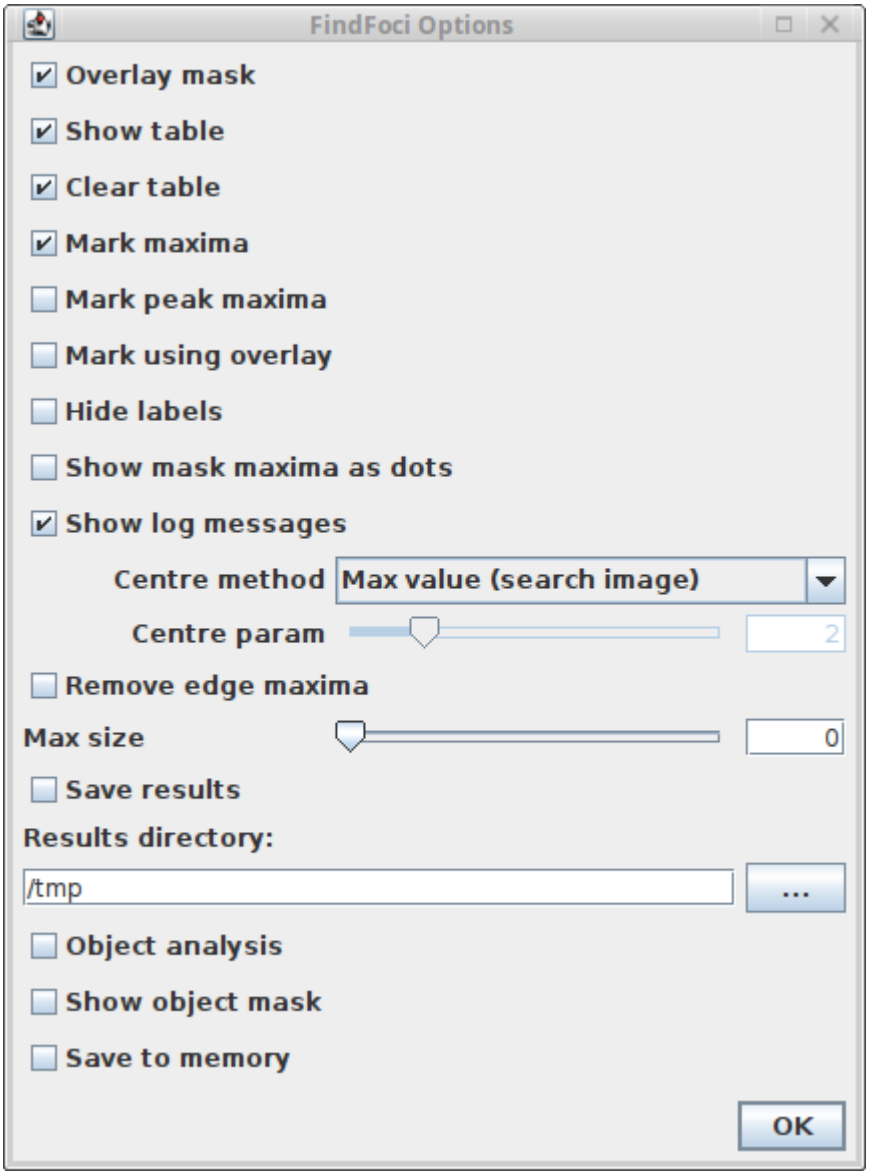
The following image shows the interface for the FindFoci GUI:



The interface has sliders to adjust the numeric parameters. The parameter can also be entered using the input text box. The range for the sliders are set to a default scale depending on the parameter. For example the background parameter is limited to the minimum and maximum values found in the image. The range for the slider can be updated by double-clicking the slider. This will display a dialog where the range can be updated:



Note: To save space within the GUI not all the options from FindFoci are directly available. Clicking on the Advanced options … button will open a second window where more parameters are available as shown below.



Preview

The FindFoci GUI interface has a preview option. If this option is checked then the plugin will automatically update the results based on any changes to the parameters. The plugin stores intermediate results in memory. When a parameter changes the plugin identifies the stages of the calculation that must be re-run and updates the results.

The preview provides the ability to change the parameters and quickly see the effects on the output mask and results table. However since the full FindFoci plugin is not run there is no output to the ImageJ log window and the command is not recorded to the ImageJ macro recorder. Only the Run button supports these options.

The Overlay mask and/or Mark maxima options are useful during the preview as they show the foci on the original image. The Mark using overlay option allows the maxima to be drawn on the image without destroying any ROI uses to restrict the analysis area. The Hide labels option is useful to turn off the point ROI number labels when the number of foci is large.

Note

The preview option requires a large amount of memory to store the intermediate results. It may not be suitable for large stack images depending on the configuration of your system. The memory used by the preview is released when the preview option is unchecked. This method can also be used to reinitialise the preview in the event that it has stopped responding to parameter changes. The preview will also need to be reinitialised if a different ROI is selected for analysis. This is because the plugin is not alerted when an image ROI is updated and so will be operating on the originally selected input pixels.

Java Requirements

The FindFoci GUI uses the BeansBinding Java framework. More details can be found in Appendix 1: BeansBinding Java framework.

FindFoci Batch

ImageJ plugin that runs the FindFoci algorithm on all the images in a chosen directory. Results are saved to an output directory.

Features

- Batch processing of images using FindFoci
- Auto-detection of mask images
- Parameters are specified using an input file
- Saves the results to an output directory
- Multi-threaded processing

Overview

The FindFoci Batch plugin can be used to speed up the analysis of large numbers of images. All the images in a single directory can be processed with the same parameters and the results recorded to an output directory.

The input images can be any format that is read by ImageJ. The output images (if selected as an output option) will use the TIFF format.

Create the FindFoci parameters

The batch plugin requires all the parameters for FindFoci to be specified in a file. This can be done using three methods:

Record the options using the ImageJ Macro Recorder

1. Start the ImageJ macro recorder using 'Plugins > Macros > Record...'
2. Run the FindFoci plugin on an example image with the desired parameters
3. The ImageJ Recorder will save the plugin name and options in the recorder using the following entry:
   run(“FindFoci”, “[options ...]”);
4. Copy the options string from the Recorder window and save it to a text file. The options should be saved to a single line in the file

The saved macro options can be used as the FindFoci parameters file.

Save the options to file using FindFoci

1. Start the FindFoci plugin on an example image with the desired parameters
2. Ensure that you provide a valid path to a results directory
3. Run the plugin

OR

1. Start the FindFoci GUI plugin
2. Select an example image and set the desired parameters
3. Click the 'Advanced options...' button
4. Ensure the 'Save results' checkbox is selected and provide a valid path to a results directory
5. Click the 'Run' button to run the FindFoci plugin

The FindFoci plugin will save a file to the results directory containing all the parameters needed for the plugin. The file will be named FindFoci-[ImageTitle]-YYYYMMDD_Hhmmss.params. The params file can be used as the FindFoci Batch parameters file. The file and file extension can be renamed if desired.

Save the options to file using the FindFoci Optimiser

1. Run the FindFoci Optimiser plugin on an example image
2. Ensure that you provide a valid path to a results file
3. The optimiser will save the top scoring parameters to file with the suffix '.params'

Note that if the FindFoci Optimiser is run in multi-image mode then the parameters file will be placed in the output directory and named 'all.params'.

Note on the parameters

The FindFoci Batch plugin will auto-detect the type of parameters file that is selected. Any line in the file with an equals ('=') character will be read. If the file contains only one line then it is assumed the file contains the macro options from the ImageJ Recorder. Otherwise the file should be a valid params file with key=value pairs for all the parameters, one pair on each line.

The FindFoci Batch plugin will ignore the entries for:

- results_directory: This is specified in the FindFoci Batch dialog
- image: The image is auto-detected from the input directory
- mask: The mask is auto-detected using filenames
- show_table: The plugin will not produce a table of results as the results are always saved to a .xls file in the results directory

Custom channel, time frame and z-slice parameters

By default the algorithm will find foci in the entire z-stack of the first channel and time frame.

To process a specific channel and/or time-frame the user will have to specify this using the parameters file. It is also possible to specify a single z-slice for processing. In this case the slice will be extracted from the z-stack before processing.

Custom dimension processing can be specified by adding entries to the parameters file as follows:

Option	Description
Image_C	Image channel to process. Default = 1. This value is only used if it is less than or equal to the number of channels in the image.
Image_Z	Image z-slice to process. Default = 0. This value is only used if it is less than or equal to the number of slices in the image.
Image_T	Image time-frame to process. Default = 1. This value is only used if it is less than or equal to the number of frames in the image.
Mask_C	Mask channel to process. Default = 1. This value is only used if it is less than or equal to the number of channels in the mask.

MASK_Z	Mask z-slice to process. Default = 0. This value is only used if it is less than or equal to the number of slices in the mask.
MASK_T	Mask time-frame to process. Default = 1. This value is only used if it is less than or equal to the number of frames in the mask.

Not all entries must be specified. If an entry is missing or invalid (i.e. outside the image/mask dimensions) the default value will be used.

Note that if the parameters file contains the options recorded by the IMAGEJ Recorder, the entries must be added at the end of the single line, e.g.

… Image_C=2 Image_T=3

Adding entries on new lines will cause the plugin to fail to recognise the Recorder options.

If the parameters file contains key-value pairs on separate lines, the entries must be added as new key-value pairs on separate lines, e.g.

...

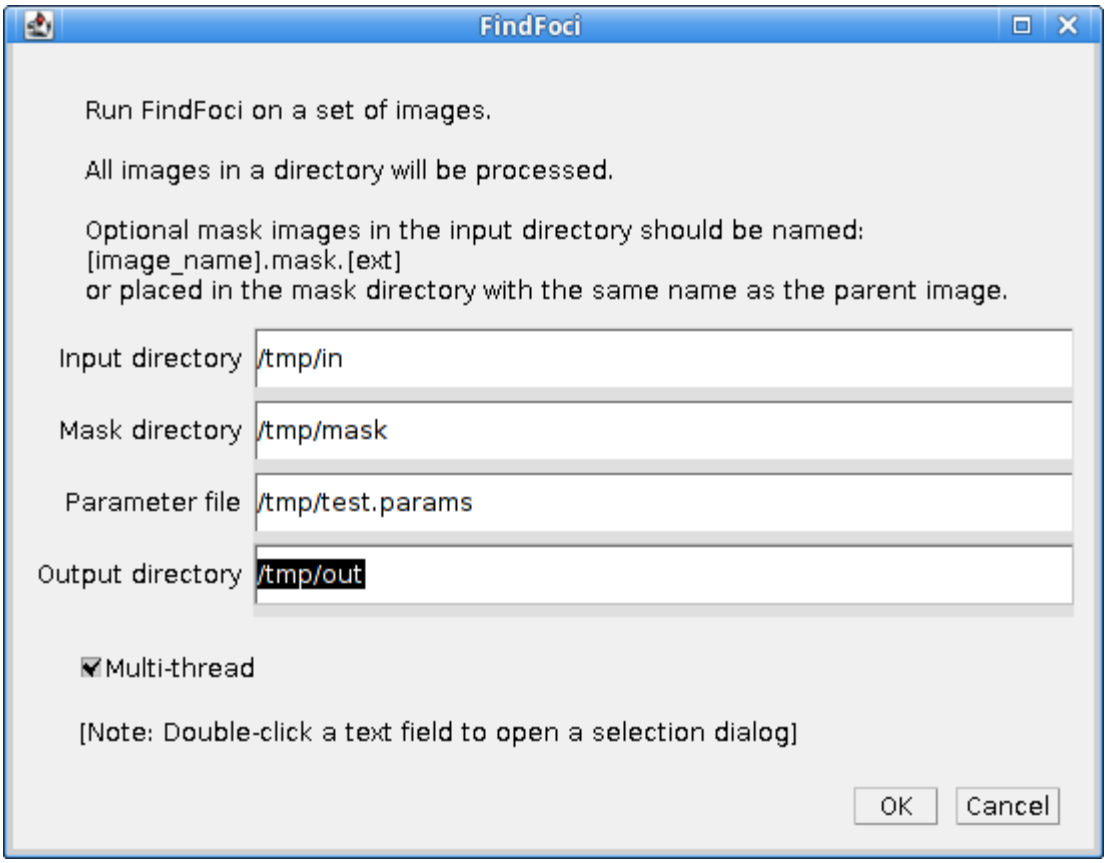
Image_T = 2

Image_T = 3

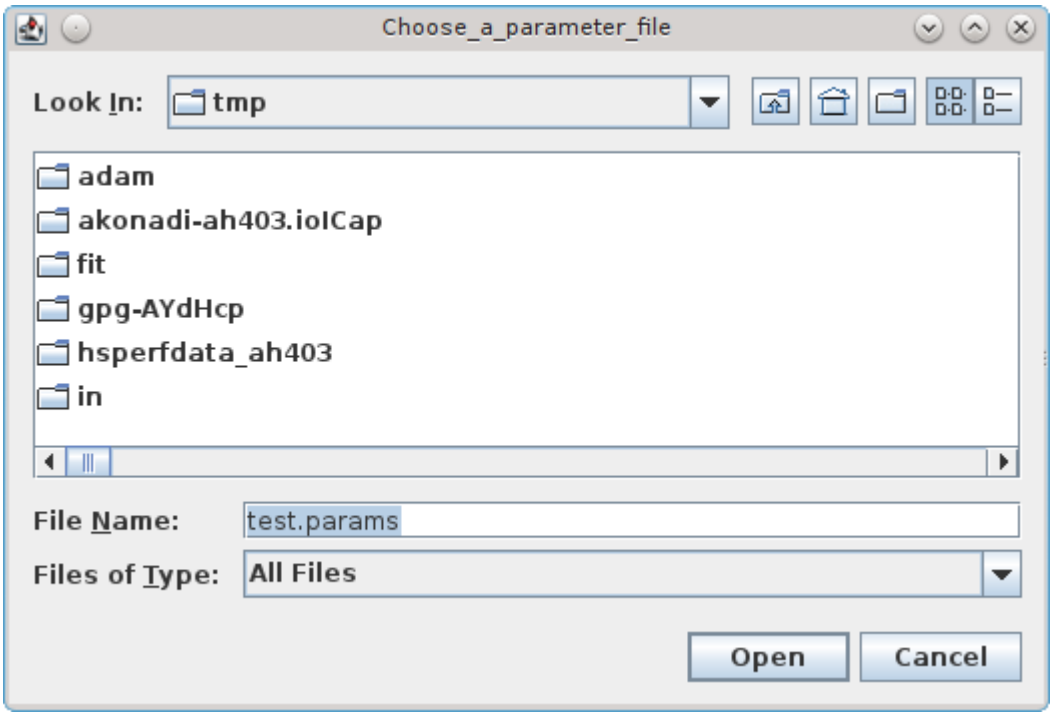
If the parameters file was generated by FINDFOCI then these options will already be present. They can be removed or updated as necessary. These entries are not recorded by the FINDFOCI OPTIMISER.

Plugin Interface

The FINDFOCI BATCH interface uses the standard IMAGEJ Generic Dialog.



The input fields must specify the full path to the directories and the parameters file. If you double-click in the text field the plugin will open a selection dialog, for example:



Parameters

Option	Description
Input directory	The directory containing the images to process. All valid images supported by ImageJ will be opened. Note that the FindFoci plugin will only work on 8 and 16 bit images so do not put 32-bit or RGB color images in the directory. Any file with the text 'mask.' will be ignored as these are assumed to be mask images. Optional mask images should be named using the image title with the text 'mask.' placed before the image extension: [image_name].mask.[ext]
Mask directory	The directory containing mask images. This parameter is optional. Mask images must be named with the same name as the parent image in the input directory. Or the images can be named using the image title with the text 'mask.' placed before the image extension: [image_name].mask.[ext]
Parameter file	The name of the FindFoci parameters file
Output directory	The output directory. Note that the plugin will overwrite files that already exist
Multi-thread	Select this option to process images in parallel. The results

	output will be the same but ImageJ log messages may be in a different order. The number of threads is taken from the ImageJ preference set in 'Edit > Options > Memory & Threads...'

The Mask directory parameter is optional. This allows the user to place the mask images in a separate directory or the same directory as the input image. The mask will be found using the following search order:

1. [Mask directory] + [Image filename]
2. [Input directory] + [image_name].mask.[ext]
3. [Mask directory] + [image_name].mask.[ext]

It should be noted that the FindFoci plugin will process the current channel and time-frame of a hyperstack image. When opening the images the channel and time-frame will be set to 1. To process a specific channel and time-frame the user will have to specify this using the parameters file, see heading 'Custom channel, time frame and z-slice parameters' in the section above.

Output

The FindFoci Batch plugin will not produce any results within ImageJ. The standard table of results or mask image will not be displayed. It is possible to see the algorithm running in the ImageJ status bar and in the ImageJ log (if Show log messages is enabled).

All the results for each image are saved to the output directory. The results will be prefixed with the image title and the channel (C), z-slice (Z) and time-frame (T) used when processing:

[Title]_c[C]z[Z]t[T]

If Z is zero then the entire z-stack was analysed. To use the original title without the CZT suffix then add the 'Original_Title' parameter to the parameters file.

The following files are produced:

File	Description
.xls	The results of the FindFoci algorithm including foci sizes and coordinates
.roi	An ImageJ ROI file that allows the foci to be loaded onto an image within ImageJ
.params	The parameters used for the analysis
.tiff	The input image saved with the marked ROI. This is only produced if the option to Mark maxima is selected
.mask.tiff	The output mask image. This is only produced if the option to Show mask is selected
.objects.tiff	The input mask image reprocessed to display the identified

	objects. This is only produced if the options Object analysis and Show object mask are selected

Note that the plugin will overwrite files that already exist in the output directory. Therefore care must be taken when selecting the output directory to be the same as the input directory. Conflicts should not be possible given the results contain the CZT position that was analysed. However existing files (including previous results) will be over-written.

Multi-thread Mode

The plugin will use the number of threads specified in the ImageJ preferences. This can be set using 'Edit > Options > Memory & Threads...'.

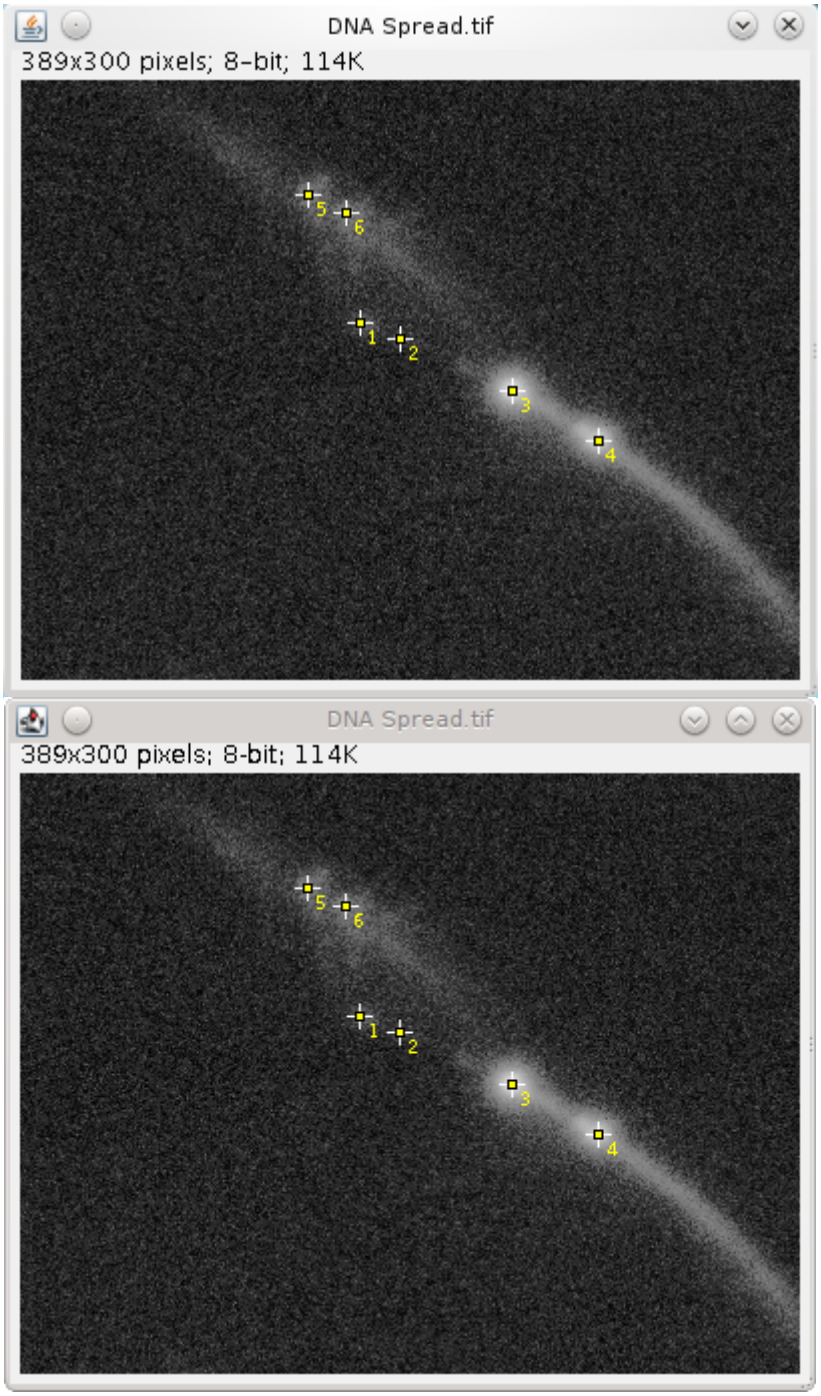
In single-thread mode messages are written immediately to the ImageJ log window. However when running in multi-thread mode all the log messages for processing a single image are stored in memory. When the processing of the image has finished the messages are recorded. This prevents log messages from processing different images to become interlaced. This does mean there will be a slight delay before progress can be followed in the log window.

The number of images processed can be followed in the ImageJ progress bar.

FindFoci Optimiser

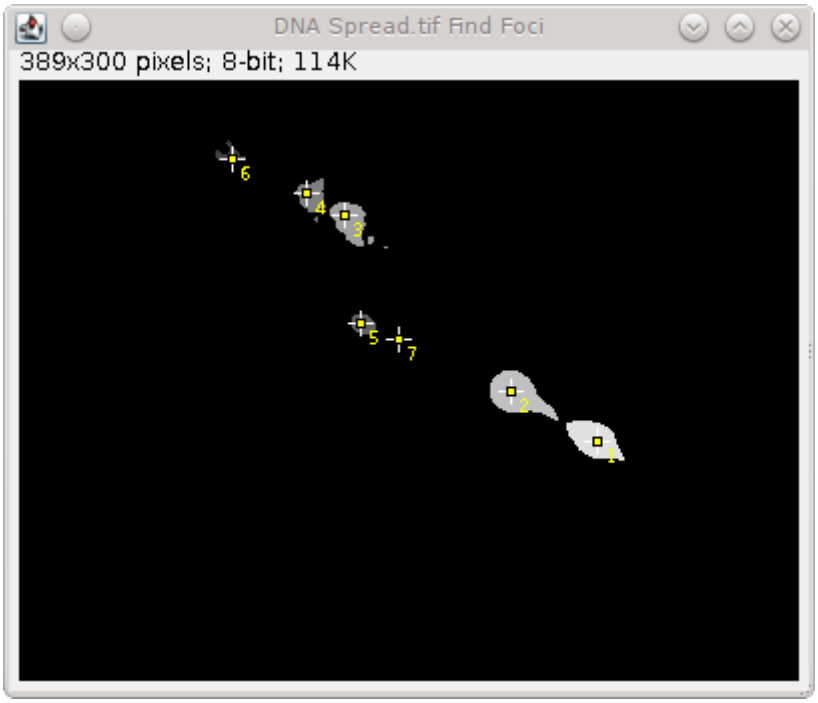
ImageJ plugin that iterates through combinations of parameters for the FindFoci plugin. Results are compared to an optimal reference image and the best parameters to match the reference result are identified.

Example Input



Input image with manually identified peaks. These are used a reference points for optimisation of the FindFoci algorithm.

Example Output



Output image showing the peaks identified with the best match to the input image.

Find Foci Optimiser Results

File Edit Font

Rank	Blur	Background method	Max	Min	Search method	Peak method
10	1.0	Std.Dev above mean (Both) : 3.50	50	9 >saddle	Fraction of peak - background : 0.00	Relative above background : 0.40
9	1.0	Std.Dev above mean (Both) : 3.50	50	9 >saddle	Above background	Relative above background : 0.40
8	1.0	Std.Dev above mean (Both) : 3.50	50	7 >saddle	Fraction of peak - background : 0.20	Relative above background : 0.40
7	1.0	Std.Dev above mean (Both) : 3.50	50	7 >saddle	Fraction of peak - background : 0.00	Relative above background : 0.40
6	1.0	Std.Dev above mean (Both) : 3.50	50	7 >saddle	Above background	Relative above background : 0.40
5	1.0	Std.Dev above mean (Both) : 3.50	50	5 >saddle	Fraction of peak - background : 0.20	Relative above background : 0.40
4	1.0	Std.Dev above mean (Both) : 3.50	50	5 >saddle	Fraction of peak - background : 0.00	Relative above background : 0.40
3	1.0	Std.Dev above mean (Both) : 3.50	50	5 >saddle	Above background	Relative above background : 0.40
2	1.0	Std.Dev above mean (Both) : 3.00	50	9 >saddle	Fraction of peak - background : 0.20	Relative above background : 0.40
1	1.0	Std.Dev above mean (Both) : 3.00	50	9 >saddle	Fraction of peak - background : 0.20	Relative above background : 0.20

Find Foci Optimiser Results

File Edit Font

Sort method	Centre method	N	TP	FP	FN	Jaccard	Precision	Recall	F0.5	F1	F2	F-beta	mSec
Total intensity	Max value (search image)	8	5	3	1	0.5556	0.6250	0.8333	0.6579	0.7143	0.7812	0.8173	2.9751
Total intensity	Max value (search image)	8	5	3	1	0.5556	0.6250	0.8333	0.6579	0.7143	0.7812	0.8173	2.9751
Total intensity	Max value (search image)	8	5	3	1	0.5556	0.6250	0.8333	0.6579	0.7143	0.7812	0.8173	2.9008
Total intensity	Max value (search image)	8	5	3	1	0.5556	0.6250	0.8333	0.6579	0.7143	0.7812	0.8173	3.0182
Total intensity	Max value (search image)	8	5	3	1	0.5556	0.6250	0.8333	0.6579	0.7143	0.7812	0.8173	2.9887
Total intensity	Max value (search image)	8	5	3	1	0.5556	0.6250	0.8333	0.6579	0.7143	0.7812	0.8173	2.9126
Total intensity	Max value (search image)	8	5	3	1	0.5556	0.6250	0.8333	0.6579	0.7143	0.7812	0.8173	2.9833
Total intensity	Max value (search image)	8	5	3	1	0.5556	0.6250	0.8333	0.6579	0.7143	0.7812	0.8173	2.9938
Total intensity	Max value (search image)	8	5	3	1	0.5556	0.6250	0.8333	0.6579	0.7143	0.7812	0.8173	3.2356
Total intensity	Max value (search image)	8	5	3	1	0.5556	0.6250	0.8333	0.6579	0.7143	0.7812	0.8173	3.2402

Result table showing the match scoring statistics comparing the identified peaks to the reference image for each set of parameters.

Features

- Enumerates combinations of parameters for the FindFoci plugin
- Compares the results to a reference image
- Reports the best parameters to identify the reference peaks
- Save the results table to file
- Staged processing allows fast enumeration of parameters
- Supports 2D/3D foci

Overview

The FindFoci Optimiser provides the ability set multiple options for each parameter of the FindFoci plugin. All the combinations are run and the results compared to a reference image to identify the best combination.

The options for each parameter are specified by the user. In the case of selection inputs and flags it is possible to specify more than one option for each parameter. In the case of numeric inputs it is possible to select a lower limit, upper limit and increment to use, e.g. 0 to 10 in 2 step intervals. The optimiser then runs the FindFoci plugin for all the combinations of parameters.

Since the FindFoci algorithm can be broken down into stages (background thresholding, maxima identification, peak growing and peak merging) it is possible to runs only the parts of the algorithm that differ when changing a subset of the parameters. This allows the optimiser to efficiently run thousands of combinations of parameters.

Each run of the FindFoci plugin is compared to a reference image with marked maxima and scored using various metrics. The final results are ranked and the best parameters identified.

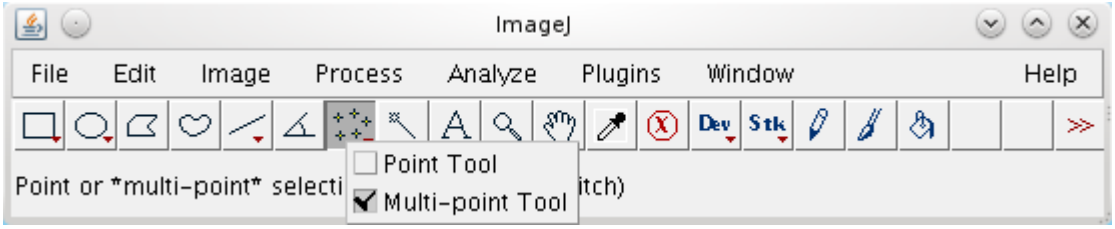
Note that the optimiser runs on a single image. It is possible to run the optimiser on multiple images and combine all the results using the Find Foci Optimiser Multi-Image plugin. This plugin is described in a later section.

Stage 1: Create Reference Foci

The optimiser requires a reference set of foci to set a benchmark for the algorithm. The aim is to match the reference foci as closely as possible. Reference foci can be created using two methods: (a) Using the ImageJ ROI tool to mark the image; or (b) Using a text file. 3D coordinates are supported using either method.

Mark Foci on the Image

Select the Multi-point tool by right-clicking on the Point Tool option in the toolbar and selecting the Multi-point Tool option.



Open a reference image. Click the image to mark a maxima. Points can be moved by hovering over an existing point and dragging them. Points can be deleted by holding down the Alt key and clicking them.

Once the image maxima have been labelled you can save the image as a TIFF. ImageJ will save the ROI points to the image and they will be shown when the image is next opened.

Note: The ImageJ ROI tool supports 3D coordinates. Each point is shown only on the slice where the image was clicked. To make it easier to see which points have been selected when scrolling through a z-stack use the point tool optionsto show the points on all slices. Double clicking the point ROI button on the toolbar will open the Point Tool options. Use Show on all slices to display the points on every slice in the stack. This option will update the ROI immediately allowing the setting to be toggled when marking the ROI.

To label foci on a 3D image with multiple slices can also use a text file. The Point Extractor plugin can be used to assist in this task by saving marked ROI points to file (see section Point Extractor).

Specify Foci in a Text File

The reference foci can be loaded from a text file. This supports both 2D and 3D coordinates.

The file must contain records with X, Y and optionally Z coordinates, i.e.

x y [z]

If no Z-coordinate is present for any record then the points will be loaded as 2D. Each field can be delimited by either a comma, space or tab character. This file can be created manually or the Point Extractor plugin can be used to create the file from points on an image or from ROIs within the ImageJ ROI Manager.

File Location

The text file for the image must be placed in the same directory as the image that will be analysed by the FindFoci Optimiser. The file should be named using the same filename as the image but using the suffix '.csv'; '.xyz'; or '.txt'. E.g.

/myFiles/image1.tiff

/myFiles/image1.csv

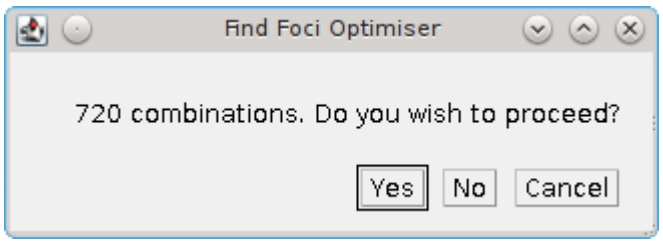
Loading the Foci

When the FindFoci Optimiser is run it identifies the original source file for the image. It then searches the same directory for a coordinate text file and attempts to load the foci. If it cannot load the foci from file then the foci are loaded directly from the image using the marked point ROI. This means that any foci marked on an image are ignored if a coordinate file exists for that image. Also note that the z-component of 3D foci loaded from file is ignored if the corresponding image is not a 3D stack.

Stage 2: Run the Optimiser

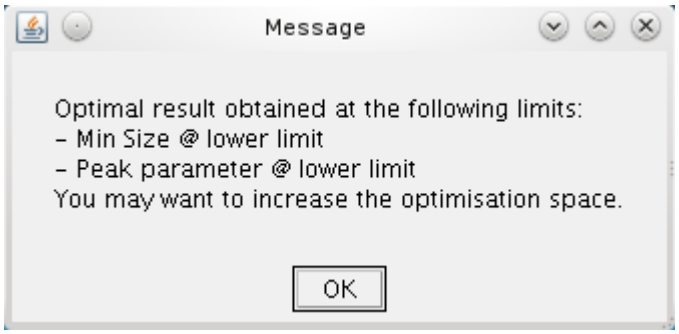
Select the different parameters for the optimiser to enumerate.

Click OK and the optimiser will show a dialogue containing the number of combinations. You can then select to continue or cancel to stop the optimiser and change the parameters.



If you run the optimiser a progress bar will show in the ImageJ window that tracks the number of combinations currently completed. When the optimiser is finished a results window will appear showing the top N results. The number of results can be configured in the optimiser dialogue.

If the best result is obtained using a numeric parameter that lies on an extreme of the test range then the plugin will show a warning message:



This allows the user to increase the range to ensure that the optimal parameters are identified.

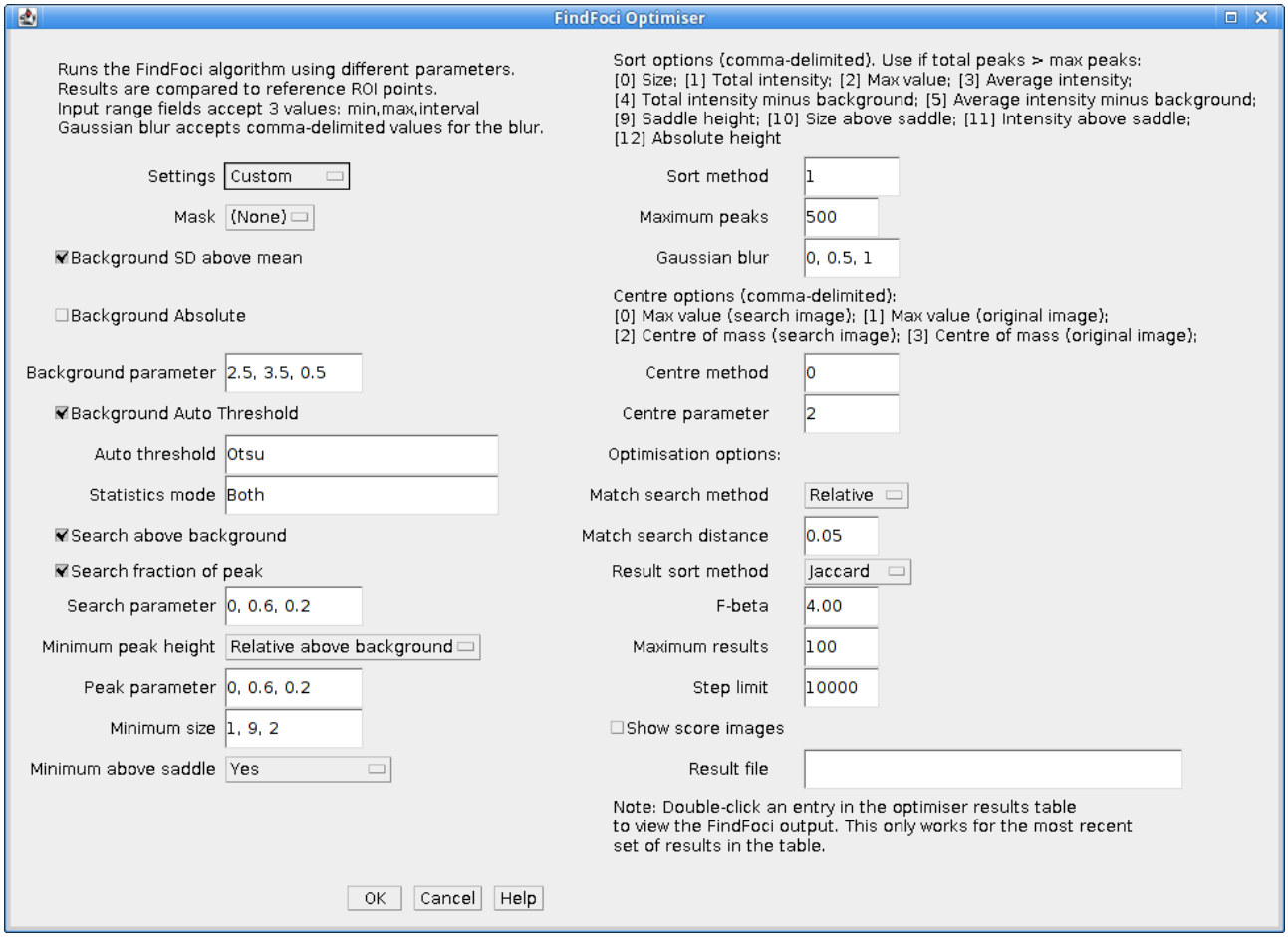
Result Assessment

The optimiser scores the results of the FindFoci plugin by comparing the locations of the identified maxima to the reference image. Maxima are deemed to be correct if they are within a set distance of a reference point. The cut-off distance is configured in the plugin options. The default is 5% of the image width/height, whichever is smaller. Note that the distance will be computed in 3D if reference foci have been loaded from a 3D coordinate file. For calibrated images the z-distance is scaled relative to the pixel unit distance of the x-dimension. For example an image with 110x110x250nm voxels will scale the z-distance by 250/110.

The maxima assignment process is iterative. The maxima closest to any reference point is assigned as a match and the pair removed from the search. This is iterated until no more reference points exist, or all maxima are too far from a reference point to be a match. The results are then scored using the metrics detailed in Appendix 2: Binary Scoring Statistics.

Plugin Interface

The FindFoci Optimiser interface uses the standard ImageJ Generic Dialog. The plugin has many options as shown in the following image.



Settings Presets

To simplify configuration of the optimiser there is a Settings drop-down item that allows the user to choose from various presets. The following presets are available:

Preset	Combinations	Description
Custom	Variable	This preset is selected when the user modifies any of the fields.
Testing	12	A small number of combinations for testing
Default	720	The default combinations to try a range of parameters that are applicable to many images
Benchmark	19,800	A set of combinations that can be used as an initial search of the entire range of the algorithm parameters

It is possible to select any of the presets as a starting point for parameter configuration. When any setting is changed (except the results filename) the Settings option will change to Custom. It is possible to revert back to the preset by selecting it again from the Settings

drop-down. In this event the current custom settings will be saved so that the user can switch back to their customised settings using the drop-down. However if any changes are made to the preset values then the saved custom settings will be discarded (since a new custom settings will be started).

Using custom settings is the best way to search for the optimum FindFoci parameters since each image analysis is unique. The following sections explain how to configure the optimiser parameters.

Initialisation Parameters

The FindFoci algorithm must be initialised with an image and optionally a mask identifying which pixels that are to be processed. The pixels are used to calculate image statistics and a background level for the peak finding algorithm. The optimiser will run the FindFoci plugin for each background option that is enabled. There are addition fields that allow parameters to be passed to the background method and control the pixels used for the statistics.

Initialisation Option	Description
Mask	Optionally select a mask to define the pixels to process. Only images with the same dimensions as the input image are listed. Any non-zero pixel is included in the analysis
Background SD above mean	Use the SD above Mean background method. It is not allowed to use this at the same time as the Absolute method
Background Absolute	Use the Absolute background method. It is not allowed to use this at the same time as the SD above Mean method
Background parameter	Set the lower limit, upper limit and increment for enumerating the background parameter passed to the background method. This applies to either SD above mean or Absolute
Background Auto Threshold	Use the Auto Threshold background method
Auto threshold	The method for performing the Auto Threshold calculation. The text field supports a comma-delimited list for multiple options
Statistics mode	When a mask image is used it is possible to set the region used to calculate the image statistics: Inside – Use the pixels inside the image mask Outside – Use the pixels outside the image mask Both – Use all the pixels in the image When no mask image is used this option is ignored

Search Parameters

The optimiser will run the FindFoci plugin for each search option that is enabled. There is an addition field that allows a parameter to be passed to the search method.

Search Option	Description
Search above background	Use the Search above background method
Search fraction of peak	Use the Fraction of peak – background method
Search parameter	Set the lower limit, upper limit and increment for enumerating the search parameter passed to the fraction of peak method

Peak Merging Parameters

The optimiser will run the FindFoci plugin for each combination of the following peak merging options.

Peak Merge Option	Description
Minimum peak height	Set the method for setting the minimum peak height
Peak parameter	Set the lower limit, upper limit and increment for enumerating the parameter passed to the minimum peak height method
Minimum size	Set the lower limit, upper limit and increment for enumerating the minimum size parameter passed to the peak merge method
Minimum above saddle	Set the option for limiting the peak size to above the saddle, including the option to compute using only connected pixels above the saddle. Select Yes; Yes – Connected; No; or All

Results Parameters

Option	Description
Sort Method	Set the method used to sort the peaks before extracting the top N. Multiple options can be specified using a comma-delimited list
Maximum peaks	Set the maximum number of peak results

Extra Parameters

Control other options available within the FindFoci plugin.

Option	Description
Gaussian blur	Set the size of the blur to apply before running the Find Peak algorithm. Multiple options can be specified using a comma-delimited list
Centre method	Set the method used to determine the centre of the peak. Multiple options can be specified using a comma-delimited list

Centre parameter	Set the lower limit, upper limit and increment for enumerating the parameter passed to the centre method

Optimisation Parameters

The FindFoci Optimiser plugin has several options for controlling how the FindFoci results are calculated and ranked.

Option	Description
Match search method	Choose the method for specifying the distance for two peaks to be a match: Relative – Use a distance relative to the shortest edge (width or height) in the image Absolute – Use an absolute distance in pixels
match search distance	Set the distance limit used to determine if a peak maxima and a reference point are a match. This value is specified as a fraction of the image short edge, e.g. 0.05 for a 600x400 pixel image would be 20 pixels (400 x 0.05), or an absolute value in pixels (see Match search method)
Result sort method	Specify the metric used to sort the results
F-beta	Specify the beta parameter used to calculate the custom F-score
Maximum results	Specify the maximum number of results to include in the result table
Step limit	The optimiser will not run if the total number of combinations (steps) is above the Step limit. This parameter can be used to increase the potential optimisation space

Additional Parameters

The FindFoci Optimiser plugin has several options for controlling how the FindFoci results are calculated and ranked.

Option	Description
Show score images	If enabled the optimiser will show three duplicates of the input image with overlay points marking the True Positives, False Positives and the False Negatives. The points are shown using an overlay as this can label points to a specific slice of a 3D stack. (Point ROI would be visible through the entire stack.)
Result file	Provide the prefix for the results files. The prefix should be a full path to a valid directory and include a filename prefix, e.g. /tmp/FindPeaks Results files will have a suffix added corresponding to the result: .results.txt – Contains the optimiser results .points.csv – Contains the peak coordinates from the best

	.params – Contains the FindFoci parameters for the best result (this can be input to the FindFoci Batch plugin)

Range Limit Fields

The plugin has several text fields that accept a lower limit, upper limit and interval. The numbers must be comma-delimited. If three numbers are present then the plugin will assume that they are lower, upper and interval. Otherwise only the upper limit is set using the first number in the text field. This is equivalent to using a single value and not a range of values.

The following fields allow range limits:

- Background parameter
- Search parameter
- Minimum size
- Peak Parameter
- Centre parameter

The plugin has several text fields that accept multiple values in a comma-delimited list. The following fields allow a comma-delimited list:

- Auto-threshold
- Statistics mode
- Sort method
- Centre method
- Gaussian blur*

* The use of a list for Gaussian blur is due to the fact that changes in the blur radius are most noticeable at increasing increments, e.g. 1, 2, 4, 8, 16. In this case a range field with a linear increment is unwanted.

Output

The FindFoci Optimiser records the parameters and results of each run of the FindFoci algorithm. The best results are shown in a result table and the complete results can be saved to file. The plugin also runs the FindFoci plugin using the optimal parameters on the input image to show the result peaks. Further details are outlined below.

Result Table

The result table shows the top scoring results from all the parameter combinations. The results are sorted using the Result sort method parameter and limited to the number specified by the Maximum results parameter. Results are presented in descending order so that the best result from the most recent analysis is shown at the bottom of the table.

In the event that two sets of parameters have an equal score then the results are ranked using a second metric. By default this is the RMSD (a measure of how close the true positives are to the answer) or if RMSD is chosen as the primary rank metric, the Jaccard.

In the event of a tie the parameters that have the most conservative settings are ranked first. The following logic is used:

- Lowest blur

- Background method: None < Auto-threshold < Std.Dev above mean < Mean < Absolute
- Lowest minimum size
- Search method: Above background < Fraction of peak - background < Half peak value
- Lowest search parameter
- Peak method: Relative above background < Relative height < Absolute height
- Lowest peak parameter
- Lowest run time

The result table shows the result rank and then columns containing the parameters. The table then shows the following scoring metrics:

Metric	**Description**
N	The number of maxima in the result
TP (True Positives)	The number of maxima in the result that match a reference point
FP (False Positives)	The number of maxima in the result that do not match a reference point
FN (False Negatives)	The number of reference points that were not identified as maxima
Jaccard	Measure of the overlap similarity between the reference and the maxima
Precision	Measure of how many identified maxima are correct
Recall	Measure of how many reference points were identified
F-score	Combined score that provides a weighted measure of the prediction accuracy: β is a weighting factor between the Precision and Recall. The F-score is calculated using β of 0.5, 1, 2 and the provided input F-bETA parameter.
Score	The score used for ranking
RMSD	The Root Mean Square Deviation (RMSD) of the true positive results from the actual. $RMSD = \sqrt{\frac{\sum d^2}{tp}}$ Where d is the distance between the predicted and actual coordinates and tp is the count of true positives.
mSec	The number of milliseconds taken to perform the calculation

Note that the metrics Jaccard, Precision, Recall and F-score all range from 0 to 1. Further details of the metrics can be found in Appendix 2: Binary Scoring Statistics.

A blank line is inserted at the end of the result table to allow easy visual separation of the

results from multiple runs of the plugin.

Showing a result

By default the FindFoci Optimiser shows the result of running the FindFoci algorithm with the best result. However it is possible to show any result by double-clicking a row in the result table. The FindFoci algorithm will then be re-run with the parameters from that row.

Note that it is only possible to view the results for the latest run of the optimiser. Parameters for prior runs are discarded from this review functionality.

FindFoci Images

Upon completion the optimiser runs the FindFoci algorithm on the input image to display the peaks. To preserve the input image ROI points a copy is made of the original. This is named with the suffix ' clone'. FindFoci is then run on the clone image to mark the peak ROI points. A mask image showing the peaks above the saddle points is also created with the suffix ' clone FindFoci'.

Score Images

If the Show score images parameter is enabled then the input image is duplicated three times. These images are then marked with the points for either the True Positives (TP), False Positives (FP) or False Negatives (FN). The points are marked using an overlay, not a point ROI. This allows marking the points using the specific slice of a 3D image. Scroll up and down through a 3D stack to view the location of the each result.

If the optimiser is re-run on the same input image then the output result images will be detected (using the image title) and updated. This feature allows the images to be updated with the results of using the specified parameters when double-clicking on the result table (see section Showing a result).

FindFoci Optimiser GUI

The plugin provides an ImageJ Graphical User Interface (GUI) that runs the FindFoci Optimiser on the selected image.

Features

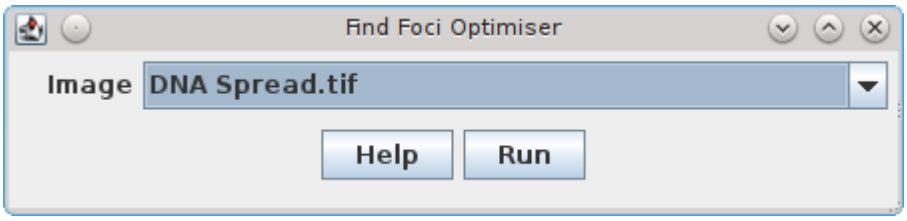
- Permanent GUI within ImageJ
- Allows different images to be selected
- Runs the FindFoci Optimiser algorithm to identify peak regions
- Supports the ImageJ macro recorder

Note that the the plugin supports the ImageJ macro recorder for the FindFoci Optimiser command. This means that a user can record their actions with the FindFoci Optimiser GUI within a macro and it will successfully record the execution of the FindFoci Optimiser plugin.

Plugin Interface

The FindFoci Optimiser GUI uses a Java frame within the ImageJ application. The plugin provides a selection box of the currently open images that can be processed by the optimiser.

The following image shows the interface for the FindFoci Optimiser GUI:



Clicking the Run button launches the FindFoci Optimiser on the selected image. Full details of the parameters can be found in the FindFoci Optimiser Plugin Interface section.

The interface has a drop-down menu that is used to select the image. This allows the user to quickly try the same parameters on different images that are open. The image list is populated with the currently open 8-bit, 16-bit and 32-bit single channel images that have an ImageJ Point ROI. Any images with the title ending in 'FindFoci' (i.e. previous results) are also ignored.

Java Requirements

The FindFoci Optimiser GUI uses the BeansBinding Java framework. More details can be found in Appendix 1: BeansBinding Java framework.

FindFoci Optimiser Multi-Image

ImageJ plugin that iterates through combinations of parameters for the FindFoci plugin. Processes all the reference images in a directory and identifies the best parameters to match the reference results.

Features

- Processes multiple reference images
- Enumerates combinations of parameters for the FindFoci plugin
- Compares the results to a reference image
- Saves the results for each reference image
- Combines the results from multiple images into a single ranking of parameters
- Reloads pre-computed results to allow re-ranking using different scoring metrics
- Supports 2D/3D foci

Input

The plugins requires a set of images in a single directory. Each image must have the foci marked on the image using the ImageJ ROI tool, or recorded in a text file in the same directory. See Stage 1: Create Reference Foci in the FindFoci Optimiser section for details on how to do this.

Method for Combined Ranking

When run on a single image the FindFoci Optimiser computes a score for each parameter set that specifies the match between the reference results and the FindFoci algorithm. The score can be chosen using the parameter Result sort method, for example the Jaccard score.

When using multiple images the overall rank for a set of parameters must combine the score from each image. The scores for each image can be converted to allow fair combination. The converted scores are then averaged across all images for the same set of parameters. The parameters are ranked using their averaged score and the results displayed.

Score Conversion

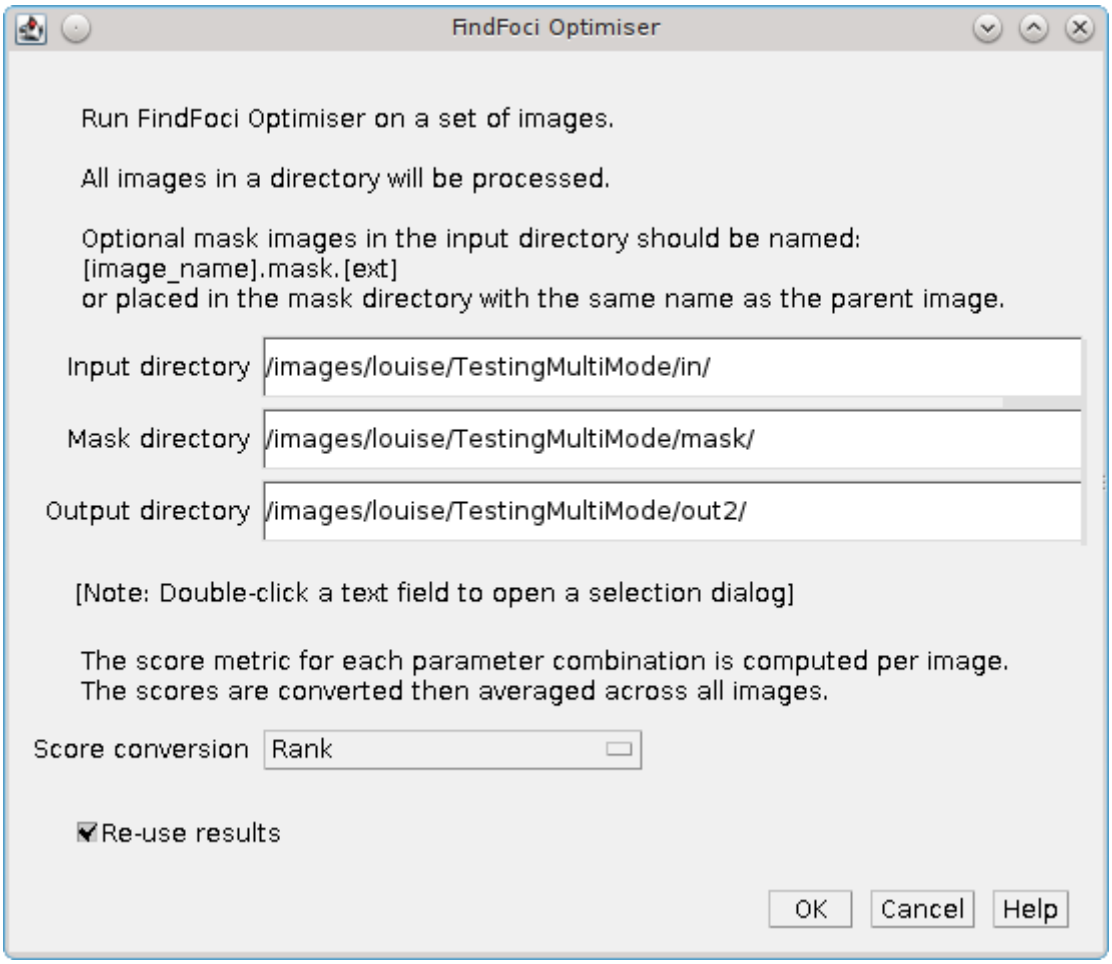
The scores on each image can be converted using the following methods:

Option	Description
Raw score metric	No conversion is made. The raw score, e.g. the Jaccard, will be averaged across all the images.
Relative	The score is converted to a relative score using the percentage drop from the top score: $relative = 100 \times \frac{score - top}{top}$
Z-score	The z-score is computed using the average and standard deviation (SD) of all the scores on the image: $z = \frac{score - average}{SD}$
Rank	The score is used to rank the results on the image and the rank is combined across images

Note that if the RMSD metric is used to score the results then only the Raw and Rank options are valid. If Relative or z-score is specified then Raw will be used. It is not recommended to use RMSD to score results since this will be zero if no true-positives are found, i.e. perfect which is not the case as the result has failed to find any foci. The metric is useful to discriminate results with the same number of true-positives as a secondary comparison.

Plugin Interface

The FindFoci Optimiser Multi-Image plugin is an extension of the FindFoci Optimiser. It uses the same processing engine to enumerate combinations of parameters for the FindFoci algorithm. The plugin shares the same main parameters as the FindFoci Optimiser. Additional parameters for processing multiple images are collected using the following dialog:



The input fields must specify the full path to the directories that exist. The plugin will not create the output directory automatically. To aid in selecting valid directories a double-click in the text field will open a directory selection dialog.

Parameters

Option	Description
Input directory	The directory containing the images to process. Any file with the text 'mask.' will be ignored as these are assumed to be mask images. Optional mask images should be named using the image title with the text 'mask.' placed before the image extension: [image_name].mask.[ext] Optional coordinate files should be named: [image_name].[csv|xyz|txt] If no coordinate file exists for an image then the reference foci will be loaded from point ROI within the image.
Mask directory	The directory containing mask images. This parameter is optional. Mask images must be named with the same name as the parent image in the input directory. Or the images can be named using the image title with the text 'mask.' placed before the image extension: [image_name].mask.[ext]
Output directory	The output directory. Note that the plugin will overwrite files that already exist. The results for each image will be saved to a file named '[image_name].results.xls'. The combined results across all images will be saved to a file named 'all.results.xls' The parameters for the best result across all images will be saved to a file named 'all.params'. This can be input to the FindFoci Batch plugin.
Score conversion	Set the score conversion method (see the Score Conversion section above)
Re-use results	Select this option to re-use results that are in the results directory. Only files with the correct number of parameter combinations will be re-used, otherwise the results will be recomputed. This option can be used to try different 'Score conversion' and 'Result sort method' parameters on the same set of results

The Mask directory parameter is optional. This allows the user to place the mask images in a separate directory or the same directory as the input image. The mask will be found using the following search order:

1. [Mask directory] + [Image filename]

2. [Input directory] + [image_name].mask.[ext]
3. [Mask directory] + [image_name].mask.[ext]

Once the parameters for processing multiple images have been collected the plugin displays the same dialog as the FindFoci Optimiser. However the plugin does not allow specification of a mask image or a results file. The mask file for each image will be detected automatically in the input directory using the filename '[image_name].mask.[ext]'. All results will be saved to the output directory.

Results

When started the plugin will run the FindFoci Optimiser on each input image. Progress can be monitored using the ImageJ progress bar in the ImageJ window. As with the FindFoci Optimiser if the best result for an image is obtained using a numeric parameter that lies on an extreme of the test range then the plugin will log a warning message to the ImageJ log window. This allows the user to increase the range to ensure that the optimal parameters are identified.

Processing is performed using multiple threads to increase speed. The number of threads can be specified using the ImageJ preferences: Edit > Options > Memory & Threads...

The final combined ranking of the top N parameters is displayed in a table in ImageJ. All the combined results are saved to the output directory.

Once the results have been computed it is possible to re-rank the parameter sets using a different scoring method without re-calculating the results. The results will be loaded from the output directory if the 'Re-use results' option is selected and the results files contain the correct number of combinations.

Result Table

The result table shows the top scoring results from all the parameter combinations. The results are sorted using the average score across all the images and limited to the number specified by the Maximum results parameter. Note that RMSD scores cannot be averaged. The combined RMSD is re-computed using the RMSD and number of true positives in each image result.

Results are presented in descending order so that the best result from the most recent analysis is shown at the bottom of the table.

In the event that two sets of parameters have an equal score then the results are ranked using a second metric. By default this is the RMSD (a measure of how close the true positives are to the answer) or if RMSD is chosen as the primary rank metric, the Jaccard.

In the event of a tie the most conservative settings are ranked first. The following logic is used:

- Lowest blur
- Background method: None < Auto-threshold < Std.Dev above mean < Mean < Absolute
- Lowest minimum size
- Search method: Above background < Fraction of peak - background < Half peak value
- Lowest search parameter
- Peak method: Relative above background < Relative height < Absolute height
- Lowest peak parameter
- Lowest run time

The result table shows the result rank and then columns containing the parameters. The table then shows the same scoring metrics as the FindFoci Optimiser. However the counts display the total across all images and the scoring metrics show the average across all images. *It is important to note that the scoring metrics, e.g. Jaccard and F-scores, are not recomputed using the total counts across all images.* The following results are displayed:

Metric	**Description**
N	The total number of maxima in the results
TP (True Positives)	The total number of maxima in the results that match a reference point
FP (False Positives)	The total number of maxima in the results that do not match a reference point
FN (False Negatives)	The total number of reference points that were not identified as maxima
Jaccard	Average Jaccard score across all the images. (Overlap similarity between the reference and the maxima)
Precision	Average Precision score across all the images. (How many identified maxima are correct)
Recall	Average Recall score across all the images. (How many reference points were identified)
F-score	Average F-score across all the images. (Provides a weighted measure of the prediction accuracy where β is a weighting factor between the Precision and Recall.) The F-score is calculated using β of 0.5, 1, 2 and the provided input F-beta parameter.
Score	The average score used for ranking across all the images. This is the score set using Result sort method parameter in the main optimiser dialog and converted per image using the method defined by the Score conversion parameter from the multi-image dialog
RMSD	The Root Mean Square Deviation (RMSD) of the true positive results from the actual. $RMSD = \sqrt{\frac{\sum d^2}{tp}}$ Where d is the distance between the predicted and actual coordinates and tp is the count of true positives. This is computed over all the images.
mSec	The total number of milliseconds across all the image

Note that the metrics Jaccard, Precision, Recall and F-score all range from 0 to 1. Further details of the metrics can be found in Appendix 2: Binary Scoring Statistics.

Invalid Results

Note that the results do not store the distance at which the scoring metrics were computed. Therefore is it left to the user to remember that if the match search distance is changed then results should be recomputed as they will not be valid. The option to Re-use results should be disabled.

However if the user wishes to change the Result sort method or the Score conversion parameters then the results can be re-used to produce the final combined ranking.

FindFoci Helper

An ImageJ plugin that aligns the manually added point Regions of Interest (ROIs) with the peaks found by the FindFoci algorithm.

Features

- Processes 8-, 16- or 32-bit images
- Extracts a single channel and time frame for processing N-dimensional images
- Computes the image peaks using the FindFoci algorithm
- Intercepts user clicks with the multi-point ROI tool and assigns the marked points to the true peaks
- Supports dragging points
- Supports 3D point ROIs (each point is associated with a z slice)
- Outputs assignments to the ImageJ log window
- Optional output of a results table of point positions with intensity

Overview

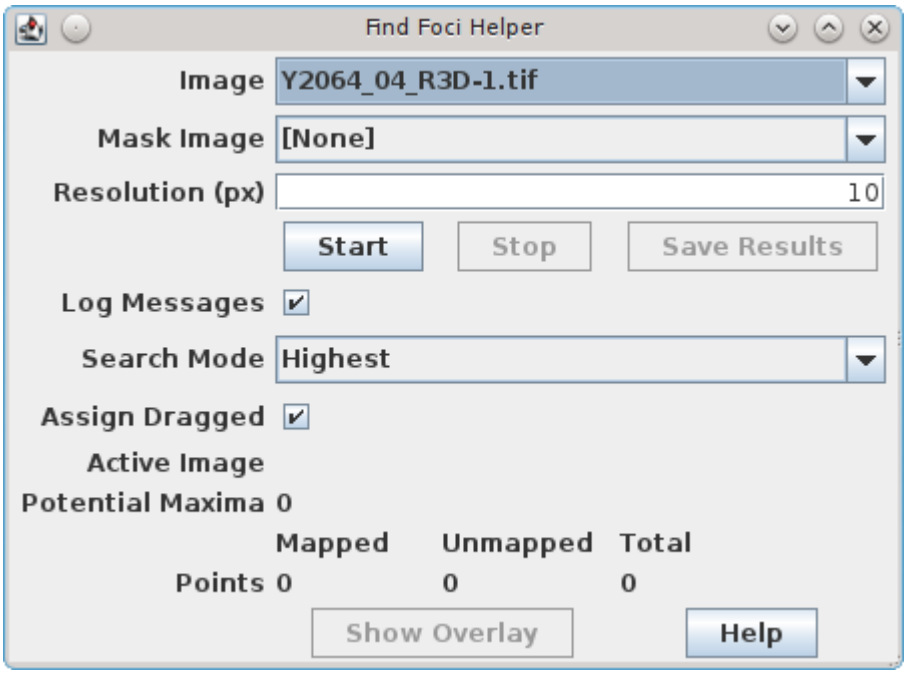
The FindFoci Helper provides a semi-automated tool to assist in manually counting the peaks in an image. The tool is designed to work in conjunction with the ImageJ multi-point ROI tool.

The plugin must be attached to an image and initialised. During initialisation all the potential peaks in an image are calculated using the FindFoci algorithm.

The plugin then listens for mouse-click events from the ImageJ multi-point ROI tool. If a new ROI point is added then the plugin attempts to align the point to an available peak within a set distance. The alignment is performed using the highest or the closest peak. If an existing point is moved then the plugin will detect the drag and attempt to align the point when it is dropped.

Plugin Interface

The FindFoci Helper uses a Java frame within the ImageJ application. The following image shows the plugin window.



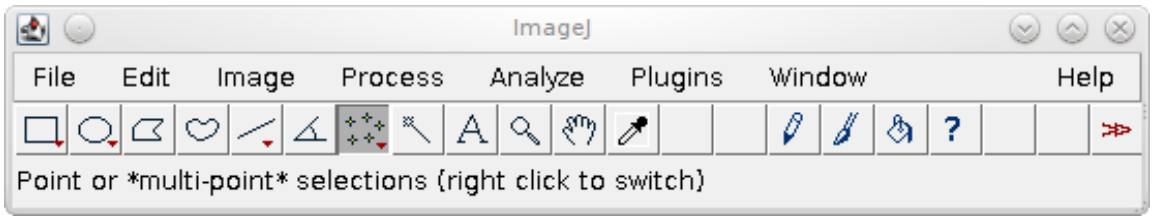
Parameters

Parameters	Description
Image	The target image
Mask Image	Optional mask image defining the region where points will be aligned
Resolution (px)	Defines the search radius for aligning a point (in pixels)
Log Messages	If selected record details of the point alignments to the ImageJ log window
Search Mode	Define the search mode for aligning points to the available maxima: Highest or Closest
Assign Dragged	If selected the plugin will re-align dragged points when they are dropped

Workflow

Multi-point ROI Mode

The ImageJ multi-point ROI tool must be activated. This can be found on the ImageJ toolbar as shown below:



If the single-point ROI mode button is shown then the multi-point mode can be enabled by right-clicking the button and toggling the mode.

The multi-point ROI mode allows the user to manually mark multiple points on an image. Each point is represented by a cross and shows a numbered label (if more than 1 point is selected).

3D Images

The multi-point ROI can associate each point of the ROI with a z slice. This is used by the FindFoci Helper to mark the exact maxima in 3D. ImageJ offers the option to render the ROI with each point only on the z slice that contains the point or on all slices. Double clicking the point ROI button on the toolbar will open the Point Tool options. Use Show on all slices to display the points on every slice in the stack. Any change to the Point Tool options take effect immediately. It is useful to leave the Point Tool options dialog open and toggle between showing on all slices (when picking new points) and showing on the specific slice (when browsing the marked maxima for correctness).

The search resolution is specified in pixels. However a 3D image may have different calibration for the width, height and depth of the voxels. The depth (and height for images with different XY calibration) is scaled relative to the pixel width using the image calibration. For example if the pixel width is 0.1 micrometres and the voxel depth is 0.25 micrometres then the distance of 1 slice is scaled to 2.5px. This allows the search

resolution to apply to 2D and 3D images as the radius of a circle or sphere respectively.

Initialisation

The FindFoci Helper shows a list of the available images that can be processed. This will be all the 8-, 16- and 32-bit images. Select the image and the list of possible masks images will be updated. Any non-zero pixel in the mask defines the region of the image where potential maxima will be detected. The mask image is optional.

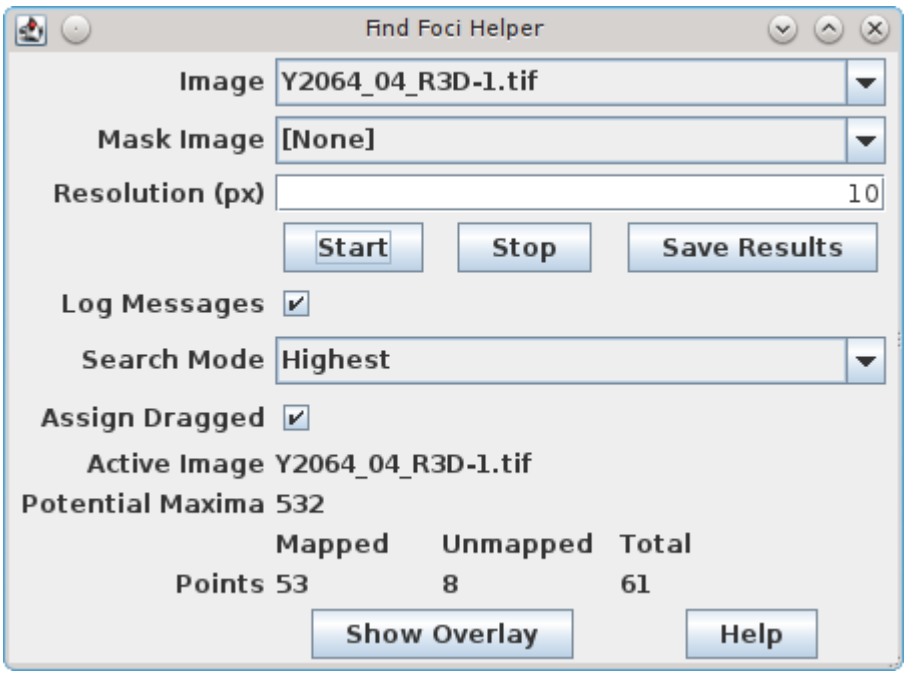
Next configure the resolution for point assignment. The resolution defines the search distance around the point that can be used to locate a maxima. Note that the resolution is dynamic and can be modified after the helper is initialised.

Click Start to initialise the helper. The plugin supports images with multiple channels or time frames but will only calculate the peaks on a single channel and time point. If a multi-dimensional image is selected then the plugin will show a dialogue requesting the user to select the channel and/or frame to be processed.

The plugin will calculate all the potential maxima above the mean of the image using the FindFoci algorithm. The mean is used as a reasonable threshold to avoid identifying false peaks.

If the image contains an existing multi-point ROI then the plugin will attempt to assign each point. If the Highest Search mode is enabled the plugin will process the points in descending height order. If the Closest Search mode is enabled the points will be processed using the closest pairing of point and maxima, iterating until no more assignments can be made.

The number of potential maxima along with the count of mapped and unmapped points will be displayed in the plugin window as shown below:



Marking Peaks

Mark a new peak by clicking on the image. The plugin will search for the closest or highest unassigned maxima within the search resolution.

If a maxima is found the point will be moved to the maxima. The mapped count on the plugin interface will increment.

If no maxima is found then the point will be unchanged. The unmapped count on the plugin

interface will increment.

If the plugin does not assign the peak as expected it can be moved by dragging as described below.

Moving Peaks

If the mouse cursor is hovered over an existing point it will change to a finger. The point can then be moved by holding down the mouse button and moving the mouse to drag the point.

The plugin will identify the point that has been dragged. If it is an aligned point the mapped count on the interface will decrement, otherwise the unmapped count will decrement.

When the mouse button is released the plugin will detect the point has been dropped. If the Assign Dragged option is enabled the plugin will attempt to align the point. Otherwise the point will be unchanged and the unmapped count will increment.

Note that is is possible to disable and re-enable the Assign Dragged option during the course of marking an image. This allows points to be moved without alignment or using alignment.

It is possible that the plugin fails to correctly detect that the peak was assigned before dragging. This will cause the mapped and unmapped counts to show errors. This can be solved by deleting a point which forces all existing points to be remapped without movement. The point can then be added back.

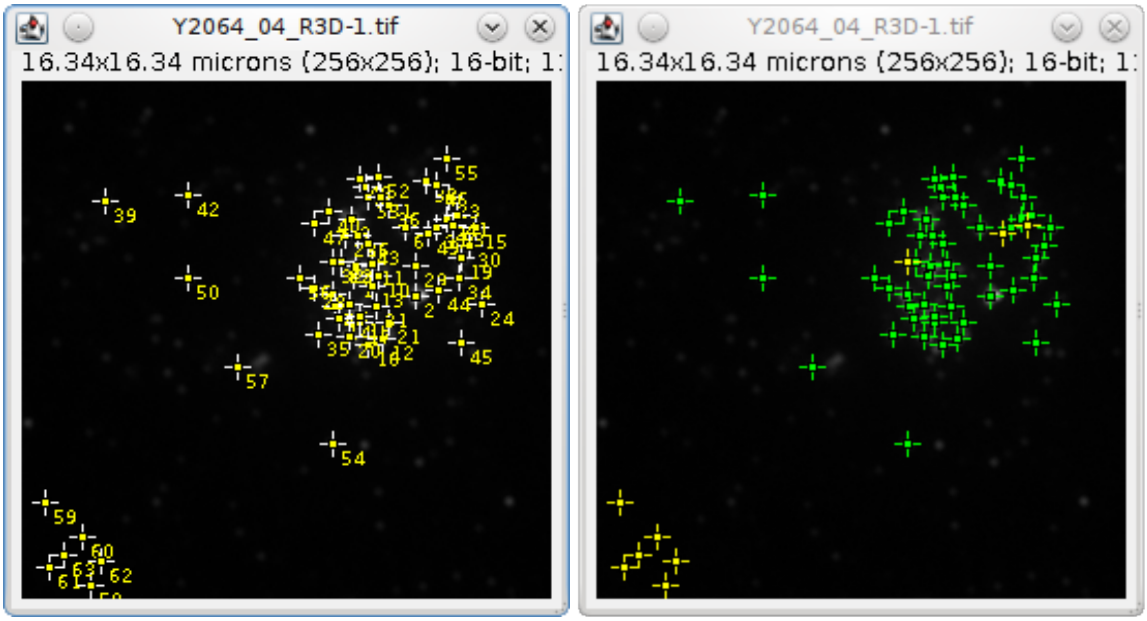
Deleting Peaks

If a the mouse cursor is hovered over an existing point it will change to a finger. It is possible to delete the point by clicking on it whilst holding down the Alt key on the keyboard.

The plugin will identify a point has been deleted. To maintain the ordering of points by ImageJ and the log message window the plugin will process all the remaining points and output the assignments using the new ImageJ point numbers.

Point Overlay

The current mapped and unmapped points can be displayed as an image overlay by clicking the Show Overlay button. This action saves the current ROI points and removes them from the image display. The points are then added back to the image as an overlay using green for the mapped points and yellow for the unmapped points. An example is show below:



The SHOW OVERLAY button is a toggle button. It will return to the off state under the following conditions:

Condition	Result
SHOW OVERLAY clicked	Removes the overlay and restores the ROI points
Image clicked	Removes the overlay and: • If using the multi-point ROI tool then the new point will be added to the ROI points • If a different ROI tool is used, for example the Rectangle tool, then the ROI points will be lost • If any other tool is used, for example to move or zoom the image, then the previous ROI is restored
STOP button clicked	Removes the overlay and restores the ROI points

Saving Results

The current point coordinates can be saved by clicking the SAVE RESULTS button. The table contains the following information:

Result	Description
Id	The point identifier. This matches the IMAGEJ point ROI label
X	The point X coordinate
Y	The point Y coordinate
Height	The image intensity. Since only XY coordinates are used this is for the first plane of a multi-plane Z-stack
Assigned	True if the point is located on a FINDFOCI maxima. False entries indicate points that could not be aligned

Stopping Point Alignment

Clicking the Stop button will halt the FindFoci Helper. It will no longer intercept any mouse interactions with the image. All the working data used by the plugin is discarded and consequently the Save Results option will be disabled.

Note that clicking the Start button will reinitialise the FindFoci Helper and as part of the initialisation will perform alignment of the existing points to maxima. This can potentially move any points that have been dragged with the Assign Dragged option disabled, i.e. points that were previously unmapped.

Alignment Log

If the Log messages option is enabled then the alignment result of each point will be recorded to the ImageJ log.

Each entry is preceded by the point ID. The numbering matches the Ids assigned by ImageJ on the image. Following the ID is the keyword Mapped or Unmapped, depending on the alignment status, and then the XYZ coordinates of the point that was clicked. If a mapped point was moved to align to the true peak then the new coordinates will be shown along with the distance moved in pixels (including scaled pixels for 3D stacks). Examples are shown below:

```
45: Mapped (166,80,1) => (167,80,1) (1.0px)
46: Mapped (201,51,1)
46: Unmapped (196,75,1)
```

Java Requirements

The FindFoci Optimiser GUI uses the BeansBinding Java framework. More details can be found in Appendix 1: BeansBinding Java framework.

FindFoci Settings

The FindFoci Settings plugin sets options for the FindFoci algorithm.

Settings can be adjusted by running the plugin:

The current settings are displayed in the dialog. If the 'OK' button is clicked then the settings are immediately updated for use in all the plugins that use the FindFoci algorithm.

The settings are saved to the ImageJ preferences so that they will be applied for all future ImageJ sessions.

Setting: Capacity

This sets the limit on the number of potential maxima for the FindFoci algorithm. When the algorithm runs each pixel that is higher than all the surrounding pixels is labelled as a potential maxima. If the number of potential maxima exceeds the capacity then the algorithm will not run. This prompts the user to adjust the FindFoci parameters to be more suitable.

The legacy capacity for the algorithm was that set by a signed 16-bit number, 32767. This is the default limit to retain backwards compatibility. This limit may be too low when analysing large images. The algorithm now uses 32-bit integers to support up to 2147483647 candidate maxima.

Setting: Empty Field

When processing multiple files using the FindFoci Batch plugin, each input image has a record written to the batch results file containing all the results (all.xls). This is true even if no foci were found. This allows the results file to be processed for all images, even those with no results.

In the event of no foci, the record written to the results file will contain the batch ID, the image title and then an empty record. All the fields in the empty record will contain the

contents of the EMPTY FIELD setting. By default this is empty but you may require a value within the results file, for example zero (0).

FindFoci Match Calculator

The FindFoci Match Calculator plugin allows two sets of FindFoci results to be compared. The FindFoci analysis must have been saved to memory using the Save to memory option.

Two results sets are required. The plugin will present a dialog allowing the results to be selected. Analysis is performed as per the Match Calculator plugin (see Match Calculator).

FindFoci Macro Extensions

The FindFoci Macro Extensions plugin registers functions that can be used in the ImageJ macro language.

To register the functions in an ImageJ macro script use the following command:

```
run("FindFoci Macro Extensions");
```

The functions can be accessed using the Ext prefix for macro extensions.

getNumberOfPeaks

Get the number of peaks from the last call to FindFoci.

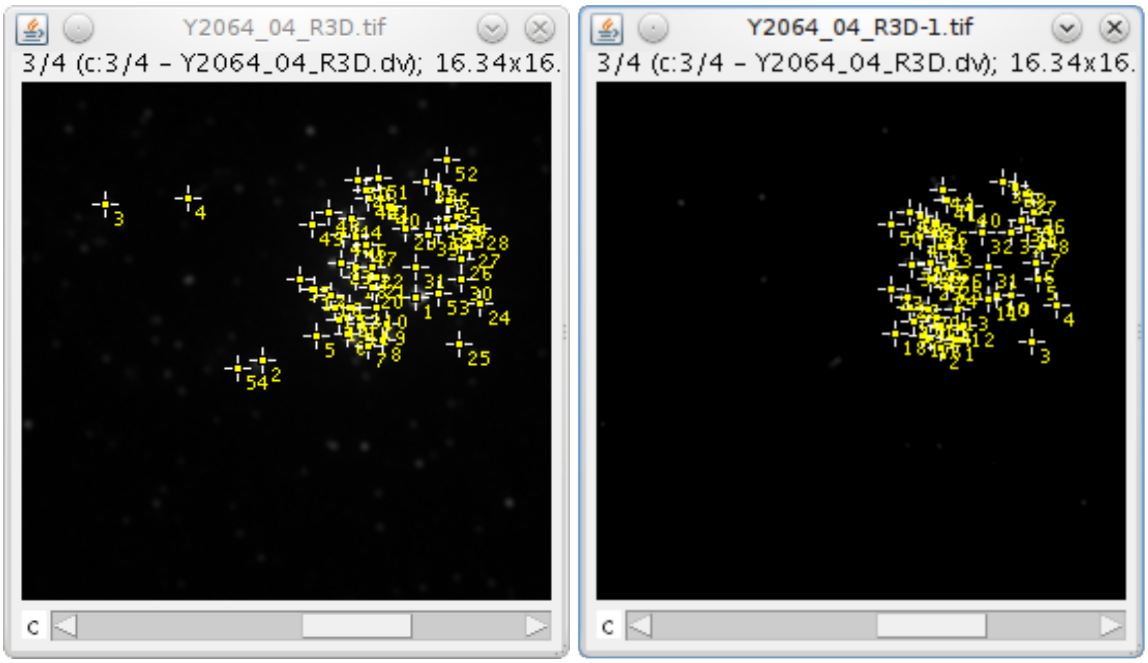
Parameter	Type	Description
numberOfPeaks	Output	The number of peaks.

```
Ext.getNumberOfPeaks(numberOfPeaks);
print(numberOfPeaks);
```

Match Calculator

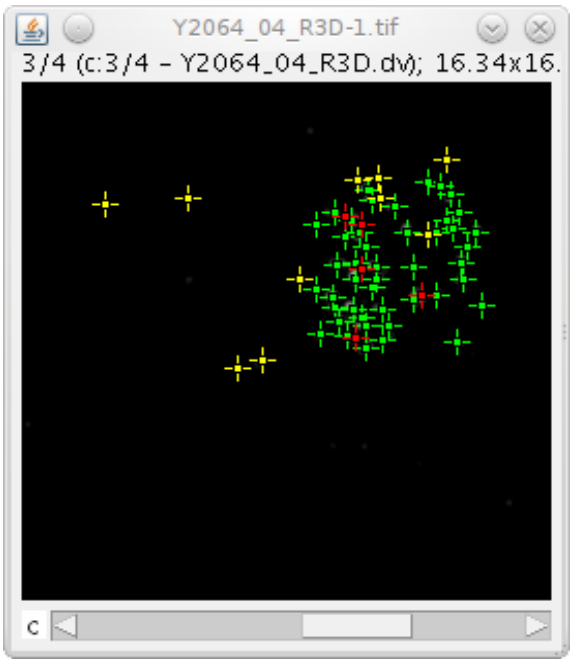
ImageJ plugin that compares the multi-point Regions of Interest (ROIs) between two images.

Example Input

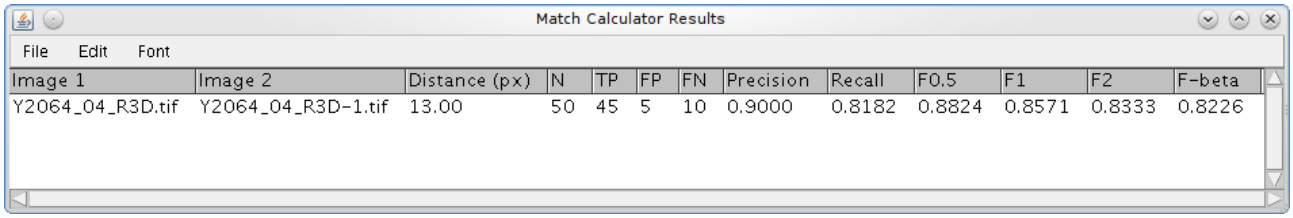


Input images 1 and 2 containing multi-point ROI

Example Output



Output image overlay showing matched points in green, unmatched points from image 1 in yellow and unmatched points from image 2 in red.



Results table showing the match statistics

Features

- Processes any images with point ROI
- Compares the ROI points and assigns matches using a nearest-neighbour method within a configurable distance threshold
- Generates a results table of match statistics
- Optional overlay of the match results onto the image
- Allows selection of channel and frame for multi-dimensional images when extracting heights
- Optional match statistics for each quartile of the data points when ranked by height
- Optional scatter plot of the height of matched and unmatched points
- Optional height analysis of the unmatched points

Overview

The Match Calculator plugin compares the ROI points on two images and computes the number of points that match. The matches can be used to generate match statistics as detailed in Appendix 2: Binary Scoring Statistics.

The plugin was developed to provide comparisons between the manual peak identification performed by different experimenters on the same image. The plugin can rapidly show differences between the peak labelling techniques used by two people.

Match Algorithm

Matches are calculated using a nearest-neighbour algorithm:

1. The coordinates of the ROI points from each image are used to build two sets, A and B
2. An all-vs-all distance matrix is computed for the two sets of points, i.e. the distance for any point in set A to any point in set B
3. The closest pair of points is identified
4. If the closest pair is below a distance threshold they are marked as a pair and removed from the sets. Step 3 is repeated
5. If the closest pair is above the distance threshold the search ends

The distance threshold can be set using an absolute number of pixels or can be set relative to the image size, e.g. 5% of the longest edge. Distances are computed in 2D or 3D if the two sets of points do not have the same z coordinate. 3D distances will use scaling of the z dimension (relative to the XY dimensions) if the input images use the same calibration, otherwise no scaling is performed.

Height Analysis

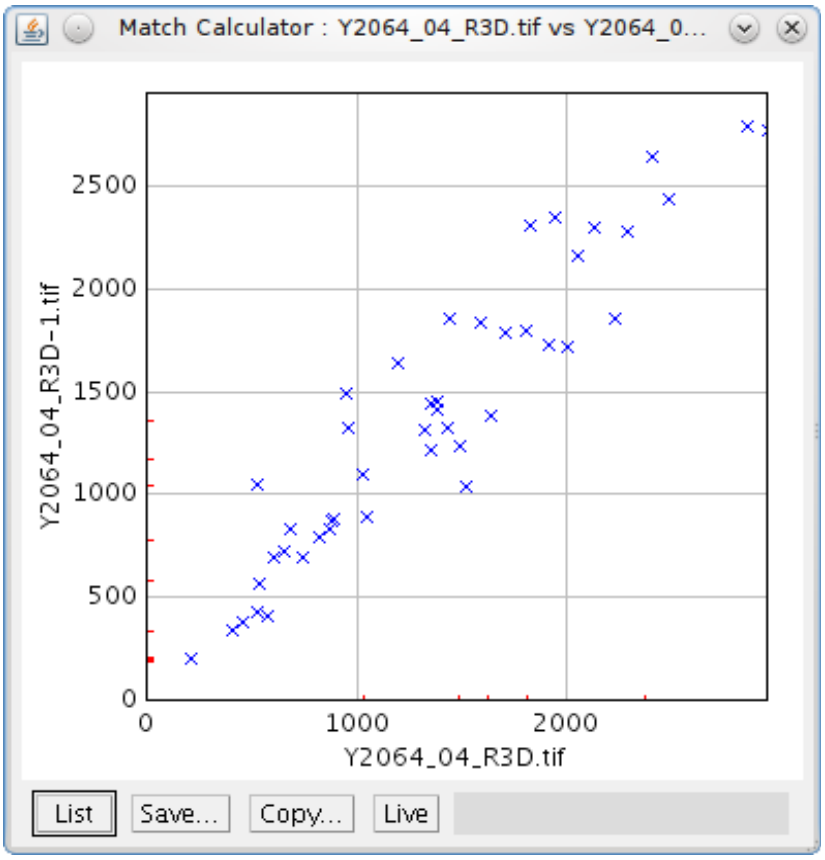
The plugin can produce an analysis of the points using their heights (image intensity values). To eliminate the requirement for an exact z position, or if the ROI do not have a z

coordinate, z-stacks are supported using a maximum intensity projection for height analysis. This requires selection of the channel/frame that is used for the heights for multi-dimensional images.

The heights for all the points can be used to divide the data into quartiles. All the heights from both sets are combined into a single dataset and used to set quartile boundaries. Each set of points is compared to the other using only the points that fall within each quartile boundary. This analysis will show if the higher points (greater intensity) have an increased number of matches. Due to subdivision of the data the analysis is less relevant when the number of points is small (e.g. less than 50).

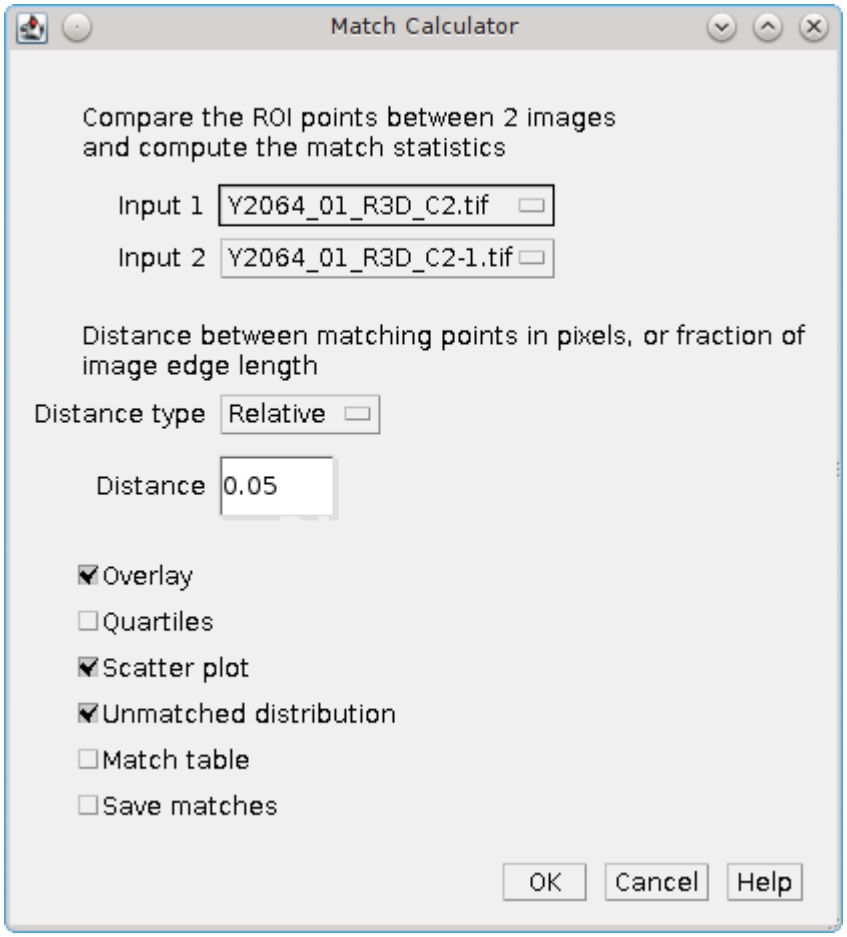
In addition the heights for the paired points can be used to divide the data. In this case the height is taken as the average height of the pair. The height distribution is used to set quartile boundaries. The unmatched points in a set can be counted for each of the height classifications defined using the boundary limits. This analysis shows where the unmatched points occur relative to points that have been matched.

A visual display of the heights for the matched and unmatched points uses a scatter plot of the point heights. The X-axis is the height of the point from set A and the Y-axis uses set B. If the points have been matched they will have both X and Y values. Otherwise they will be plotted on the axis. An example is shown below where matched points are shown in blue and unmatched points in red:



Plugin Interface

The MATCH CALCULATOR plugin uses the standard IMAGEJ Generic Dialog. The input parameters are shown in the following image and are described below.



Parameters

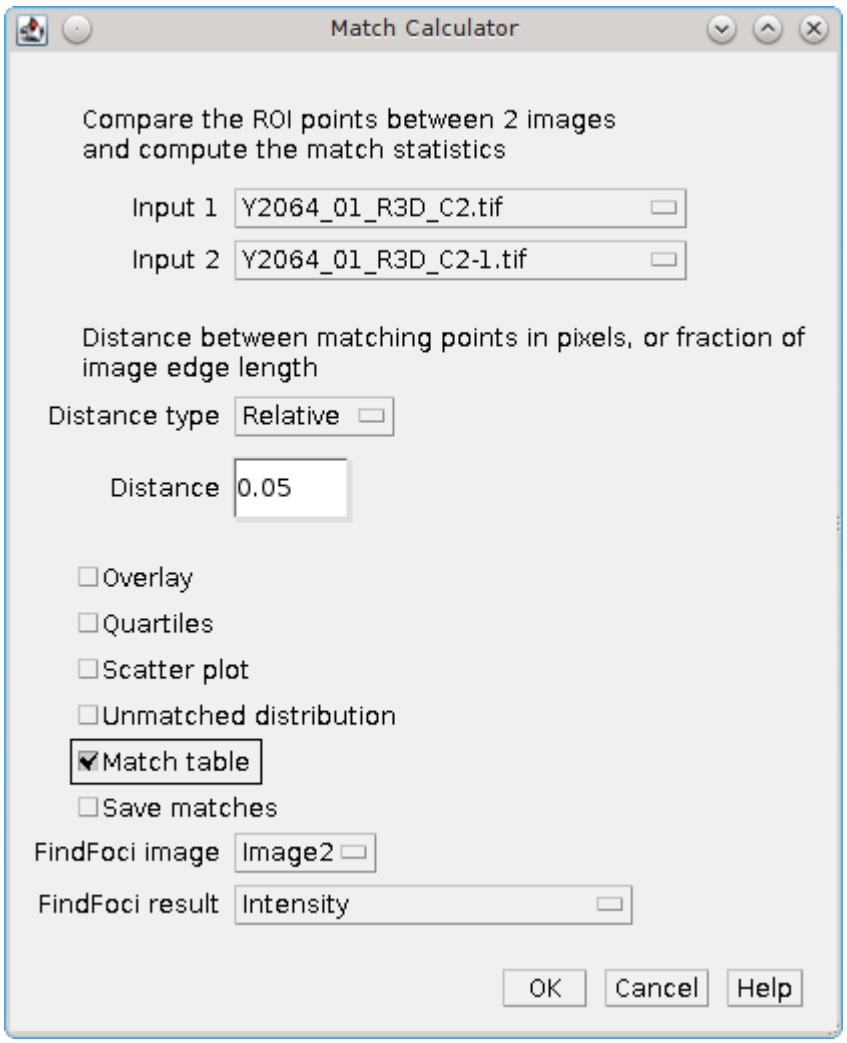
Parameter	Description
INPUT 1 & INPUT 2	The first and second input images. Note: The selection options only show images labelled with ROI points
DISTANCE TYPE	Define the type for the distance parameter: Relative – A fraction of the longest image edge Absolute – A number of pixels
DISTANCE	Specify the maximum distance allowed for a matched pair of points
OVERLAY	Show the matched and unmatched points using a coloured overlay on image 2 (see Example Output)
QUARTILES	Compute the match statistics for the points divided into 4 sets using their heights to define quartiles

Scatter plot	Output a scatter plot of the heights of set A verses set B, matched points a plotted as height pairs
Unmatched distribution	Output a result table of the counts of unmatched points within the limits defined by the quartile boundaries of the matched points
Match table	Show a table with the the coordinates and values of the points, paired into matches and then the unmatched points from one image and then the other
Save matches	Save the matches to a tab delimited file. A file selection dialogue will be shown to allow the filename to be chosen. The file contains the coordinates and values of the points and a summary header containing the match statistics

Extra Parameters

If the FindFoci plugin has been run on an image the results are stored in memory. The Match Calculator can add values from these results to the match table if they correspond to one of the input images. The plugin dialogue presents extra parameters that allow the user to specify which image (if any) is marked with the current FindFoci results. If the selected image has the same number of ROI points as the FindFoci results then the value from the FindFoci results can be added to the match table.

In the example below the second image contains the ROI points identified by the FindFoci plugin. The output match table will contain an extra column showing the peak intensity from the FindFoci results for each point from image 2. Addition of the data from the FindFoci results allows a comparison of a manually marked image to the points identified by the algorithm with data that may explain why points have been matched (e.g. the intensity above background or pixel count is higher in matched points).



Note that if the selected image for the FindFoci image parameter is [None] or the image has the wrong number of marked ROI points then no extra data will be output (since it is impossible to assign the points to the FindFoci results).

Results Table

The results table contains details about the match statistics. The table contains the following information:

Result	Description
Image 1	The first input image
Image 2	The second input image
Distance (px)	The distance limit for matched pairs. Note this is always specified in pixels even when using a relative distance type
N 1	The number of points in the first image
N 2	The number of points in the second image
Match	The number of points that match
Unmatch 1	The number of points in Image 1 that do not match Image 2

Unmatch 2	The number of points in Image 2 that do not match Image 1
Jaccard	The Jaccard score
Recall 1	The proportion of points in Image 1 that appear in Image 2
Recall 2	The proportion of points in Image 2 that appear in Image 1
F1	The F1-score, a balanced weighting of Recall 1 and Recall 2

If the Quartiles parameter is selected then an additional set of columns will be appended for each quartile Q1 through to Q4. There will be a column for N 1, N 2, Matches (M), Unmatched 1 (U1), Unmatched 2 (U2), Jaccard, Recall 1, Recall 2 and F1 for each quartile.

Unmatched Results Table

If the Unmatched distribution parameter is selected then a result table will be generated containing the height distribution of the unmatched points from each image relative to the matched points.

Result	**Description**
Image 1/2	The first/second input image title
N	The number of unmatched points
% <Q1	The percentage of points below the height of the lowest matched point
% Q1	The percentage of points within the first quartile of matched points
% Q2	The percentage of points within the second quartile of matched points
% Q3	The percentage of points within the third quartile of matched points
% Q4	The percentage of points within the forth quartile of matched points
% >Q4	The percentage of points above the height of the highest matched point

Matched Results Table

If the Matched table parameter is selected then a result table will be generated containing the matched and unmatched points.

Result	**Description**
Image 1/2	The first/second input image title
Id	The ROI point ID

X	The X coordinate
Y	The Y coordinate
Distance	The distance between the matched points in pixels
Extra	If the FindFoci results stored in memory have been assigned to one of the images an extra column will be added containing the selected value from the FindFoci results

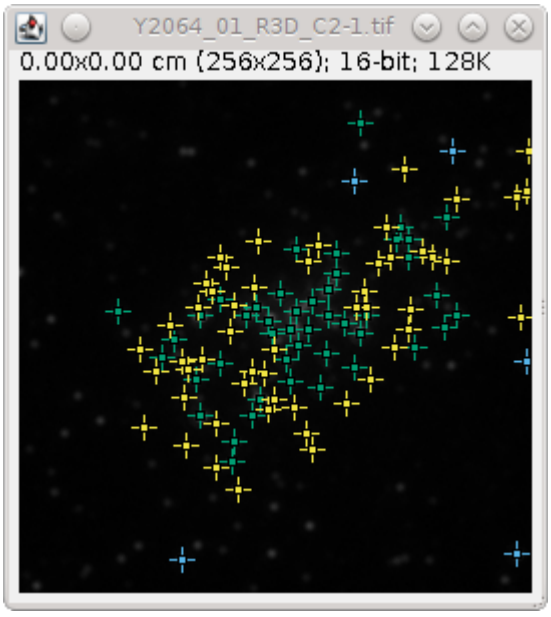
Overlay

The results of the match can be shown as an overlay on the original image 2 using the Overlay parameter. The following key is used for the overlay:

Classification	**Colour**
Matched	Green
Unmatched image 1	Yellow
Unmatched image 2	Blue

Note: The colours have been chosen so that the different points should be visible to those with colour vision deficiency.

The following image shows an example of an overlay applied to the image:



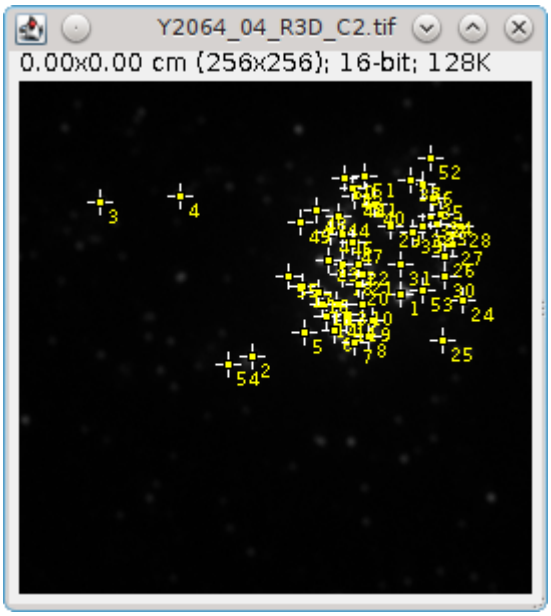
The overlay can be removed using Image > Overlay > Remove Overlay.

Note that the point ROI has been removed from the image. This can be restored using the ImageJ menu option Edit > Selection > Restore Selection (Ctrl+Shift+E). The point ROI are not removed if the overlay option is disabled.

Point Aligner

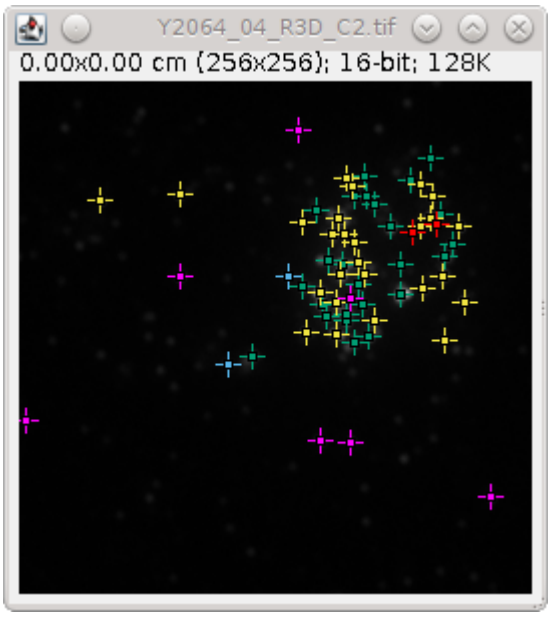
ImageJ plugin that aligns the marked multi-point Regions of Interest (ROIs) of an image with the peaks found by the FindFoci algorithm.

Example Input



Input image containing multi-point ROI

Example Output



Output overlay showing aligned points in green, aligned but moved points in yellow, conflict points in red, unaligned points in blue and missed points in magenta

Point Aligner Results

File Edit Font

Title	Image	OK	Moved	Av.Move	Conflict	NoAlign	Missed	N	TP	FP	FN	Precision	Recall	F1-score
Demo	Y2064_04_R3D_C2.tif	25	26	1.08	2	2	7	55	51	4	7	0.9273	0.8793	0.9027

Results table showing the match statistics

Features

- Processes any single channel image with point ROI
- Computes the image peaks using the FindFoci algorithm
- Assigns the marked points to the true peaks
- Generates a results table of alignment statistics
- Identifies any peaks that have been missed by the marked points
- Optional overlay of the alignment results onto the image
- Optional output of marked points that cannot by aligned (conflict regions) as images

Overview

The Point Aligner plugin aligns the ROI points on an image to the peaks found using the FindFoci algorithm. Comparison of the original points and the aligned points can be used to generate match statistics as detailed in Appendix 2: Binary Scoring Statistics.

The plugin was developed to re-align the manual peak identification performed by an experimenter to the correct peak maxima location.

Alignment Algorithm

Alignments are calculated using the following algorithm:

1. The image ROI points are extracted into an XYZ coordinate set
2. The approximate maxima height of each point is calculated from the XY position using a maximum intensity projection of the image
3. A background level is set using the lowest point height minus the standard deviation of the maximum intensity projection
4. The FindFoci algorithm is used to find maxima above the background level. No merging is performed thus the output peak map contains all potential maxima. Noisy images will benefit from use of a smoothing filter on the input image to eliminate noise and reduce the number of potential maxima.
   The maximum number of maxima may be reached for large images. This can be changed using the FindFoci Settings plugin, or avoided by using a mask to target regions of the input image
5. Points are processed in height order, highest first. Each point is aligned using the peak map (the pixels allocated to each maxima) effectively aligning the point by only proceeding up an intensity gradient to a peak. The maxima is then excluded from future alignments
6. If the point aligns to a previously assigned maxima a search is conducted within the distance to the excluded maxima for an alternative. The highest maxima within the radius is selected
7. Following alignment the heights of all the assigned maxima are analysed. The heights are used to set a limit for the minimum peak height that can be used for an alignment. Any point that aligns to a peak below the minimum peak height is reset to unaligned

Minimum Peak Height Limit

The minimum peak height is used to avoid points aligning to insignificant peaks. Note that the FindFoci algorithm will identify any pixel higher than its neighbours and extend the peak using a downhill gradient. This can produce false peaks at a low intensity that are not above the image background. These are available to the alignment algorithm and so

should be filtered at the end of the process.

The height limit is set using the distribution of the assigned peak heights. The purpose is to identify any outlier low peaks that do not fit the distribution, similar to plotting outlier points on a box plot. The following methods are available, each method requires an input parameter *f*:

Method	**Description**
None	No height limit
Q1 - *f* * IQR	The limit is set using a factor of the Q1-Q2 interquartile range (IQR) subtracted from the Q1-Q2 boundary. A typical factor is 1.5
Mean - *f* * SD	The limit is set using a factor of the standard deviation subtracted from the mean. A factor of 3 will cover 99% of all the data assuming a standard normal distribution
nth Percentile	The limit is set using the *f*-th percentile of the heights. Using zero will not exclude any peaks. 100 will exclude all peaks. Note: This method is poor since it will not target outliers but penalises any low points
% Missed < *f*	The number of unassigned (missed) and assigned peaks above the lowest assigned peak are counted. The percentage missed peaks is calculated: Percentage missed = missed / (missed + assigned) If the percentage missed is above the specified factor then the lowest assigned peak is removed and the process repeated. Finally the limit is set using the lowest assigned peak height. Note: This method assumes that the purpose of the initial ROI point marking was to label the highest intensity values. If there are a large number of missed peaks above the highest assigned peak the method will fail. This can be the case if points were only labelled in a certain region of the image. This can be overcome using a mask image parameter

Alignment Results

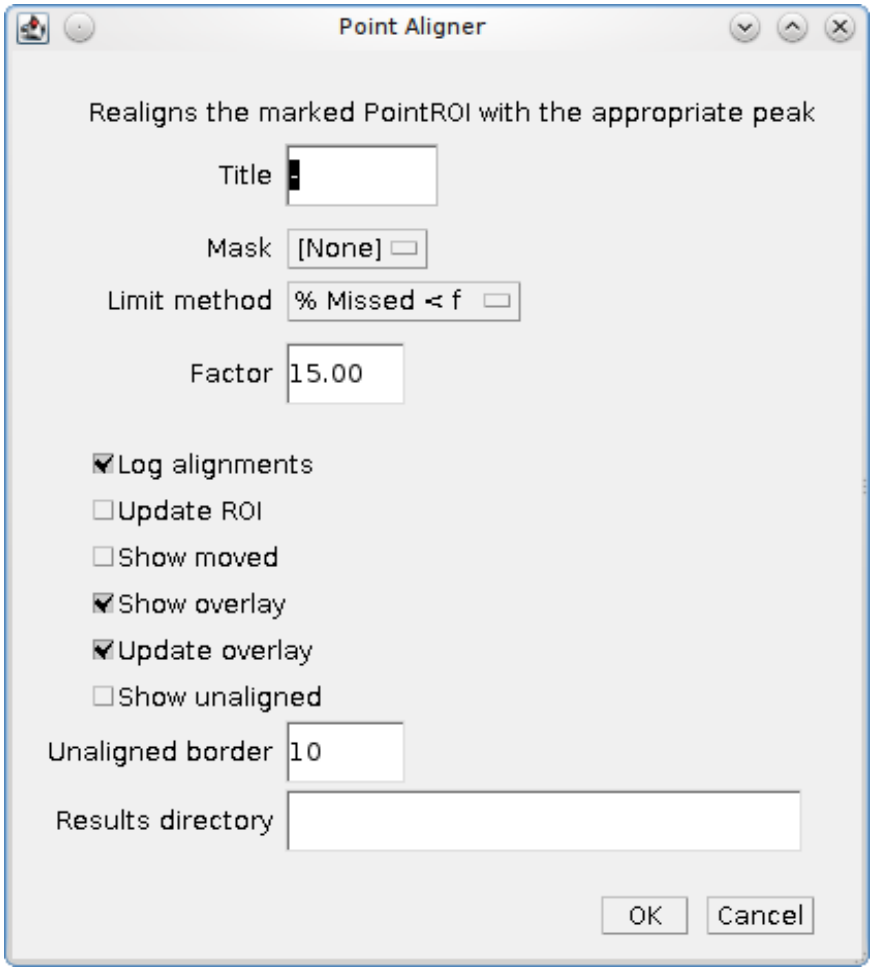
Each point in the original image is given an alignment result. In addition it is possible to analyse maxima found by the FindFoci algorithm. The following alignment results are provided:

Result	**Description**
OK	The point aligns with a maxima and is in the correct location
Moved	The point aligns with a maxima but must be moved
Conflict	The point aligns with a maxima that has been assigned to a higher point, i.e. it conflicts with another aligned point
NoAlign	The point does not align with a maxima

Missed	The unassigned maxima is higher than the lowest maxima that has been assigned, i.e. it was missed in the original input points

Plugin Interface

The Point Aligner interface uses the standard ImageJ Generic Dialog. The input parameters are shown in the following image and are described below.



Parameters

Parameters	**Description**
Title	A text title to be inserted into the results table. For example this can be used within automated scripts to identify the experimenter who scored the image
Mask	Allows selection of an image mask. Only images with compatible dimensions are listed. The mask can be used to restrict the region of peaks provided by the FindFoci algorithm; any peak located in a zero-value pixel will be excluded
Limit Method	The method used to set the minimum height for assigned peaks
Factor	The factor parameter used by the Limit method
Log alignments	Output the alignments to the ImageJ log window

Update ROI	If selected the point ROI on the image will be moved to the position of their assigned peaks. Unassigned points are removed
Show moved	Show the aligned points that have been moved separately in the results table and overlay. The default is to show the OK and Moved points together
Show overlay	Produce a coloured overlay on the image showing the points that were OK, moved, conflicting, missed or unaligned
Update overlay	If selected the overlay will show the moved points relocated to the correct positions. The default is to use the original location
Show unaligned	Extract an image containing each unaligned point. A sub-image is created containing the unaligned point plus a border of extra pixels. This allows examination of any points that cannot be aligned. If the point is unaligned due to a conflict then the previously assigned maxima is included in the image. An overlay is added showing the point location (and the maxima if available)
Unaligned border	The number of border pixels to use for an unaligned point image
Results directory	If provided the unaligned images will be recorded to this directory. The directory path must exist. Note: The Show unaligned option does not have to be enabled

Results Table

The results table contains details about the alignment results. The success of the alignment is assessed using binary scoring statistics. The original ROI points are compared against the potential maxima found by the FindFoci algorithm. Only maxima above the lowest assigned maxima are included. Further details of the statistics can be found in Appendix 2: Binary Scoring Statistics.

The table contains the following information:

Result	**Description**
Title	The Title parameter
Image	The input image
Method	The method used to set the minimum height for assigned peaks
Factor	The factor parameter used by the Limit method
Min Height	The minimum height of the input ROI points
Threshold	The height threshold computed using the heights of the assigned FindFoci results
Min Assigned Height	The minimum height of the assigned FindFoci results

OK	The number of points that were aligned without a move
Moved	The number of points that were aligned with a move. This is always zero if Show moved is not enabled
Av.Move	The average distance that an aligned point is moved in pixels. If Show moved is disabled the average includes the points that are not moved and will be lower
Conflict	The number of points that conflict with another aligned point
NoAlign	The number of points that cannot be aligned
Missed	The number of maxima that are above the lowest assigned maxima
N	The number ROI points in the image
TP	The True Positives, i.e. OK + Moved
FP	The False Positives, i.e. Conflict + NoAlign
FN	The False Negatives, i.e. Missed
Precision	The proportion of points in the image that are aligned
Recall	The proportion of identified peaks in the image that have been assigned
F1	The F1-score, a balanced weighting of Precision and Recall. This provides an overall measure of the alignment success

Alignment Log

If the Log alignments option is enabled then the alignment result of each point will be recorded to the ImageJ log. The following text shows examples of each situation:

OK / Moved

If the peak can be aligned then the new coordinates and the pixel distance for the move are provided. The records are of the format:

```
Point [ID] <height> @ X,Y[,Z] => <height> @ Xnew,Ynew[,Znew] (distance)
```

E.g.

```
Point [49] 793 @ 140,70 => 831 @ 141,70 (1.00)
Point [50] 784 @ 173,57 => 784 @ 173,57 (0.00)
```

Conflict

If the peak is in conflict then the ID of the other peak is provided. The coordinates of both peaks should be in close proximity. E.g.

```
Point [24] 1350 @ 205,67 => 1367 @ 205,68 (1.00)
Point [52] 717 @ 208,71 conflicts for assigned point [24]
```

NoAlign

If the peak is below the height threshold or cannot be aligned this is recorded.

```
Point [55] 479 @ 104,141 ~> 479 @ 104,141 (0.00) below height threshold (< 666)
Point [56] 186 @ 11,208 cannot be aligned
```

Unaligned Images

If a point cannot be aligned then the pixels surrounding the point can be extracted into a sub-image. A point can be unaligned for 3 reasons:

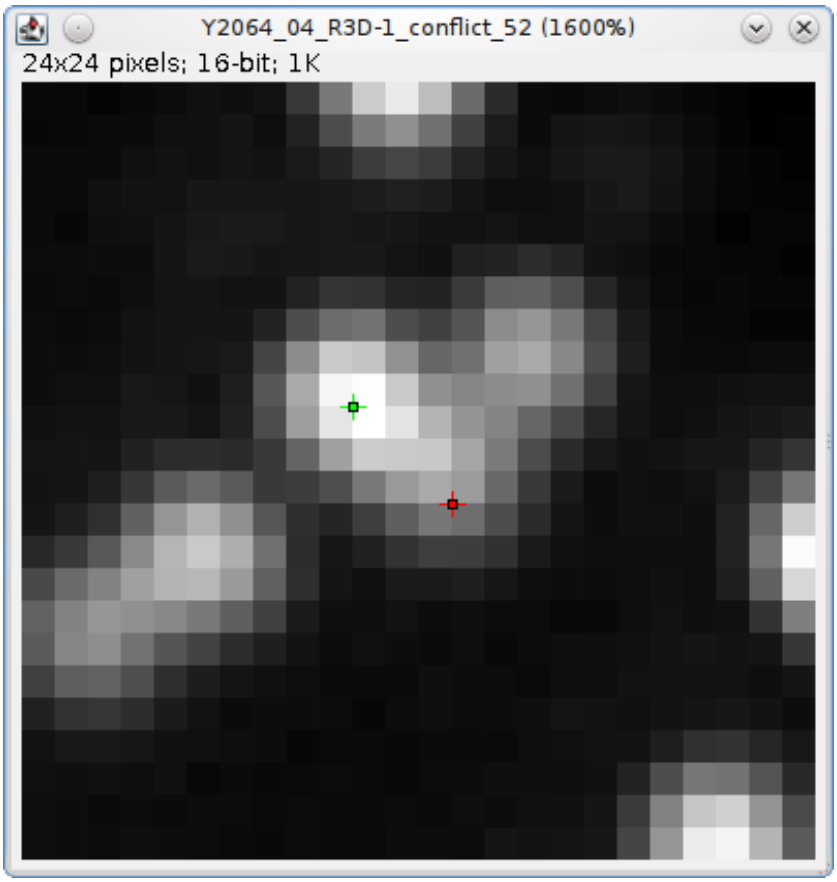
1. There is no suitable maxima [noalign].
   Note: This may be because the point is below the background level for FindFoci
2. It is below the height limit threshold [below_threshold]
3. It conflicts with another aligned point [conflict]

The sub-image is named using the Input image name, the classification and the point ID:

<Input image>_<classification>_<ID>

If the image shows a conflict point or a below_threshold point then the potential maxima is included in the image. An overlay is added to mark the unaligned point and the potential maxima (if available). The images can be displayed using the Show unaligned option and/or saved to the Results directory.

The following is an example of unaligned point sub-image.



Note that in this case the conflict point (red) has been manually placed where there appears to be intensity from a second spot. However the FindFoci algorithm does not identify this as a point since it uses a downhill gradient to define peaks from local maxima. This example demonstrates how the conflict images can be used to review where the

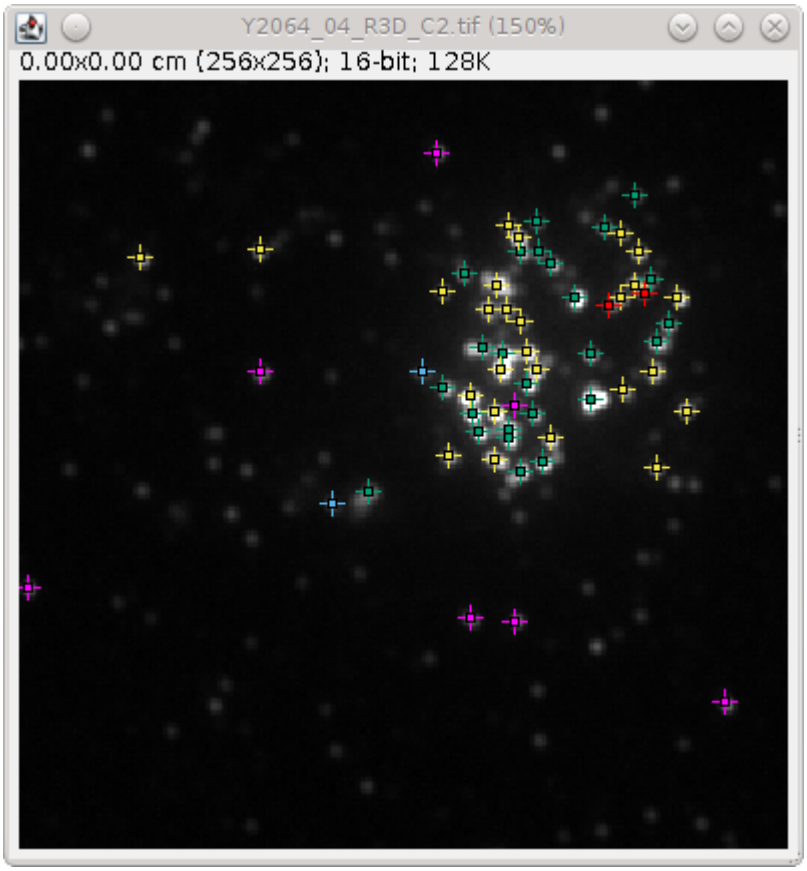
FindFoci algorithm differs from a manual peak assignment. Knowledge of the FindFoci behaviour will allow better use of the semi-automated peak labelling tool provided by the FindFoci Helper plugin (see section FindFoci Helper).

Overlay

The results of the alignment can be shown as an overlay on the original image using the Show overlay parameter. The following key is used for the overlay:

Classification	Colour
OK	Green
Moved	Yellow
Conflict	Red
NoAlign	Cyan
Missed	Magenta

The following image shows an example of an overlay applied to the image:



The overlay can be removed using Image > Overlay > Remove Overlay.

Note that the point ROI has been removed from the image to aid the display of the overlay. This can be done using Edit > Selection > Select None (Ctrl+Shift+A). The point ROI can be restored using the ImageJ menu option Edit > Selection > Restore Selection (Ctrl+Shift+E).

Point Extractor

ImageJ plugin that saves the XYZ coordinates from marked multi-point Regions of Interest (ROIs) in an image or the ImageJ ROI Manager to file.

Features

- Save points from a point ROI on an image
- Save points from the ROI Manager
- Supports 2D/3D coordinates
- Optionally extract a pixel value for the coordinates from the current image
- Optionally filters the points to those within a 2D/3D mask

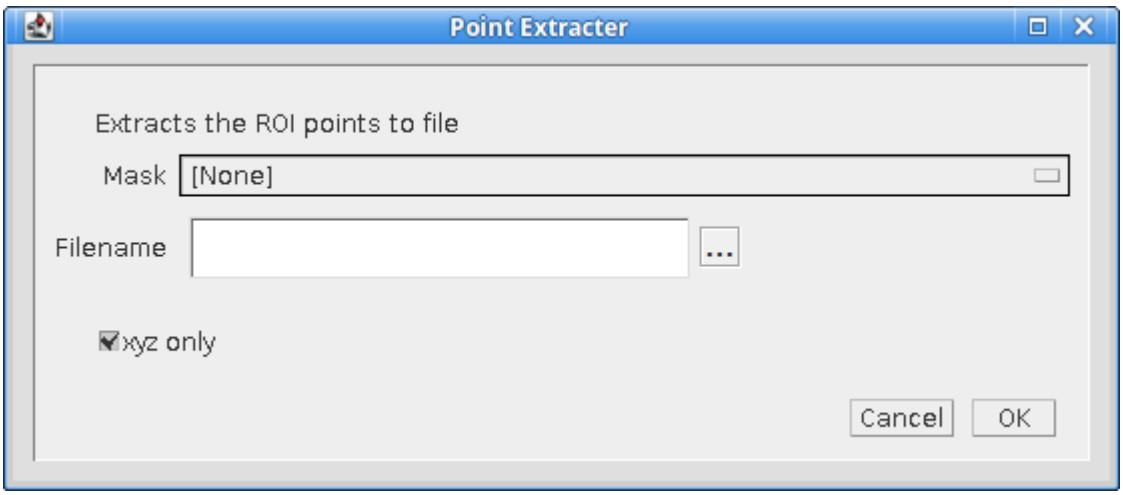
Overview

The Point Extractor plugin saves the coordinates from point ROIs to a coordinate file. The point ROI may be marked on an image. Z coordinates are supported if the points are marked on a specific slice. Alternatively the point ROI may be read from the ImageJ ROI Manager. Each ROI in the manager can be associated with a slice or hyperstack position. The plugin can attempt to extract Z coordinates from all the point ROI in the ROI Manager using the current image to decode the stack position to a z coordinate. The coordinate file can be used as input to train the FindFoci Optimiser.

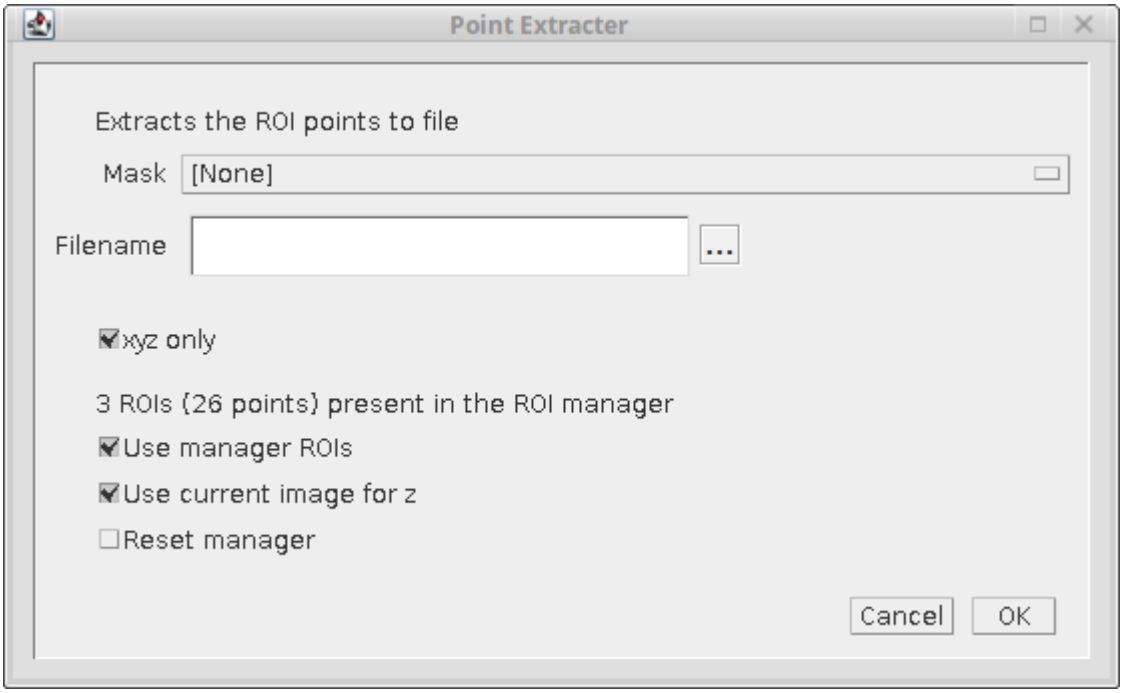
The Point Extractor has options to save the pixel value of each coordinate from the current image to the output file. It can also restrict the coordinates to a specific subset using a mask.

When the plugin is run it counts the number of point ROI in the ImageJ ROI Manager. If no point ROI are present then the current image must have point ROI otherwise an error is shown.

If the image has point ROI then the following dialog is shown:



If the ROI Manager has point ROI then the following dialog is shown:



Parameters

Parameters	Description
Mask	Allows selection of an image mask. The mask can be used to restrict the coordinates. Any point located in a zero-value pixel will be excluded
Filename	The name of the file used to save the coordinates
xyz only	This option is only available if there is a current image. If selected only the XYZ coordinates are saved to file. If not selected then the value of the XYZ coordinate in the current image is obtained and the output file will contain records of: ID,T,X,Y,Z,Value ID will be ascending from 1. T will be the current frame. If the current image has multiple channels or frames then the current channel and frame will be used
Use manager ROIs	This option is available if there are point ROIs in the ImageJ ROI Manager. Select this option to obtain the XYZ coordinates from all the point ROIs in the ROI Manager
Use current image for z	Use the current image to decode stack position information from the ROIs in the ROI Manager
Reset manager	This option is available if there are point ROIs in the ImageJ ROI Manager.

	Select this option to remove all the ROIs from the ROI Manager. This can be used as part of a workflow to label an image, save the ROIs then reset the manager

Creating a 3D coordinate file

ImageJ point ROIs store the XY coordinates and each point can be associated with a stack index. This is performed by adding the points manually in ImageJ using the point ROI tool. This allows the Point Extractor plugin to extract XYZ coordinates directly from an image ROI.

To make it easier to see which points have been selected when scrolling through a z-stack use the point tool options to show the points on all slices. Double clicking the point ROI button on the toolbar will open the Point Tool options. Use Show on all slices to display the points on every slice in the stack. This option will update the ROI immediately allowing the setting to be toggled when marking the ROI.

For images with few foci it is often possible to label all the desired foci in a single ROI. Large images with many foci may require processing parts of the image in stages. Each stage can be saved to the ROI manager. Each ROI saved to the manager can be reapplied to the image to check the points. All points can be extracted together when labelling is complete. Note that the Point Extractor plugin will remove duplicate XYZ positions when creating the output file thus any labelling duplications will be avoided.

The following is a suggested workflow for creating 3D coordinate files via the ROI manager.

1. Remove all ROIs from the current image: Edit > Selection > Select None (Ctrl+Shift+A)
2. Clear the ROI Manager: Analyze > Tools > ROI Manager...
   Select all the ROIs (if present) and click 'Delete'
3. Select the ImageJ point ROI tool. Switch to single-point or multi-point mode by right-clicking the tool and selecting 'Point Tool' or ‘Multi-point Tool’
4. Scroll up and down through the image z-stack until you have decided the correct z-slice for a point. Click the point to label it
5. Add the point to the ROI Manager: Edit > Select > Add To Manager (Ctrl+T).
   When using the multi-point tool the current ROI should be cleared after saving (Ctrl+Shift+A)
6. Continue labelling XY points (step 4-5).
   Note if in single point mode you can create a multi-point ROI by holding the Shift key. This will be replaced by a single-point if you let go of the Shift key for the next click
7. When all foci have been labelled and added to the ROI manager you can review them by clicking on each named ROI in the ROI Manager window. This will show the point ROI on the corresponding image slice
8. If all foci are correct run the Point Extractor plugin to save the foci to file. The ROI Manager can be cleared in preparation for the next image to be labelled using the Reset Manager option

Assign Foci to Objects

The Assign Foci To Objects plugin can align the most recent foci identified by the algorithm to objects on a mask image, for example foci within cells.

Features

- Identify objects within a mask image
- Optionally remove objects based on size
- Identify foci using the latest FindFoci results or a Point ROI
- Assign foci to objects using 2D distance, even if the foci are outside the objects
- Limit assignment using a distance threshold
- Outputs summary statistics of the number of foci per object

Note that analysis is limited to two dimensions (2D).

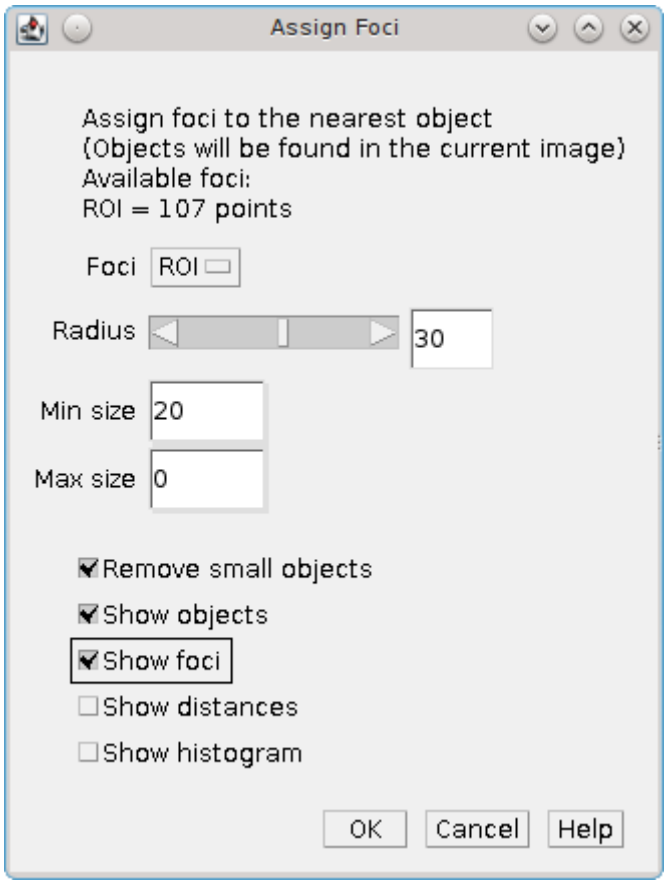
Overview

The Assign Foci To Objects plugin requires a set of foci and an image containing objects.

The foci from the last successful run of the FindFoci plugin are always stored in memory and can be used as the foci. Alternatively a point ROI on the current input image can be used to define the foci, for example if the 'Process > Find Maxima...' command has been run. Note that the point ROI will have to be on the image containing the objects. If the point ROI are on another image then this can be done using 'Edit > Selection > Restore Selection' to copy the ROI from the previously active image (e.g. where the foci have been identified) to the object image.

The object image can be any image. The plugin will find objects defined by contiguous pixels of the same value. So the input image can be a binary mask image or for example an output mask from FindFoci where each foci object has a different pixel value.

If valid foci are available then the following dialog is shown:



Parameters

Parameters	Description
Foci	Select the foci to use. Options are only shown if results are available from that source
Radius	The maximum search radius from each foci to look for objects. Foci will be assigned to the closest object (or not assigned)
Min size	The minimum size for objects. Any objects below this size will be ignored
Max size	The maximum size for objects. Any objects above this size will be ignored. Set below the Min size parameter to ignore
Remove small objects	Enable filtering of objects using the size parameters before foci are assigned. This allows foci to be assigned to a different object if their closest object does not pass the size filter. The alternative is that foci are assigned to their closest objects, even if they do not pass the size filter. The foci are then labelled as ignored in the results
Show objects	Show a mask where each identified object is a different pixel value

Show foci	Show an overlay on the object image where each foci is colour-coded. Green = assigned to an object; Yellow = assigned to an object that failed the size filter; Red = not assigned to an object
Show distances	Output a table showing the distance of each focus to the nearest object
Show histogram	Output a histogram of the count of foci per object

Results

When the plugin is run it identifies all the objects using contiguous pixels of the same value. The objects are then optionally filtered using size criteria. Each foci is then assigned to the closest object, or not assigned if no objects are within the search radius.

If size criteria were used to exclude objects but the objects were not removed then any foci assigned to these invalid objects will be marked as ignored.

The results are reported using tables. A table is shown containing the details of each of the objects with the following information:

Result	**Description**
Image	The title of the object image
Object	The ID of the object
cx	The X centre-of-mass
cy	The Y centre-of-mass
Size	The number of pixels in the object
Valid	True if the object passes the size criteria
Count	The number of foci assigned to the object

A summary table is shown with the following information:

Result	**Description**
Image	The title of the object image
nObjects	The number of object
Valid	The number of valid object
nFoci	The number of foci
In	The number of foci assigned to a valid object
Ignored	The number of foci assigned to a invalid object
Out	The number of foci not assigned to an object

Min	The minimum number of foci in an object
Max	The maximum number of foci in an object
Av	The average number of foci in an object
Med	The median number of foci in an object
SD	The standard deviation of the number of foci in an object

Optionally a table is shown with the distance of each focus to its nearest object:

Result	**Description**
Image	The title of the object image
Foci	The focus ID
x	The focus x position
y	The focus y position
Object	The ID of the nearest object. Will be empty if no objects are closer than the search radius
Valid	True if the object passes the size criteria
Distance	The distance to the object

The plugin also shows graphical results. Optionally a histogram can be shown of the foci count per object. When there are many objects and foci this histogram may present patterns in the number of foci per object.

If the Show objects option has been selected the plugin will shows the objects, optionally with the foci labelled on the image (Show foci option), e.g:

- Green = Assigned to an object
- Yellow = Assigned to an object that fails the size filter
- Red = Not assigned to an object

Assign Foci to Clusters

The Assign Foci To Objects plugin can perform 2D clustering using the most recent foci identified by the algorithm.

Features

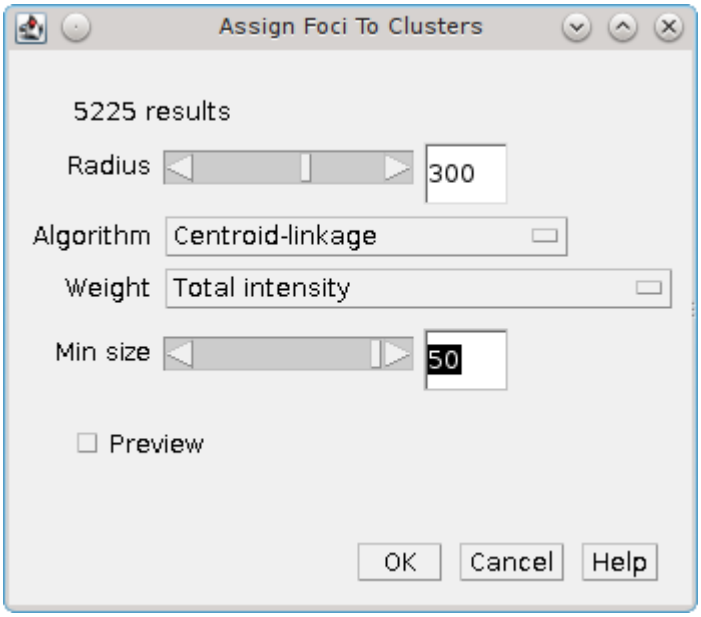
- Perform clustering of foci
- Support display of clusters on a FindFoci mask
- Optionally remove clusters based on size
- Supports results preview
- Outputs a table of clusters

Note that analysis is limited to two dimensions (2D).

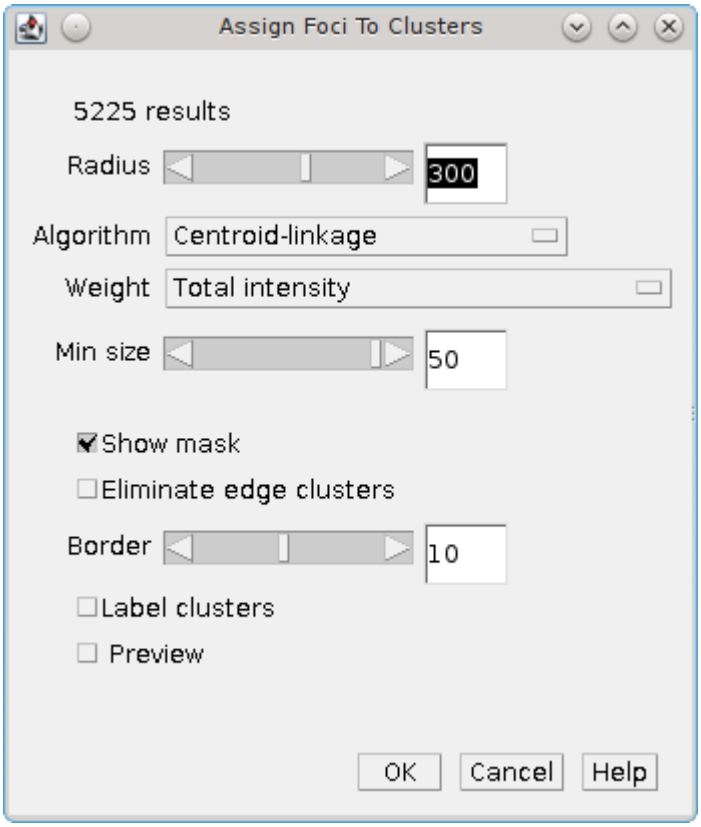
Overview

The Assign Foci To Clusters plugin requires the last set of foci calculated by the FindFoci plugin. These are automatically stored in memory and available for other plugins to use. If no foci are found then the plugin will error.

If valid foci are available then the following dialog is shown:



Optionally the plugin can be run with an input image that is the mask output from FindFoci. The foci loaded from memory are checked against the currently active image when the plugin is run. If the results can be matched to the image pixel values at the focus coordinates then is assumed that the image is a valid input FindFoci mask. The plugin then allows additional options for mask output:



Parameters

Parameters	Description
Radius	The clustering radius
Algorithm	The clustering algorithm (see below). The fastest method is Pairwise which is useful when using the preview
Weight	The weight for each foci. The options are set using various output values from the FindFoci algorithm
Min size	The minimum cluster size
Show mask	Show an output mask coloured by cluster
Eliminate edge clusters	Remove any cluster within the border
Border	Define the border size in pixels
Label clusters	Mark each cluster centre with the cluster ID using an ImageJ point overlay
Preview	Check this to preview results

Clustering Algorithm

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.
Pairwise	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.
Pairwise without neighbours	A variant of Pairwise is to join the closest pairs only if the number of neighbours for each item of the pair is 1, i.e. there are no alternatives. In the event that no pairs have only a single neighbour then only the closest pair is joined

The clustering algorithms iteratively join foci together to create clusters. The centre of the cluster is updated after each join. The centre is computed using a weighted centre-of-mass calculation. Each foci can have a weight assigned to it using output from the FindFoci algorithm, e.g. Pixel count or Total intensity. This allows the algorithm to adapt to the size of the foci it is combining. The weight is set using the Weight parameter.

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. 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.

Results

When the plugin is run it clusters the FindFoci results using the selected clustering

parameters. The clusters are then optionally filtered and the results displayed.

The Preview checkbox can be used to show a live preview of the results. If a valid FindFoci mask is available then the clusters will be shown on the mask image. If not then the number of clusters will be shown on a label on the plugin dialog.

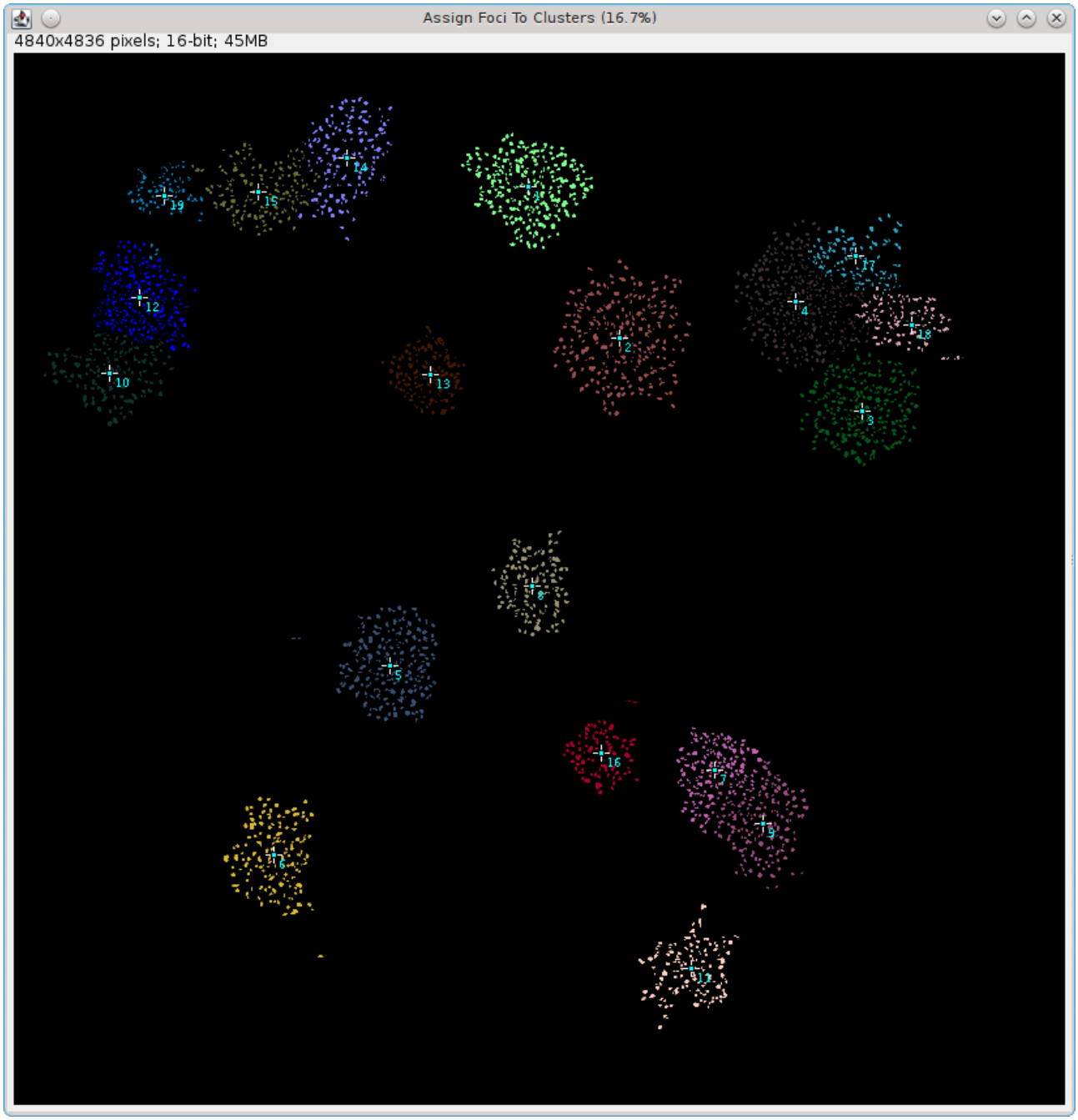
The results are reported using tables. A table is shown containing the details of each of the clusters with the following information:

Result	Description
Title	The title of the FindFoci mask image. If no mask image is available then the title will be 'Result N' with N incrementing for each set of results
Cluster	The ID of the cluster. This will match the pixel value of the cluster on the output mask
cx	The X centre-of-mass
cy	The Y centre-of-mass
Size	The number of foci in the cluster
W	The total sum of the weights of foci in the cluster

A summary table is shown with the following information:

Result	Description
Image	The title of the FindFoci mask image. If no mask image is available then the title will be 'Result N' with N incrementing for each set of results
Radius	The clustering radius
Foci	The total number of foci
Clusters	The total number of clusters
Min	The minimum number of foci in a cluster
Max	The maximum number of foci in a cluster
Av	The average number of foci in a cluster
Min W	The minimum total weight of foci in a cluster
Max W	The maximum total weight of foci in a cluster
Av W	The average total weight of foci in a cluster

An example of the coloured output following clustering is shown below:



References

Herbert AD, Carr AM, Hoffmann E (2014) FindFoci: A Focus Detection Algorithm with Automated Parameter Training That Closely Matches Human Assignments, Reduces Human Inconsistencies and Increases Speed of Analysis. *PLoS ONE* **9**(12): e114749. doi: 10.1371/journal.pone.0114749

Appendix 1: BeansBinding Java framework

The GDSC plugins require the BeansBinding framework. If you have installed the plugins using the ImageJ2 update site then all the required files will be included. However if you have installed the plugins manually then in addition to the GDSC plugins jar file you must install the BeansBinding jar. The file can be obtained from:

> http://www.sussex.ac.uk/gdsc/intranet/files/beansbinding-1.2.1.jar

This file can be placed in the ImageJ plugins directory.

If the BeansBinding jar cannot be found any dependant plugin will show an error message in the ImageJ log window when initialising. The error message should provide useful information for fixing the problem.

The following plugins use the BeansBinding framework:

- FindFoci GUI
- FindFoci Optimiser GUI
- FindFoci Helper

Appendix 2: Binary Scoring Statistics

Several plugins within the GDSC package compute matches between points. When comparing the points the following combinations are possible:

Set 1	Set 2	Classification	Label
Present	Present	Match; True Positive	*tp*
Present	Absent	Unmatch 1; False Negative	*fn*
Absent	Present	Unmatch 2; False Positive	*fp*

The match/unmatch nomenclature is used when it is not clear which set is the real answer. When one set is the actual result and the other is the prediction then it is more conventional to use the tp/fn/fp nomenclature.

The classification counts can be used to compute binary scoring statistics as described below.

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:

$$jaccard = \frac{A \cap B}{A \cup B} = \frac{tp}{tp + fp + fn}$$

A score of 1 indicates that the overlap is perfect. Zero indicates no overlap.

Recall

Recall measures the number of actual points that are correctly predicted. It is also known as the True Positive Rate 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.

Precision

Precision measures the confidence of the predicted points. It is also known as the Positive Predicted Value.

$$precision = \frac{tp}{tp + fp}$$

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.

F-score

The precision and recall can be combined in a weighted score, the $F_ß$-measure or F-score.

$$F_B = (1 + B^2) \cdot \frac{precision \cdot recall}{B^2 \cdot precision + recall}$$

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 ß 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 attaches ß times as much importance to recall as precision.

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. If it is inappropriate to label either set as the actual result then the scores can be labelled as Recall 1 and Recall 2.