Cell Animation

Introduction

CellAnimation is a framework for microscopy assays. To allow for fast assay creation and code reuse each assay is implemented as a modular pipeline. A CellAnimation assay is a chain of MATLAB structures. Each structure describes a module and its connectivity. We have provided a series of assays that we have developed for various microscopy tasks for our lab and others. The intended purpose for each of the assays is described in the “Assays” section.

The CellAnimation core functions are responsible for reading the module chain, creating a dependency tree, populating the input values, executing each module and saving those output values required by modules further downstream (Supp. Fig. 1A). A module is a reusable set of functions that has a narrow specific use (image input-output, thresholding, segmentation, annotation, etc.). Each module we provide is documented in the “Internal Modules” section. We have also included a set of control modules. These are special modules that operate on other modules. Through the use of control modules we can implement looping and branching at the pipeline level. The benefits to this approach are twofold. First, the assay logic becomes easier to follow and modify. Second, it allows us to use smaller, more reusable modules. Individual control modules are documented in the “Control Modules” section.

In addition to our modules, a pipeline may include modules that are just simple wrappers for MATLAB functions or modules that encapsulate functions developed by others. For the MATLAB wrapper modules the documentation may be found in the in the MATLAB help file for that particular function. The caveat is that a particular wrapper may only provide access to some of the functionality of the MATLAB function. In general, MATLAB wrapper modules are named using the name of the MATLAB function concatenated with the string “Wrapper” however, there are some older wrapper modules that only use the name of the MATLAB function. For other external modules the documentation is provided by the original developers.

Assay Editor

Usage

The assay editor is used to create new assays and edit existing assays. To start the assay editor make sure the current directory in MATLAB is set to the directory containing the CellAnimation source files and type *assayEditorGUI* at the MATLAB command prompt.

Main Window

The main window of the assay editor displays all the available CellAnimation modules, the modules in the current assay and a description of the currently selected assay. The available modules are displayed in the top left corner listbox. Clicking on a module in the listbox shows its description in the textbox below. The listbox on the right shows the modules in the current assay. Modules whose names are in a bold font are control modules and have their own submodules. The submodule lists are indicated by an indented italic font immediately below the name of the control module. Double-clicking on a submodule list expands it so the submodules it contains become visible. Double-clicking the submodule list again collapses it. Double-clicking a module name in the *Current Assay* listbox opens the *Edit Module Parameters* dialog box.

Editing an Assay

Open the assay to be edited using *File|Open.* In the *Open Existing Assay* dialog box the listbox on the left shows the available assays while the textbox on the right shows a description of the currently selected assay. To open an assay, double-click its name or select it and then click the *OK* button. Once an assay is opened its main modules become visible in the current assay listbox. There are three different levels of assay editing. Most users will only need to change script parameters of an existing assay to get the assay to work with their data set.

To modify a script variable, click on *Edit|Script Variables*. The *Edit Script Variables* dialog box displays the script variables listed in order of change frequency in the *Name* listbox. Variables whose values are likely to be changed often are displayed at the top of the list while variables that are unlikely to change are displayed towards the bottom of the list. A description of each variable can be found in the assay .m file or in this help file. To change the value of a variable, click on its name, then type a new value in the *Value* textbox on the right.

Advanced users can change values of variables at the individual module level. To change the value of module variables, double-click the desired module level in the *Current Assay* listbox in the main window. This opens the *Edit Module Parameters* dialog box, which displays the input values to the module along with the providers of those values. Each input argument name is displayed in regular font in the *Input Arguments* listbox. Below each argument name is a list of providers for that argument in indented italic font. Input arguments for which no provider has been yet specified are displayed in red. In general each input argument needs at least one provider with the exception of optional arguments. There are two types of providers and they are indicated by the prefix *Value* or *Output*.

Providers starting with the prefix *Value* are constant values that are entered manually by the user. To change a static value, click on a static provider name (usually *Value1*) and type a new value in the *Manual Value* textbox. The value is automatically saved when selecting something else in the *Input Arguments* box or clicking the *OK* button. String values such as file names must be surrounded by single quotation marks. Text entered without quotation marks is interpreted either as a script variable name or a number, if the text consists only of digits. To add a new static value provider, click on the input argument name, enter the value in the *Manual Value* textbox, then click on the *Add Manual Value* button.

Providers starting with the prefix *Output* are dynamic values that are provided at runtime by another module. To modify the provider for a dynamic value, click on a dynamic provider name (*Output1*, *Output2*, etc.) then select another module from the *Module Instance* popup box and/or another output argument from the *Output Argument* popup box. To add a new dynamic provider, click on the input argument name for which you wish to add the provider, select the appropriate module instance and output argument from the popup boxes on the right, then click on *Add Output Argument*.

It is not unusual for an input argument to have both a static and a dynamic provider, with the dynamic provider downstream from the module receiving the value. The static provider is used as an initial value, while subsequent values are obtained from the dynamic provider and change as the assay iterates. The values from the dynamic provider overwrite the static ones as soon as the module providing the value is called.

Creating a New Assay

To create a new assay, click on *File|New*. To add a module to the assay select it in the *Available Modules* listbox, then click on the button displaying an arrow pointing to the right. In the dialog box that appears type an instance name for the current module. The instance name must be a unique name that is not used by any other modules in the assay. You may also select a parent module from the popup box. If the parent module is blank the module will be added as a main module. If the parent module is not blank the module will be added as a submodule belonging to a control module. The choice between a main module and a submodule depends on the intended purpose of the module. A main module will be called exactly once while a submodule may be called multiple times if it's part of a control module that iterates, such as a forLoop module. Once a module instance has been filled in, click the *OK* button. A module that has been added by mistake can be removed by selecting it in the *Current Assay* listbox, then clicking on the button displaying the red X shape.

After the module has been added, providers need to be specified for each of its input arguments. A provider is a value specified either statically when the assay is edited or provided dynamically by one of the modules when the assay is run. To specify providers, double-click on the module, then specify dynamic or static parameters as described in the “Editing an Assay” section.

Modules will be executed in the order they appear in the current assay listbox so you need to make sure that each module appears in its proper place. For example, a *readImage* module should appear above any modules that operate on that image. To change the position of a module in the assay, select it then click either the up arrow button or the down arrow button to move the module up or down respectively. As the module is moved it may become part of a submodule list or exit a submodule list and become a main module instead. This is indicated by the indent level of the module. Main modules are not indented from the left side of the *Current Assay* listbox while submodules are indented one or more spaces from the left side depending on the level of their parent module.

If a parameter is common to several modules or it's likely to change every time an assay is run we recommend you add it as a script variable. To add a script variable, click on *Edit|Script Variables*. In the *Edit Script Variables* dialog box click on the *Add* button. Type the name of the new script variable in the *Name* textbox, then click the *OK* button. This name should be different from any existing script variables belonging to this assay. Type the value of the new script variable in the *Value* textbox, then click on the *OK* button or another script variable name to save it. In the same way as with module variables, a string literal must be enclosed within single quotation marks or it will be interpreted as a script variable name or a number if it consists only of digits. An example of a script variable may be an image file name that is used both by the *readImage* module and the *saveImage* module and is likely to change every time the assay is run. Instead of editing the file name every time the assay is run in both modules, one may add a script variable named *ImageFileName* and set its value to a string literal containing the image file name, for example*'C:/test/image1.jpg'*. Then in the modules, instead of entering the string literal one can simply enter *ImageFileName* and the assay will retrieve the value from the script variable. The value of the script variable can be easily changed when the assay is run on another image as described in “Editing an Assay” and both modules will receive the updated value.

Finally, we recommend adding an assay description so others can easily see the purpose of the assay when they click on it. An assay description can be added by clicking on *Edit*|*Assay Description* and typing the assay description in the *Assay Description* textbox. Once all the modules are in their proper positions and the input parameters have at least one provider, save the assay using *File*|*Save*. A saved assay may be run using *File*|*Run* or by typing the assay name at the MATLAB command prompt.

Exporting Script Variables

Once an assay has been optimized for a particular set of images we recommend exporting the set of optimized script variables to the directory containing the images using *Tools*|*Export Script Variables*. This way if the analysis needs to be repeated at a later date it can be easily done by importing the optimized variables from the image directory using *Tools*|*Import Script Variables*.

Incorporating External MATLAB Functions

At times functionality may be required that is not available as a CellAnimation module but is available in some MATLAB function written by others. The assay editor makes it easy to access such functions via a CellAnimation wrapper module. The function which is to be made available to CellAnimation needs to be in a folder that is part of the MATLAB path. The MATLAB path can be modified using *File*|*Set Path* from the main MATLAB window. To wrap an external function in a CellAnimation module, click on *Tools*|*Wrap MATLAB Function*. Select the MATLAB function to wrap using the dialog box. A wrapper module will be automatically generated and a dialog box presented to the user that allows selection of the directory and file name where the wrapper module is to be saved. We recommend using the automatically generated file name as it makes it easy to recognize the module as a wrapper module. The wrapper module needs to be saved in the directory where the CellAnimation source files reside.

Assays

assayBrightFieldCytoTestNN

Usage

This assay has been optimized for tracking cells imaged using bright-field microscopy using a nearest-neighbor algorithm. The assay tracks the cells, detects mitotic events and records tracking data, cell shape parameters and ancestry information to spreadsheets. For each tracked image, an overlayed image is saved to disk that displays the detected cell outlines, cell IDs and cell generation.

Script Variables

ImageFolder - String variable that specifies the absolute location of the directory which contains the time-lapse images. An example of such a string variable would be *'c:/sample images/high-density'*.

ImageFilesRoot - String variable specifying the root image file name. The root image file name for a set of images is the image file name of any of the images without the number or the file extension. For example, if the file name is *'Experiment-0002_Position(8)_t021.tif'* the root image file name will be *'Experiment-0002_Position(8)_t'.*

ImageExtension – String variable specifying the image file extension including the preceding dot. For example, if the file name is *'image003.jpg'* the image extension is *'.jpg'*.

StartFrame – Number specifying the first image in the sequence to be analyzed. The minimum value for this variable depends on the numbering of the image sequence, so if the first image in the sequence is 'image003.tif' then the minimum value is 3.

FrameCount – Number specifying how many images from the image sequence should be processed.

TimeFrame – Number specifying the time between consecutive images in minutes.

FrameStep - Number specifying the step size when reading images. Set this variable to 1 to read every image in the sequence, 2 to read every other image and so on.

NumberFormat – String value specifying the number of digits in the image file names in the sequence. For example, if the image file name is '*image020.jpg'* the value for the *NumberFormat* is *'%03d'*, while if the file name is '*image000020.jpg'* the value should be *'%06d'*.

MaxFramesMissing – Number specifying for how many frames a cell may disappear before its track is ended.

OutputFolder – The folder where the overlayed images and track data will be saved. By default this value is set to a folder named *'output'* within the folder where the images to be analyzed are located.

AncestryFolder – The folder where the overlayed images and ancestry data will be saved. By default this value is set to a folder named *'ancestry'* within the output folder.

AncestrySpreadsheet – The path name to the spreadsheet containing the ancestry data. By default this value is set to a file named *'ancestry.csv'* within the ancestry folder.

ShapesSpreadsheet – The path name to the spreadsheet containing the position and shape properties for each cell in the time-lapse sequence at every time point. By default this is set to to a file named *'shapes.csv'* within the ancestry folder.

TracksFolder – The folder where the label matrixes containing the cell outlines are saved. By default this value is set to a folder named *'track'* within the output folder.

SegmentationFilesRoot – The root file name of the label matrixes containing the cell outlines.

ImageFileBase – The path name to the images. This value is generated from the *ImageFolder* and the *ImageFilesRoot* and should not be changed.

GradientThreshold – Number specifying the threshold value for the image generated by the *generateBinImgUsingGradient* filter. Any pixels in the gradient image below this value will be set to zero while the rest will be set to one.

ObjectArea – Number specifying the threshold area for the areaFilterLabel, clearSmallObjects, segmentObjectsUsingClusters filters. Objects below this value will be removed from the filtered image.

ObjectReduce – Number specifying how much a cluster of objects should be reduced before it is processed by the *segmentObjectsUsingClusters* module. By default this value is set to 1 and it should not be changed unless *segmentObjectsUsingClusters* throws an out-of-memory error. In that case the value can be reduced to something lower than 1.
ClusterDist – Number specifying the height at which the hierarchical cluster tree will be cut to determine the number of clusters. Setting this value higher results in a cluster being split into fewer objects by the *segmentObjectsUsingClusters* module.
ClearBorder – Boolean value specifying whether objects next to or touching the image border in the binary images generated by the *generateBinImgUsingGradient* module will be erased (*true*) or not (*false*).
ClearBorderDist – Number specifying how close to the border objects may be and still be erased if the *ClearBorder* parameter is set to *true* in the *generateBinImgUsingGradient* module.
ResizeImageScale – Number specifying by what ratio the images will be resized before they are processed. By default this value is set to 0.5. Used by the *resizeImage* module.
ApproximationDistance – Number specifying how close the convex hull in the *getConvexObjects* module approximates the object outline. By default this value is set to 2.5. Setting it to a lower value will result in convex hulls that more closely resemble the object outlines, however this increases the chance of detecting insignificant concavities.
MaxMergeDistance – Number specifying the maximum distance that one track may be from another track for the duration and still be considered for possible merging with the other track. Used by *detectMergeCandidatesUsingDistance* module.
MaxSplitArea – Number specifying the maximum area a nucleus may be and still be considered as a part of a possible mitotic event. Used by *detectMitoticEvents* module.
MaxSplitDistance – Number specifying the maximum distance a new nucleus may be from another nucleus and still be considered as part of a possible mitotic event. Used by *detectMitoticEvents* module.
MinSplitEccentricity – Number specifying the minimum eccentricity a new nucleus may have and still be considered as part of a possible mitotic event. Used by *detectMitoticEvents* module.
MaxSplitEccentricity – Number specifying the maximum eccentricity a new nucleus may have and still be considered as part of a possible mitotic event. Used by *detectMitoticEvents* module.
MinTimeForSplit – Number specifying the minimum time in minutes a track needs to exist before it is considered for a possible mitotic event. Used by *detectMitoticEvents* module.
MinLifespan – Number specifying the minimum length in frames a frame has to be to not be removed by the *removeShortTracks* module.

Important Modules

areaFilterLabel, assignCellToTrackUsingNN, clearSmallObjects, detectMergeCandidatesUsingDistance, detectMitoticEvents, generateBinImgUsingGradient, removeShortTracks, segmentObjectsUsingClusters, segmentObjectsUsingMarkers, splitTracks.

assayBrightFieldCytoTestWG

Usage

This assay has been optimized for tracking cells imaged using bright-field microscopy using a custom algorithm. This algorithm is more accurate and flexible than the nearest-neighbor algorithm at the expense of execution speed. The assay tracks the cells, detects mitotic events and records tracking data, cell shape parameters and ancestry information to spreadsheets. For each tracked image, an overlayed image is saved to disk that displays the detected cell outlines, cell IDs and cell generation.

Script Variables

ImageFolder - String variable that specifies the absolute location of the directory which contains the time-lapse images. An example of such a string variable would be *'c:/sample images/high-density'*.
ImageFilesRoot - String variable specifying the root image file name. The root image file name for a set of images is the image file name of any of the images without the number or the file extension. For example, if the file name is *'Experiment-0002_Position(8)_t021.tif'* the root image file name will be *'Experiment-0002_Position(8)_t'.*
ImageExtension – String variable specifying the image file extension including the preceding dot. For example, if the file name is *'image003.jpg'* the image extension is *'.jpg'*.
StartFrame – Number specifying the first image in the sequence to be analyzed. The minimum value for this variable depends on the numbering of the image sequence, so if the first image in the sequence is 'image003.tif' then the minimum value is 3.
FrameCount – Number specifying how many images from the image sequence should be processed.
TimeFrame – Number specifying the time between consecutive images in minutes.
FrameStep - Number specifying the step size when reading images. Set this variable to 1 to read every image in the sequence, 2 to read every other image and so on.
NumberFormat – String value specifying the number of digits in the image file names in the sequence. For example, if the image file name is '*image020.jpg'* the value for the *NumberFormat* is *'%03d'*, while if the file name is '*image000020.jpg'* the value should be *'%06d'*.
MaxFramesMissing – Number specifying for how many frames a cell may disappear before its track is ended.
OutputFolder – The folder where the overlayed images and track data will be saved. By default this value is set to a folder named *'output'* within the folder where the images to be analyzed are located.
AncestryFolder – The folder where the overlayed images and ancestry data will be saved. By default this value is set to a folder named *'ancestry'* within the output folder.
AncestrySpreadsheet – The path name to the spreadsheet containing the ancestry data. By default this value is set to a file named *'ancestry.csv'* within the ancestry folder.
ShapesSpreadsheet – The path name to the spreadsheet containing the position and shape properties for each cell in the time-lapse sequence at every time point. By default this is set to to a file named *'shapes.csv'* within the ancestry folder.
TracksFolder – The folder where the label matrixes containing the cell outlines are saved. By default this value is set to a folder named *'track'* within the output folder.

SegmentationFilesRoot – The root file name of the label matrixes containing the cell outlines.
ImageFileBase – The path name to the images. This value is generated from the *ImageFolder* and the *ImageFilesRoot* and should not be changed.
GradientThreshold – Number specifying the threshold value for the image generated by the *generateBinImgUsingGradient* filter. Any pixels in the gradient image below this value will be set to zero while the rest will be set to one.
ObjectArea – Number specifying the threshold area for the areaFilterLabel, clearSmallObjects, segmentObjectsUsingClusters filters. Objects below this value will be removed from the filtered image.
ObjectReduce – Number specifying how much a cluster of objects should be reduced before it is processed by the *segmentObjectsUsingClusters* module. By default this value is set to 1 and it should not be changed unless *segmentObjectsUsingClusters* throws an out-of-memory error. In that case the value can be reduced to something lower than 1.
ClusterDist – Number specifying the height at which the hierarchical cluster tree will be cut to determine the number of clusters. Setting this value higher results in a cluster being split into fewer objects by the *segmentObjectsUsingClusters* module.
ClearBorder – Boolean value specifying whether objects next to or touching the image border in the binary images generated by the *generateBinImgUsingGradient* module will be erased (*true*) or not (*false*).
ClearBorderDist – Number specifying how close to the border objects may be and still be erased if the *ClearBorder* parameter is set to *true* in the *generateBinImgUsingGradient* module.
ResizeImageScale – Number specifying by what ratio the images will be resized before they are processed. By default this value is set to 0.5. Used by the *resizeImage* module.
ApproximationDistance – Number specifying how close the convex hull in the *getConvexObjects* module approximates the object outline. By default this value is set to 2.5. Setting it to a lower value will result in convex hulls that more closely resemble the object outlines, however this increases the chance of detecting insignificant concavities.
MaxMergeDistance – Number specifying the maximum distance that one track may be from another track for the duration and still be considered for possible merging with the other track. Used by *detectMergeCandidatesUsingDistance* module.
MaxSearchRadius – Number specifying the absolute lower bound for the search radius to prevent selecting too few candidate objects for a track. Used by *assignCellToTrackUsingAll* module.
MinSearchRadius – Number specifying the absolute higher bound for the search radius to prevent selecting too many candidate objects for a track. Used by *assignCellToTrackUsingAll* module.
MinSecondDistance – Number specifying the minimum significant distance between the closest candidate object to a track and the second closest. Used to determine when distance should be used as a ranking parameter. Used by *assignCellToTrackUsingAll* module.
MaxDistRatio – Number specifying the maximum allowed distance ratio between the two nearest candidate objects. If the ratio is higher than this value, distance ranking will not be used. Used by *assignCellToTrackUsingAll* module.

MaxAngleDiff – Number specifying the maximum allowed angle difference between a track and a candidate object. If the angle is larger than this value, direction ranking will not be used for this object. Used by *assignCellToTrackUsingAll* module.
NrParamsForSureMatch – Number specifying the minimum number of closest matches between a candidate object parameters and a track's object parameters that make the candidate object a sure match to the track. Used by *assignCellToTrackUsingAll* module.
SearchRadiusPct – Number specifying the size of the neighborhood from which candidate objects for matching the track are selected. It is a multiple of the distance to the nearest candidate in the current frame. Setting this variable equal to 1 turns this module into a nearest-neighbor algorithm (only the nearest cell can be a candidate). It does not make sense to have a value lower than 1. Used by *assignCellToTrackUsingAll* module.
MaxSplitArea – Number specifying the maximum area a nucleus may be and still be considered as a part of a possible mitotic event. Used by *detectMitoticEvents* module.
MaxSplitDistance – Number specifying the maximum distance a new nucleus may be from another nucleus and still be considered as part of a possible mitotic event. Used by *detectMitoticEvents* module.
MinSplitEccentricity – Number specifying the minimum eccentricity a new nucleus may have and still be considered as part of a possible mitotic event. Used by *detectMitoticEvents* module.
MaxSplitEccentricity – Number specifying the maximum eccentricity a new nucleus may have and still be considered as part of a possible mitotic event. Used by *detectMitoticEvents* module.
MinTimeForSplit – Number specifying the minimum time in minutes a track needs to exist before it is considered for a possible mitotic event. Used by *detectMitoticEvents* module.
MinLifespan – Number specifying the minimum length in frames a frame has to be to not be removed by the *removeShortTracks* module.
FrontParams – Numeric array specifying a set of column indices from the shape and motility parameters matrix. The parameters in those columns will be heavily weighted, and have more influence in determining the best match for a track from a list of objects. Used by *assignCellToTrackUsingAll* module.
DefaultParamWeights – Numeric array specifying a set of weights that is assigned to each shape and motility parameter based on its prediction power. Parameters with high prediction power are assigned high weights and parameters with low prediction power are assigned lower weights. Used by *assignCellToTrackUsingAll* module.
DistanceRankingOrder – Numeric array specifying the default order of shape and motility parameters for slow-moving objects when it cannot be determined based on prediction power. Used by *assignCellToTrackUsingAll* module.
DirectionRankingOrder – Numeric array specifying the default order of shape and motility parameters for fast-moving directional objects when it cannot be determined based on prediction power. Used by *assignCellToTrackUsingAll* module.
RelevantParametersIndex – Boolean array specifying column indexes in the shape and motility matrix that have been determined to be irrelevant for tracking. This indicates to the module not to use the parameters of those columns in computing track assignment probabilities. The order of column indexes is provided in *TracksLayout* variable. Used by *assignCellToTrackUsingAll* module.

UnknownParamWeights – Numeric array specifying a set of weights to be assigned to shape and motility parameters when the prediction power of the parameters cannot be determined. Used by *assignCellToTrackUsingAll* module.
UnknownRankingOrder – Numeric array specifying the order of the shape and motility parameters when their predictive power cannot be determined. If the objects cannot be categorized as either slow-moving or fast directional the parameter order provided in this variable is used. Used by *assignCellToTrackUsingAll* module.

Important Modules

areaFilterLabel, assignCellToTrackUsingAll, clearSmallObjects, detectMergeCandidatesUsingDistance, detectMitoticEvents, generateBinImgUsingGradient, removeShortTracks, segmentObjectsUsingClusters, segmentObjectsUsingMarkers, splitTracks.

assayCellCoverage

Usage

This assay is used to determine what percentage of an image is occupied by objects.

Script Variables

ImageFileName – String variable specifying the absolute image file name of the image to be analyzed.
ImageDirectory – String variable specifying the directory where the image to be analyzed is located.
MaskFileName – String variable specifying the file name of the resulting binary image from which the object percentage is calculated.

Important Modules

manualSegmentationReview.

assayCellPropsSingleImage

Usage

This assay is used to segment objects in an image subject to manual review, then extract their shape properties.

Script Variables

ImageDirName – String variable specifying the directory where the image to be analyzed is located. An example of such a variable would be *‘c:/test/’*.
FileRoot – String variable specifying the image file name without the extension. For example, if the file name is *'Experiment-0002_Position(8)_t021.tif'* the root file name will be *'Experiment-0002_Position(8)_t021'.*
ImageExt – String variable specifying the extension of the image file name. For example, if the file name is *'Experiment-0002_Position(8)_t021.tif'* the root file name will be *'.tif’.*
ImageFileName – String variable that is automatically generated from the *ImageDirName*, *FileRoot* and *ImageExt.*
OutputDir – String variable specifying the directory where the spreadsheet containing the shape properties and the label matrix containing the cell outlines will be saved.

SpreadsheetFileName – String variable specifying the file name of the spreadsheet containing the shape properties of the objects in the original image. This variable is automatically generated from the *OutputDir* and *FileRoot* variables.
LabelFileName – String variable specifying the file name of the label matrix containing the detected objects in the original image. This variable is automatically generated from the *OutputDir* and *FileRoot* variables.
BrightnessGradientThreshold – Number specifying the threshold value for the image generated by the *generateBinImgUsingLocAvg* filter. Any pixels in the gradient image below this value will be set to zero while the rest will be set to one.
ClearBorder – Boolean value specifying whether objects next to or touching the image border in the binary images generated by the *generateBinImgUsingGradient* module will be erased (*true*) or not (*false*).
ClearBorderDist – Number specifying how close to the border objects may be and still be erased if the *ClearBorder* parameter is set to *true* in the *generateBinImgUsingGradient* module.
MedianFilterSize – Number specifying the size of the median filter used by the *distanceWatershed* module. Setting this to a higher integer value will reduce the number of objects detected by the module and can be used to prevent oversegmentation.
ObjectArea – Number specifying the threshold area for the *clearSmallObjects* filter. Objects below this value will be removed from the filtered image.
Strel – String variable specifying the type of filter used to generate the local average image in *generateBinImgUsingLocAvg*. Currently *‘disk’* is the only value supported.
StrelSize – Number specifying the size of the local neighborhood used to calculate the average for each pixel in the local average image generated by the *generateBinImgUsingLocAvg* module.

Important Modules

clearSmallObjects, distanceWatershed, generateBinImgUsingLocAvg, manualSegmentationReview, segmentObjectsUsingMarkers.

assayFlCytoLNCapManSegWG

Usage

This assay is used to manually review segmentation of objects labeled using a fluorescent dye after they have been segmented using another assay.

Script Variables

ImageFolder - String variable that specifies the absolute location of the directory which contains the time-lapse images. An example of such a string variable would be *‘c:/sample images/high-density’*.
ImageFilesRoot - String variable specifying the root image file name. The root image file name for a set of images is the image file name of any of the images without the number or the file extension. For example, if the file name is *'Experiment-0002_Position(8)_t021.tif'* the root image file name will be *'Experiment-0002_Position(8)_t'.*

ImageExtension – String variable specifying the image file extension including the preceding dot. For example, if the file name is *'image003.jpg'* the image extension is *'.jpg'*.
StartFrame – Number specifying the first image in the sequence to be analyzed. The minimum value for this variable depends on the numbering of the image sequence, so if the first image in the sequence is 'image003.tif' then the minimum value is 3.
FrameCount – Number specifying how many images from the image sequence should be processed.
FrameStep - Number specifying the step size when reading images. Set this variable to 1 to read every image in the sequence, 2 to read every other image and so on.
NumberFormat – String value specifying the number of digits in the image file names in the sequence. For example, if the image file name is '*image020.jpg'* the value for the *NumberFormat* is *'%03d'*, while if the file name is '*image000020.jpg'* the value should be *'%06d'*.
OutputFolder – The folder where the overlayed images and track data will be saved. By default this value is set to a folder named *'output'* within the folder where the images to be analyzed are located.
TracksFolder – The folder where the label matrixes containing the cell outlines are saved. By default this value is set to a folder named *'track'* within the output folder.
SegmentationFilesRoot – The root file name of the label matrixes containing the cell outlines.
ImageFileBase – The path name to the images. This value is generated from the *ImageFolder* and the *ImageFilesRoot* and should not be changed.
ApproximationDistance – Number specifying how close the convex hull in the *getConvexObjects* module approximates the object outline. By default this value is set to 2.5. Setting it to a lower value will result in convex hulls that more closely resemble the object outlines, however this increases the chance of detecting insignificant concavities.
BrightnessThresholdPct – Number specifying the percentage threshold value for the image generated by the *generateBinImgUsingLocAvg* filter. Any pixel in the original image smaller than the threshold value times the corresponding value in the local average image below this value will be set to zero while the rest will be set to one.
ClearBorder – Boolean value specifying whether objects next to or touching the image border in the binary images generated by the *generateBinImgUsingGradient* module will be erased (*true*) or not (*false*).
ClearBorderDist – Number specifying how close to the border objects may be and still be erased if the *ClearBorder* parameter is set to *true* in the *generateBinImgUsingGradient* module.
MedianFilterSize – Number specifying the size of the median filter used by the *distanceWatershed* module. Setting this to a higher integer value will reduce the number of objects detected by the module and can be used to prevent oversegmentation.
ObjectArea – Number specifying the threshold area for the *clearSmallObjects, polygonalAssistedWatershed* filter. Objects below this value will be removed from the filtered image.
Strel – String variable specifying the type of filter used to generate the local average image in *generateBinImgUsingLocAvg*. Currently *'disk'* is the only value supported.
StrelSize – Number specifying the size of the local neighborhood used to calculate the average for each pixel in the local average image generated by the *generateBinImgUsingLocAvg* module.

ResizeImageScale – Number specifying by what ratio the images will be resized before they are processed. By default this value is set to 0.5. Used by the *resizeImage* module.

Important Modules

clearSmallObjects, distanceWatershed, generateBinImgUsingLocAvg, getConvexObjects, manualSegmentationReview, polygonalAssistedWatershed, segmentObjectsUsingMarkers.

assayFlCytoLNCapTracksReviewWG

Usage

This assay is used to manually review the automatic tracks generated using a tracking assay.

Script Variables

ImageFolder - String variable that specifies the absolute location of the directory which contains the time-lapse images. An example of such a string variable would be *'c:/sample images/high-density'*.
ImageFilesRoot - String variable specifying the root image file name. The root image file name for a set of images is the image file name of any of the images without the number or the file extension. For example, if the file name is *'Experiment-0002_Position(8)_t021.tif'* the root image file name will be *'Experiment-0002_Position(8)_t'.*
ImageExtension – String variable specifying the image file extension including the preceding dot. For example, if the file name is *'image003.jpg'* the image extension is *'.jpg'*.
StartFrame – Number specifying the first image in the sequence to be analyzed. The minimum value for this variable depends on the numbering of the image sequence, so if the first image in the sequence is 'image003.tif' then the minimum value is 3.
FrameCount – Number specifying how many images from the image sequence should be processed.
TimeFrame – Number specifying the time between consecutive images in minutes.
FrameStep - Number specifying the step size when reading images. Set this variable to 1 to read every image in the sequence, 2 to read every other image and so on.
NumberFormat – String value specifying the number of digits in the image file names in the sequence. For example, if the image file name is '*image020.jpg'* the value for the *NumberFormat* is *'%03d'*, while if the file name is '*image000020.jpg'* the value should be *'%06d'*.
MaxFramesMissing – Number specifying for how many frames a cell may disappear before its track is ended.
OutputFolder – The folder where the overlayed images and track data will be saved. By default this value is set to a folder named *'output'* within the folder where the images to be analyzed are located.
AncestryFolder – The folder where the overlayed images and ancestry data will be saved. By default this value is set to a folder named *'ancestry'* within the output folder.
AncestrySpreadsheet – The path name to the spreadsheet containing the ancestry data. By default this value is set to a file named *'ancestry.csv'* within the ancestry folder.
ShapesSpreadsheet – The path name to the spreadsheet containing the position and shape properties for each cell in the time-lapse sequence at every time point. By default this is set to a file named *'shapes.csv'* within the ancestry folder.

TracksFolder – The folder where the label matrixes containing the cell outlines are saved. By default this value is set to a folder named *‘track’* within the output folder.
SegmentationFilesRoot – The root file name of the label matrixes containing the cell outlines.
ImageFileBase – The path name to the images. This value is generated from the *ImageFolder* and the *ImageFilesRoot* and should not be changed.

Important Modules

manualTrackingReview.

assayFlCytoLNCapTrackWG

Usage

This assay is used to automatically track cells that have been segmented using another assay.

Script Variables

ImageFolder - String variable that specifies the absolute location of the directory which contains the time-lapse images. An example of such a string variable would be *‘c:/sample images/high-density’*.
ImageFilesRoot - String variable specifying the root image file name. The root image file name for a set of images is the image file name of any of the images without the number or the file extension. For example, if the file name is *'Experiment-0002_Position(8)_t021.tif'* the root image file name will be *'Experiment-0002_Position(8)_t'.*
ImageExtension – String variable specifying the image file extension including the preceding dot. For example, if the file name is *‘image003.jpg’* the image extension is *'.jpg'*.
StartFrame – Number specifying the first image in the sequence to be analyzed. The minimum value for this variable depends on the numbering of the image sequence, so if the first image in the sequence is ‘image003.tif’ then the minimum value is 3.
FrameCount – Number specifying how many images from the image sequence should be processed.
TimeFrame – Number specifying the time between consecutive images in minutes.
FrameStep - Number specifying the step size when reading images. Set this variable to 1 to read every image in the sequence, 2 to read every other image and so on.
NumberFormat – String value specifying the number of digits in the image file names in the sequence. For example, if the image file name is ‘*image020.jpg’* the value for the *NumberFormat* is *'%03d'*, while if the file name is ‘*image000020.jpg’* the value should be *‘%06d’*.
MaxFramesMissing – Number specifying for how many frames a cell may disappear before its track is ended.
OutputFolder – The folder where the overlayed images and track data will be saved. By default this value is set to a folder named *‘output’* within the folder where the images to be analyzed are located.
AncestryFolder – The folder where the overlayed images and ancestry data will be saved. By default this value is set to a folder named *‘ancestry’* within the output folder.
AncestrySpreadsheet – The path name to the spreadsheet containing the ancestry data. By default this value is set to a file named *‘ancestry.csv’* within the ancestry folder.

ShapesSpreadsheet – The path name to the spreadsheet containing the position and shape properties for each cell in the time-lapse sequence at every time point. By default this is set to to a file named *'shapes.csv'* within the ancestry folder.
TracksFolder – The folder where the label matrixes containing the cell outlines are saved. By default this value is set to a folder named *‘track’* within the output folder.
SegmentationFilesRoot – The root file name of the label matrixes containing the cell outlines.
ImageFileBase – The path name to the images. This value is generated from the *ImageFolder* and the *ImageFilesRoot* and should not be changed.
MaxSearchRadius – Number specifying the absolute lower bound for the search radius to prevent selecting too few candidate objects for a track. Used by *assignCellToTrackUsingAll* module.
MinSearchRadius – Number specifying the absolute higher bound for the search radius to prevent selecting too many candidate objects for a track. Used by *assignCellToTrackUsingAll* module.
MinSecondDistance – Number specifying the minimum significant distance between the closest candidate object to a track and the second closest. Used to determine when distance should be used as a ranking parameter. Used by *assignCellToTrackUsingAll* module.
MaxDistRatio – Number specifying the maximum allowed distance ratio between the two nearest candidate objects. If the ratio is higher than this value, distance ranking will not be used. Used by *assignCellToTrackUsingAll* module.
MaxAngleDiff – Number specifying the maximum allowed angle difference between a track and a candidate object. If the angle is larger than this value, direction ranking will not be used for this object. Used by *assignCellToTrackUsingAll* module.
NrParamsForSureMatch – Number specifying the minimum number of closest matches between a candidate object parameters and a track’s object parameters that make the candidate object a sure match to the track. Used by *assignCellToTrackUsingAll* module.
SearchRadiusPct – Number specifying the size of the neighborhood from which candidate objects for matching the track are selected. It is a multiple of the distance to the nearest candidate in the current frame. Setting this variable equal to 1 turns this module into a nearest-neighbor algorithm (only the nearest cell can be a candidate). It does not make sense to have a value lower than 1. Used by *assignCellToTrackUsingAll* module.
MaxSplitArea – Number specifying the maximum area a nucleus may be and still be considered as a part of a possible mitotic event. Used by *detectMitoticEvents* module.
MaxSplitDistance – Number specifying the maximum distance a new nucleus may be from another nucleus and still be considered as part of a possible mitotic event. Used by *detectMitoticEvents* module.
MinSplitEccentricity – Number specifying the minimum eccentricity a new nucleus may have and still be considered as part of a possible mitotic event. Used by *detectMitoticEvents* module.
MaxSplitEccentricity – Number specifying the maximum eccentricity a new nucleus may have and still be considered as part of a possible mitotic event. Used by *detectMitoticEvents* module.
MinTimeForSplit – Number specifying the minimum time in minutes a track needs to exist before it is considered for a possible mitotic event. Used by *detectMitoticEvents* module.
MinLifespan – Number specifying the minimum length in frames a frame has to be to not be removed by the *removeShortTracks* module.

FrontParams – Numeric array specifying a set of column indices from the shape and motility parameters matrix. The parameters in those columns will be heavily weighted, and have more influence in determining the best match for a track from a list of objects. Used by *assignCellToTrackUsingAll* module.
DefaultParamWeights – Numeric array specifying a set of weights that is assigned to each shape and motility parameter based on its prediction power. Parameters with high prediction power are assigned high weights and parameters with low prediction power are assigned lower weights. Used by *assignCellToTrackUsingAll* module.
DistanceRankingOrder – Numeric array specifying the default order of shape and motility parameters for slow-moving objects when it cannot be determined based on prediction power. Used by *assignCellToTrackUsingAll* module.
DirectionRankingOrder – Numeric array specifying the default order of shape and motility parameters for fast-moving directional objects when it cannot be determined based on prediction power. Used by *assignCellToTrackUsingAll* module.
RelevantParametersIndex – Boolean array specifying column indexes in the shape and motility matrix that have been determined to be irrelevant for tracking. This indicates to the module not to use the parameters of those columns in computing track assignment probabilities. The order of column indexes is provided in *TracksLayout* variable. Used by *assignCellToTrackUsingAll* module.
UnknownParamWeights – Numeric array specifying a set of weights to be assigned to shape and motility parameters when the prediction power of the parameters cannot be determined. Used by *assignCellToTrackUsingAll* module.
UnknownRankingOrder – Numeric array specifying the order of the shape and motility parameters when their predictive power cannot be determined. If the objects cannot be categorized as either slow-moving or fast directional the parameter order provided in this variable is used. Used by *assignCellToTrackUsingAll* module.

Important Modules

assignCellToTrackUsingAll, detectMitoticEvents, splitTracks.

assayFluoNuclTestCA

Usage

This module is used to track cells that have been stained with a nuclear stain using a custom CellAnimation algorithm. This algorithm is more accurate and flexible than the nearest-neighbor algorithm at the expense of execution speed. The assay tracks the cells, detects mitotic events and records tracking data, cell shape parameters and ancestry information to spreadsheets. For each tracked image, an overlayed image is saved to disk that displays the detected cell outlines, cell IDs and cell generation.

Script Variables

ImageFolder - String variable that specifies the absolute location of the directory which contains the time-lapse images. An example of such a string variable would be *‘c:/sample images/high-density’*.

ImageFilesRoot - String variable specifying the root image file name. The root image file name for a set of images is the image file name of any of the images without the number or the file extension. For example, if the file name is *'Experiment-0002_Position(8)_t021.tif'* the root image file name will be *'Experiment-0002_Position(8)_t'.*
ImageExtension – String variable specifying the image file extension including the preceding dot. For example, if the file name is *'image003.jpg'* the image extension is *'.jpg'*.
StartFrame – Number specifying the first image in the sequence to be analyzed. The minimum value for this variable depends on the numbering of the image sequence, so if the first image in the sequence is 'image003.tif' then the minimum value is 3.
FrameCount – Number specifying how many images from the image sequence should be processed.
TimeFrame – Number specifying the time between consecutive images in minutes.
FrameStep - Number specifying the step size when reading images. Set this variable to 1 to read every image in the sequence, 2 to read every other image and so on.
NumberFormat – String value specifying the number of digits in the image file names in the sequence. For example, if the image file name is '*image020.jpg'* the value for the *NumberFormat* is *'%03d'*, while if the file name is '*image000020.jpg'* the value should be *'%06d'*.
MaxFramesMissing – Number specifying for how many frames a cell may disappear before its track is ended.
OutputFolder – The folder where the overlayed images and track data will be saved. By default this value is set to a folder named *'output'* within the folder where the images to be analyzed are located.
AncestryFolder – The folder where the overlayed images and ancestry data will be saved. By default this value is set to a folder named *'ancestry'* within the output folder.
AncestrySpreadsheet – The path name to the spreadsheet containing the ancestry data. By default this value is set to a file named *'ancestry.csv'* within the ancestry folder.
ShapesSpreadsheet – The path name to the spreadsheet containing the position and shape properties for each cell in the time-lapse sequence at every time point. By default this is set to a file named *'shapes.csv'* within the ancestry folder.
TracksFolder – The folder where the label matrixes containing the cell outlines are saved. By default this value is set to a folder named *'track'* within the output folder.
SegmentationFilesRoot – The root file name of the label matrixes containing the cell outlines.
ImageFileBase – The path name to the images. This value is generated from the *ImageFolder* and the *ImageFilesRoot* and should not be changed.
BrightnessThresholdPct – Number specifying the percentage threshold value for the image generated by the *generateBinImgUsingLocAvg* filter. Any pixel in the original image smaller than the threshold value times the corresponding value in the local average image below this value will be set to zero while the rest will be set to one.
ObjectArea – Number specifying the threshold area for the *clearSmallObjects, polygonalAssistedWatershed* and *areaFilterLabel* filters. Objects below this value will be removed from the filtered image.
Strel – String variable specifying the type of filter used to generate the local average image in *generateBinImgUsingLocAvg*. Currently *'disk'* is the only value supported.

StrelSize – Number specifying the size of the local neighborhood used to calculate the average for each pixel in the local average image generated by the *generateBinImgUsingLocAvg* module.
ClearBorder – Boolean value specifying whether objects next to or touching the image border in the binary images generated by the *generateBinImgUsingLocAvg* module will be erased (*true*) or not (*false*).
ClearBorderDist – Number specifying how close to the border objects may be and still be erased if the *ClearBorder* parameter is set to *true* in the *generateBinImgUsingLocAvg* module.
MedianFilterSize – Number specifying the size of the median filter used by the *distanceWatershed* module. Setting this to a higher integer value will reduce the number of objects detected by the module and can be used to prevent oversegmentation.
MinSolidity – Number specifying a threshold solidity value for the *solidityFilterLabelObjects* filter. Objects whose solidity is below this value will be removed from the filtered image.
MinAreaOverPerimeter – Number specifying a threshold AOP ratio value for the *areaOverPerimeterFilterLabel* filter. Objects with an AOP ratio smaller than this value will be removed.
ResizeImageScale – Number specifying by what ratio the images will be resized before they are processed. By default this value is set to 0.5. Used by the *resizeImage* module.
ApproximationDistance – Number specifying how close the convex hull in the *getConvexObjects* module approximates the object outline. By default this value is set to 2.5. Setting it to a lower value will result in convex hulls that more closely resemble the object outlines, however this increases the chance of detecting insignificant concavities.
MaxSearchRadius – Number specifying the absolute lower bound for the search radius to prevent selecting too few candidate objects for a track. Used by *assignCellToTrackUsingAll* module.
MinSearchRadius – Number specifying the absolute higher bound for the search radius to prevent selecting too many candidate objects for a track. Used by *assignCellToTrackUsingAll* module.
MinSecondDistance – Number specifying the minimum significant distance between the closest candidate object to a track and the second closest. Used to determine when distance should be used as a ranking parameter. Used by *assignCellToTrackUsingAll* module.
MaxDistRatio – Number specifying the maximum allowed distance ratio between the two nearest candidate objects. If the ratio is higher than this value, distance ranking will not be used. Used by *assignCellToTrackUsingAll* module.
MaxAngleDiff – Number specifying the maximum allowed angle difference between a track and a candidate object. If the angle is larger than this value, direction ranking will not be used for this object. Used by *assignCellToTrackUsingAll* module.
NrParamsForSureMatch – Number specifying the minimum number of closest matches between a candidate object parameters and a track's object parameters that make the candidate object a sure match to the track. Used by *assignCellToTrackUsingAll* module.
SearchRadiusPct – Number specifying the size of the neighborhood from which candidate objects for matching the track are selected. It is a multiple of the distance to the nearest candidate in the current frame. Setting this variable equal to 1 turns this module into a nearest-neighbor algorithm (only the nearest cell can be a candidate). It does not make sense to have a value lower than 1. Used by *assignCellToTrackUsingAll* module.

MaxMergeDistance – Number specifying the maximum distance that one track may be from another track for the duration and still be considered for possible merging with the other track. Used by *detectMergeCandidatesUsingDistance* module.
MaxSplitArea – Number specifying the maximum area a nucleus may be and still be considered as a part of a possible mitotic event. Used by *detectMitoticEvents* module.
MaxSplitDistance – Number specifying the maximum distance a new nucleus may be from another nucleus and still be considered as part of a possible mitotic event. Used by *detectMitoticEvents* module.
MinSplitEccentricity – Number specifying the minimum eccentricity a new nucleus may have and still be considered as part of a possible mitotic event. Used by *detectMitoticEvents* module.
MaxSplitEccentricity – Number specifying the maximum eccentricity a new nucleus may have and still be considered as part of a possible mitotic event. Used by *detectMitoticEvents* module.
MinTimeForSplit – Number specifying the minimum time in minutes a track needs to exist before it is considered for a possible mitotic event. Used by *detectMitoticEvents* module.
MinLifespan – Number specifying the minimum length in frames a frame has to be to not be removed by the *removeShortTracks* module.
FrontParams – Numeric array specifying a set of column indices from the shape and motility parameters matrix. The parameters in those columns will be heavily weighted, and have more influence in determining the best match for a track from a list of objects. Used by *assignCellToTrackUsingAll* module.
DefaultParamWeights – Numeric array specifying a set of weights that is assigned to each shape and motility parameter based on its prediction power. Parameters with high prediction power are assigned high weights and parameters with low prediction power are assigned lower weights. Used by *assignCellToTrackUsingAll* module.
DistanceRankingOrder – Numeric array specifying the default order of shape and motility parameters for slow-moving objects when it cannot be determined based on prediction power. Used by *assignCellToTrackUsingAll* module.
DirectionRankingOrder – Numeric array specifying the default order of shape and motility parameters for fast-moving directional objects when it cannot be determined based on prediction power. Used by *assignCellToTrackUsingAll* module.
RelevantParametersIndex – Boolean array specifying column indexes in the shape and motility matrix that have been determined to be irrelevant for tracking. This indicates to the module not to use the parameters of those columns in computing track assignment probabilities. The order of column indexes is provided in *TracksLayout* variable. Used by *assignCellToTrackUsingAll* module.
UnknownParamWeights – Numeric array specifying a set of weights to be assigned to shape and motility parameters when the prediction power of the parameters cannot be determined. Used by *assignCellToTrackUsingAll* module.
UnknownRankingOrder – Numeric array specifying the order of the shape and motility parameters when their predictive power cannot be determined. If the objects cannot be categorized as either slow-moving or fast directional the parameter order provided in this variable is used. Used by *assignCellToTrackUsingAll* module.

Important Modules
areaFilterLabel, areaOverPerimeterFilterLabel, assignCellToTrackUsingAll, clearSmallObjects, detectMergeCandidatesUsingDistance, detectMitoticEvents, distanceWatershed, generateBinImgUsingLocAvg, getConvexObjects, polygonalAssistedWatershed, removeShortTracks, segmentObjectsUsingMarkers, solidityFilterLabel, splitTracks.

assayFluoNuclTestNN

Usage
This module is used to track cells that have been stained with a nuclear stain using a nearest-neighbor algorithm. The assay tracks the cells, detects mitotic events and records tracking data, cell shape parameters and ancestry information to spreadsheets. For each tracked image, an overlayed image is saved to disk that displays the detected cell outlines, cell IDs and cell generation.

Script Variables
ImageFolder - String variable that specifies the absolute location of the directory which contains the time-lapse images. An example of such a string variable would be *‘c:/sample images/high-density’*.
ImageFilesRoot - String variable specifying the root image file name. The root image file name for a set of images is the image file name of any of the images without the number or the file extension. For example, if the file name is *'Experiment-0002_Position(8)_t021.tif'* the root image file name will be *'Experiment-0002_Position(8)_t'.*
ImageExtension – String variable specifying the image file extension including the preceding dot. For example, if the file name is *‘image003.jpg’* the image extension is *'.jpg'*.
StartFrame – Number specifying the first image in the sequence to be analyzed. The minimum value for this variable depends on the numbering of the image sequence, so if the first image in the sequence is ‘image003.tif’ then the minimum value is 3.
FrameCount – Number specifying how many images from the image sequence should be processed.
TimeFrame – Number specifying the time between consecutive images in minutes.
FrameStep - Number specifying the step size when reading images. Set this variable to 1 to read every image in the sequence, 2 to read every other image and so on.
NumberFormat – String value specifying the number of digits in the image file names in the sequence. For example, if the image file name is ‘*image020.jpg’* the value for the *NumberFormat* is *'%03d'*, while if the file name is ‘*image000020.jpg’* the value should be *‘%06d’*.
MaxFramesMissing – Number specifying for how many frames a cell may disappear before its track is ended.
OutputFolder – The folder where the overlayed images and track data will be saved. By default this value is set to a folder named *‘output’* within the folder where the images to be analyzed are located.
AncestryFolder – The folder where the overlayed images and ancestry data will be saved. By default this value is set to a folder named *‘ancestry’* within the output folder.
AncestrySpreadsheet – The path name to the spreadsheet containing the ancestry data. By default this value is set to a file named *‘ancestry.csv’* within the ancestry folder.

ShapesSpreadsheet – The path name to the spreadsheet containing the position and shape properties for each cell in the time-lapse sequence at every time point. By default this is set to to a file named *'shapes.csv'* within the ancestry folder.
TracksFolder – The folder where the label matrixes containing the cell outlines are saved. By default this value is set to a folder named *'track'* within the output folder.
SegmentationFilesRoot – The root file name of the label matrixes containing the cell outlines.
ImageFileBase – The path name to the images. This value is generated from the *ImageFolder* and the *ImageFilesRoot* and should not be changed.
BrightnessThresholdPct – Number specifying the percentage threshold value for the image generated by the *generateBinImgUsingLocAvg* filter. Any pixel in the original image smaller than the threshold value times the corresponding value in the local average image below this value will be set to zero while the rest will be set to one.
ObjectArea – Number specifying the threshold area for the areaFilterLabel, clearSmallObjects, segmentObjectsUsingClusters filters. Objects below this value will be removed from the filtered image.
Strel – String variable specifying the type of filter used to generate the local average image in *generateBinImgUsingLocAvg*. Currently *'disk'* is the only value supported.
StrelSize – Number specifying the size of the local neighborhood used to calculate the average for each pixel in the local average image generated by the *generateBinImgUsingLocAvg* module.
ClearBorder – Boolean value specifying whether objects next to or touching the image border in the binary images generated by the *generateBinImgUsingGradient* module will be erased (*true*) or not (*false*).
ClearBorderDist – Number specifying how close to the border objects may be and still be erased if the *ClearBorder* parameter is set to *true* in the *generateBinImgUsingGradient* module.
MedianFilterSize – Number specifying the size of the median filter used by the *distanceWatershed* module. Setting this to a higher integer value will reduce the number of objects detected by the module and can be used to prevent oversegmentation.
MinSolidity – Number specifying a threshold solidity value for the *solidityFilterLabelObjects* filter. Objects whose solidity is below this value will be removed from the filtered image.
MinAreaOverPerimeter – Number specifying a threshold AOP ratio value for the *areaOverPerimeterFilterLabel* filter. Objects with an AOP ratio smaller than this value will be removed.
ResizeImageScale – Number specifying by what ratio the images will be resized before they are processed. By default this value is set to 0.5. Used by the *resizeImage* module.
ApproximationDistance – Number specifying how close the convex hull in the *getConvexObjects* module approximates the object outline. By default this value is set to 2.5. Setting it to a lower value will result in convex hulls that more closely resemble the object outlines, however this increases the chance of detecting insignificant concavities.
MaxMergeDistance – Number specifying the maximum distance that one track may be from another track for the duration and still be considered for possible merging with the other track. Used by *detectMergeCandidatesUsingDistance* module.
MaxSplitArea – Number specifying the maximum area a nucleus may be and still be considered as a part of a possible mitotic event. Used by *detectMitoticEvents* module.

MaxSplitDistance – Number specifying the maximum distance a new nucleus may be from another nucleus and still be considered as part of a possible mitotic event. Used by *detectMitoticEvents* module.
MinSplitEccentricity – Number specifying the minimum eccentricity a new nucleus may have and still be considered as part of a possible mitotic event. Used by *detectMitoticEvents* module.
MaxSplitEccentricity – Number specifying the maximum eccentricity a new nucleus may have and still be considered as part of a possible mitotic event. Used by *detectMitoticEvents* module.
MinTimeForSplit – Number specifying the minimum time in minutes a track needs to exist before it is considered for a possible mitotic event. Used by *detectMitoticEvents* module.
MinLifespan – Number specifying the minimum length in frames a frame has to be to not be removed by the *removeShortTracks* module.

Important Modules

areaFilterLabel, areaOverPerimeterFilterLabel, assignCellToTrackUsingNN, clearSmallObjects, detectMergeCandidatesUsingDistance, detectMitoticEvents, distanceWatershed, generateBinImgUsingLocAvg, getConvexObjects, polygonalAssistedWatershed, removeShortTracks, segmentObjectsUsingMarkers, solidityFilterLabel, splitTracks.

assayFluoNuclThresholding

Usage

This module is used to threshold a series of images (no object segmentation and no tracking are performed).

Script Variables

ImageFolder - String variable that specifies the absolute location of the directory which contains the time-lapse images. An example of such a string variable would be *‘c:/sample images/high-density’*.
ImageFilesRoot - String variable specifying the root image file name. The root image file name for a set of images is the image file name of any of the images without the number or the file extension. For example, if the file name is *'Experiment-0002_Position(8)_t021.tif'* the root image file name will be *'Experiment-0002_Position(8)_t'.*
ImageExtension – String variable specifying the image file extension including the preceding dot. For example, if the file name is *‘image003.jpg’* the image extension is *'.jpg'*.
StartFrame – Number specifying the first image in the sequence to be analyzed. The minimum value for this variable depends on the numbering of the image sequence, so if the first image in the sequence is ‘image003.tif’ then the minimum value is 3.
FrameCount – Number specifying how many images from the image sequence should be processed.
FrameStep - Number specifying the step size when reading images. Set this variable to 1 to read every image in the sequence, 2 to read every other image and so on.
NumberFormat – String value specifying the number of digits in the image file names in the sequence. For example, if the image file name is ‘*image020.jpg’* the value for the *NumberFormat* is *'%03d'*, while if the file name is ‘*image000020.jpg’* the value should be *‘%06d’*.

OutputFolder – The folder where the thresholded images will be saved. By default this value is set to a folder named *‘output’* within the folder where the images to be analyzed are located.
BrightnessThresholdPct – Number specifying the percentage threshold value for the image generated by the *generateBinImgUsingLocAvg* filter. Any pixel in the original image smaller than the threshold value times the corresponding value in the local average image below this value will be set to zero while the rest will be set to one.
ClearBorder – Boolean value specifying whether objects next to or touching the image border in the binary images generated by the *generateBinImgUsingGradient* module will be erased (*true*) or not (*false*).
ClearBorderDist – Number specifying how close to the border objects may be and still be erased if the *ClearBorder* parameter is set to *true* in the *generateBinImgUsingGradient* module.
ObjectArea – Number specifying the threshold area for the *clearSmallObjects, polygonalAssistedWatershed* filter. Objects below this value will be removed from the filtered image.
Strel – String variable specifying the type of filter used to generate the local average image in *generateBinImgUsingLocAvg*. Currently *‘disk’* is the only value supported.
StrelSize – Number specifying the size of the local neighborhood used to calculate the average for each pixel in the local average image generated by the *generateBinImgUsingLocAvg* module.

Important Modules

generateBinImgUsingLocAvg.

assayGetStageOffset

Usage

This assay is used to calculate the microscope stage offset at each frame by manually clicking on a fixed point in every image.

Script Variables

ImageFolder - String variable that specifies the absolute location of the directory which contains the time-lapse images. An example of such a string variable would be *‘c:/sample images/high-density’*.
ImageFilesRoot - String variable specifying the root image file name. The root image file name for a set of images is the image file name of any of the images without the number or the file extension. For example, if the file name is *'Experiment-0002_Position(8)_t021.tif'* the root image file name will be *'Experiment-0002_Position(8)_t'.*
ImageFileBase – The path name to the images. This value is generated from the *ImageFolder* and the *ImageFilesRoot* and should not be changed.
ImageExtension – String variable specifying the image file extension including the preceding dot. For example, if the file name is *‘image003.jpg’* the image extension is *'.jpg'*.
NumberFormat – String value specifying the number of digits in the image file names in the sequence. For example, if the image file name is ‘*image020.jpg’* the value for the *NumberFormat* is *'%03d'*, while if the file name is ‘*image000020.jpg’* the value should be *‘%06d’*.

StartFrame – Number specifying the first image in the sequence to be analyzed. The minimum value for this variable depends on the numbering of the image sequence, so if the first image in the sequence is 'image003.tif' then the minimum value is 3.
FrameStep - Number specifying the step size when reading images. Set this variable to 1 to read every image in the sequence, 2 to read every other image and so on.
FrameCount – Number specifying how many images from the image sequence should be processed.
XYOffsets – Initial value for the stage offsets array. Set to zero by default.

Important Modules

None.

assayOffsetFrames

Usage

This module is used to offset and crop frames to remove incorrect offsets due to errors in automatic stage return. Use with offset data acquired using an assay such as assayGetStageOffset. The original images will be untouched and the cropped images will be saved in a "cropped frames" directory under the original image folder.

Script Variables

ImageFolder - String variable that specifies the absolute location of the directory which contains the time-lapse images. An example of such a string variable would be *'c:/sample images/high-density'*.
ImageFilesRoot - String variable specifying the root image file name. The root image file name for a set of images is the image file name of any of the images without the number or the file extension. For example, if the file name is *'Experiment-0002_Position(8)_t021.tif'* the root image file name will be *'Experiment-0002_Position(8)_t'.*
ImageFileBase – String variable specifying the path name to the images. This value is generated from the *ImageFolder* and the *ImageFilesRoot* and should not be changed.
CroppedImagesFolder – String variable specifying the folder where the cropped images will be saved. Set by default to a folder named *'cropped'* within the original images folder.
ImageExtension – String variable specifying the image file extension including the preceding dot. For example, if the file name is *'image003.jpg'* the image extension is *'.jpg'*.
NumberFormat – String value specifying the number of digits in the image file names in the sequence. For example, if the image file name is '*image020.jpg'* the value for the *NumberFormat* is *'%03d'*, while if the file name is '*image000020.jpg'* the value should be *'%06d'*.
StartFrame – Number specifying the first image in the sequence to be analyzed. The minimum value for this variable depends on the numbering of the image sequence, so if the first image in the sequence is 'image003.tif' then the minimum value is 3.
FrameStep - Number specifying the step size when reading images. Set this variable to 1 to read every image in the sequence, 2 to read every other image and so on.
FrameCount – Number specifying how many images from the image sequence should be processed.

Important Modules
None.

Control Modules

forLoop

Usage
This module is used to loop through a series of modules. The modules that will be looped through are listed in the *LoopFunctions* input structure member.

Input Structure Members
EndLoop – The loop will stop when the loop counter reaches this value.
IncrementLoop – The value by which the loop counter will be incremented every time one loop cycle is completed.
LoopFunctions – The list of module input structures that will be looped through.
StartLoop – The loop counter will be initialized to this value.
Any other arguments may be added to the *FunctionArgs* structure and they will be available to the modules contained in the *LoopFunctions* structure member. This allows the loop modules to have access to values generated by modules outside the loop.

Output Structure Members
Any values generated by the modules in the loop can be made available to modules outside the loop by adding them to the *KeepValues* structure.

if_statement

Usage
This module is used to create branching execution in an assay. Depending on the result of a test function, either the modules in the *IfFunctions* or the modules in the *ElseFunctions* structures will be executed.

Input Structure Members
TestVariable – The module which will be used to determine what subset of modules will be executed. This module needs to return a *true*/*false* value.
IfFunctions – Set of modules which will be executed if the value returned by the test function is *true*.
ElseFunctions – Set of modules which will be executed if the value returned by the test function is *false*.

whileLoop

Usage
This module is used to loop through a series of modules. The modules that will be looped through are listed in the *LoopFunctions* input structure member.

Input Structure Members
TestFunction – The module which will be used to determine how many times the loop will iterate over the modules in *LoopFunctions*. This module needs to return a *true*/*false* value and be part of the *whileLoop*'s module loop functions.

Internal Modules

addFunction

Usage
This module adds two variables.

Input Structure Members
Number1 – The first variable to be added.
Number2 – The second variable to be added.

Output Structure Members
Sum – The result of the addition.

areaFilterLabel

Usage
This module is used to remove objects below and/or above a certain area from a label matrix.

Input Structure Members
MaxArea – Objects with an area larger than this value will be removed.
MinArea – Objects with an area smaller than this value will be removed.
ObjectsLabel – The label matrix from which objects will be removed.

Output Structure Members
LabelMatrix – The filtered label matrix.

areaOverPerimeterFilterLabel

Usage
This module is used to remove objects below and/or above a certain area/perimeter ratio from a label matrix. Montages consisting of multiple images can exhibit noise at the points where the individual images are joined. The objects from this type of noise have a higher area/perimeter ratio than true objects and can be removed using this filter.

Input Structure Members

MaxAreaOverPerimeter – Objects with an AOP ratio larger than this value will be removed.
MinAreaOverPerimeter – Objects with an AOP ratio smaller than this value will be removed.
ObjectsLabel – The label matrix from which objects will be removed.

Output Structure Members

LabelMatrix – The filtered label matrix.

assignCellToTrackUsingAll

Usage

This module tracks objects in a time-lapse sequence of label matrices. It uses distance, speed, direction and shape parameters to determine which candidate object best matches a track. It can be optimized for tracking fast-moving directional objects or slow-moving objects.

Input Structure Members

CheckCellPath – To allow paths that go through another cell, set this value to true, otherwise set it to false.
CellsCentroids – Current object centroids.
CellsLabel – The current time step label matrix.
CurrentTracks – Matrix containing the track assignments and shape parameters for the objects in the previous frame.
DefaultParamWeights – A set of weights is assigned to each parameter based on its prediction power. Parameters with high prediction power are assigned high weights and parameters with low prediction power are assigned lower weights. If an object can be assigned to a matching group or is a sure match for a track the weights listed in this variable are used.
DirectionRankingOrder – When the order of the parameters based on prediction power cannot be determined using a group match, a set of default parameter ranks is used. The parameter order for fast directional objects is provided in this variable.
DistanceRankingOrder – When the order of the parameters based on prediction power cannot be determined using a group match, a set of default parameter ranks is used. The parameter order for slow-moving objects is provided in this variable.
ExcludedTracks – List of track assignments that should not be changed.
FrontParams – To force a set of parameters to the front so they are heavily weighted, enter their column indices in the variable; otherwise set the variable to an empty vector.
MatchingGroups – Matrix of current matching groups (vectors of shape or motility indices ordered by their prediction power).
MatchingGroupsStats – Matrix of mean values for each parameter in each group.
MaxAngleDiff – The maximum allowed angle difference between a track and a candidate object. If the angle is larger than this value, direction ranking will not be used for this object.
MaxDistRatio – The maximum allowed distance ratio between the two nearest candidate objects. If the ratio is higher than this value, distance ranking will not be used.

MaxSearchRadius - Sets an absolute lower bound for the search radius to prevent selecting too few candidate objects for a track.
MaxTrackID – Current maximum track ID.
MinSecondDistance – Minimum significant distance between the closest candidate object to a track and the second closest. Used to determine when distance should be used as a ranking parameter.
MinSearchRadius – Sets an absolute higher bound for the search radius to prevent selecting too many candidate objects for a track.
NrParamsForSureMatch – This value is used to indicate the minimum number of closest matches between a candidate object parameters and a track's object parameters that make the candidate object a sure match to the track.
ParamsCoeffOfVariation – Matrix containing the coefficient of variation for each parameter.
PreviousCellsLabel – The matrix label from the previous time step.
PreviousTracks – Matrix containing the track assignments and shape parameters for the objects in the frame previous to those in *CurrentTracks*.
RelevantParametersIndex – Shape or motility parameters that have been determined to be irrelevant for tracking the objects can be eliminated by setting the corresponding index in the variable to false. This indicates to the module not to use the parameters in computing track assignment probabilities.
SearchRadiusPct – This value determines the size of the neighborhood from which candidate objects for matching the track are selected. It is a multiple of the distance to the nearest candidate in the current frame. Setting this variable equal to 1 turns this module into a nearest-neighbor algorithm (only the nearest cell can be a candidate). It does not make sense to have a value lower than 1.
ShapeParameters – Shape parameters (area, eccentricity, perimeter, etc.) extracted from the current label.
TrackAssignments – List of track assignments that have already been completed.
TracksLayout – Matrix describing the order of the columns in the tracks matrix.
UnassignedCells – List of object IDs currently unassigned.
UnknownParamWeights – A set of weights is assigned to each parameter based on its prediction power. Parameters with high prediction power are assigned high weights and parameters with low prediction power are assigned lower weights. If an object cannot be assigned to a matching group and is not a sure match for a track the weights listed in this variable are used.
UnknownRankingOrder – When the order of the parameters based on prediction power cannot be determined using a group match, a set of default parameter ranks is used. If the objects cannot be categorized as either slow-moving or fast directional the parameter order provided in this variable is used.

Output Structure Members

UnassignedIDs – List of object IDs currently unassigned.
TrackAssignments – List of track assignments that have already been completed.
MatchingGroups – Matrix of mean values for each parameter in each group.
GroupIndex – Indicates the index of the group to which the object has been assigned.

ExcludedTracks – List of track assignments that should not be changed.

assignCellToTrackUsingNN

Usage

This module tracks objects in a time-lapse sequence of label matrices using a nearest-neighbor algorithm.

Input Structure Members

CellsCentroids – Current object centroids.
CurrentTracks – Matrix containing the track assignments for the objects in the previous frame.
MaxTrackID – Current maximum track ID.
TrackAssignments – List of track assignments that have already been completed.
TracksLayout – Matrix describing the order of the columns in the tracks matrix.
UnassignedCells – List of object IDs currently unassigned.

Output Structure Members

ExcludedTracks – Returns empty value. Included for compatibility only.
GroupIndex – Returns zero. Included for compatibility only.
MatchingGroups – Returns empty value. Included for compatibility only.
TrackAssignments – List of track assignments that have already been completed.
UnassignedIDs – List of object IDs currently unassigned.

calculateCropSize

Usage

This module is used to calculate how much a image will need to be cropped to remove the microscope stage offsets.

Input Structure Members

Image – Image in the sequence to be cropped..
XYOffsets – Array containing the offsets for all the images in the series.

Output Structure Members

CropSize – Array indicating how much the image will need to be cropped in each direction.

clearSmallObjects

Usage

This module is used to remove objects below a certain area from a binary image.

Input Structure Members

Image – Binary image from which objects will be removed.
MinObjectArea – Objects with an area smaller than this value will be removed.

Output Structure Members
Image – Filtered binary image.

combineImages

Usage
This module is used to combine two binary images using logical operations.

Input Structure Members
CombineOperation – String value indicating the logical operation used to combine the images. Currently, only ‘AND’ and ‘OR’ are supported.
Image1 – First binary image.
Image2 – Second binary image.

Output Structure Members
Image – Binary image resulting from the logical operation.

compareValues

Usage
This module is used to compare two numerical values using the specified operation.

Input Structure Members
Arg1 – First numerical value.
Arg2 – Second numerical value.
Operation – String representing the mathematical operation to be performed. Currently, ‘>’,’<’,’>=’,’<=’ and ’==’ are supported.

Output Structure Members
BooleanOut – The result of the operation. Can be either 1 (true) or 0 (false).

continueTracks

Usage
This module is used to continue the tracks with the new track assignments as tracking progresses from frame to frame.

Input Structure Members
CurFrame – Integer value representing the current frame number.
TrackAssignments – Matrix containing the new track assignments.
TimeFrame – Integer value representing the current time frame.

Output Structure Members
NewTracks – Matrix containing the new tracks for the current frame.
Tracks – Matrix containing the tracks including the new track assignments.

cropImage

Usage
This module is used to crop an image from a point specified from the (0,0) coordinate to the specified size.

Input Structure Members
CropSize – Two number vector containing the desired dimensions for the cropped image.
Image – The image to be processed.
XYOffset – Offset point (from the (0,0) coordinate) from which the image should be cropped.

Output Structure Members
Image – Cropped image.

cropImageWithOffset

Usage
This module is used to crop an image from a point specified from the center of the image to the specified size.

Input Structure Members
CropSize – Two number vector containing the desired dimensions for the cropped image.
Image – The image to be processed.
XYOffset – Offset point (from the center of the original image) at which the cropped image should be centered.

Output Structure Members
Image – Cropped image.

detectMergeCandidatesUsingDistance

Usage
This module is used to detect tracks that are never further than a small distance apart for possible merging.

Input Structure Members
MaxMergeDistance – Maximum distance between tracks. Any tracks that drift further apart than this distance at any time point will be merged.
TrackIDs – Track IDs to be tested for merging.
Tracks – Tracks matrix including time stamps and object centroids.
TracksLayout – Matrix describing the order of the columns in the tracks matrix.

Output Structure Members
TracksToBeMerged – Paired list of the track IDs to be merged.

detectMitoticEvents

Usage

This module is used to detect mitotic events in time-lapse movies of cells stained with a nuclear stain.

Input Structure Members

MaxSplitArea – Maximum area a nucleus may have at the time it splits.
MaxSplitDistance – Nuclei that are further apart than this distance will not be considered as a potential split pair.
MaxSplitEccentricity – Any nucleus with an eccentricity above this value will not be considered a split candidate.
MinSplitEccentricity – Any nucleus with an eccentricity below this value will not be considered a split candidate.
MinTimeForSplit – A cell needs to have a lifespan above this value to be considered a split candidate.
Tracks – Tracks matrix including time stamps and object centroids.
TracksLayout – Matrix describing the order of the columns in the tracks matrix.
UntestedIDs - Track IDs to be tested for mitosis.

Output Structure Members

SplitCells – List of mitotic cell pairs.

displayAncestryData

Usage

This module is used to overlay cell outlines (using different colors to indicate different cell generations) and cell labels on the original images after tracking and save the resulting image.

Input Structure Members

AncestryLayout – Matrix describing the order of the columns in the tracks matrix.
CellsAncestry – Matrix containing the ancestry records for the cells in the image.
CellsLabel – The label matrix containing the cell outlines for the current image.
ColorMap – Color map to be used in drawing the cell outlines for each generation. Each generation will use the next color in the color map until all colors have been used. Afterwards, the colors in the map are recycled.
CurFrame – Integer containing the current frame number.
CurrentTracks – The list of the tracks for the current image.
DS – The directory separator to be used when generating file names ("\" for Windows, "/" for Unix/Linux).
Image – The original image which will be used to generate the image with overlayed outlines and labels.
ImageFileName – The root of the image file name to be used when generating the image file name for the current image in combination with the current frame number.

NumberFormat – A string indicating the number format of the file name to be used when saving the overlayed image.
ProlDir – Output directory where the resulting image will be saved.
TracksLayout – Matrix describing the order of the columns in the tracks matrix.

Output Structure Members

None.

displayImage

Usage

This module is used to display an image in the specified figure number.

Input Structure Members

FigureNr – The figure number where the image should be displayed.
Image – Matrix containing the image to be displayed.

Output Structure Members

None.

displayTracksData

Usage

This module is used to overlay cell outlines (using different colors to indicate different cell generations) and cell labels on the original images after tracking and save the resulting image.

Input Structure Members

CellsLabel – The label matrix containing the cell outlines for the current image.
CurFrame – Integer containing the current frame number.
CurrentTracks – The list of the tracks for the current image.
FileRoot – The root of the image file name to be used when generating the image file name for the current image in combination with the current frame number.
Image – The original image which will be used to generate the image with overlayed outlines and labels.
NumberFormat – A string indicating the number format of the file name to be used when saving the overlayed image.
TracksLayout – Matrix describing the order of the columns in the tracks matrix.

Output Structure Members

None.

displayVariable

Usage

This module is used to display a variable name and its value.

Input Structure Members
Variable – The variable whose value is to be displayed.
VariableName – The variable name to be displayed along with the variable value.

Output Structure Members
None.

distanceWatershed

Usage
This module is used to compute the distance watershed of a binary image.

Input Structure Members
Image – The binary image for which the distance watershed will be computed.
MedianFilterNhood – The size of the median filter which will be used to smooth the watershed.

Output Structure Members
LabelMatrix – The result of the distance watershed.

eccentricityFilterLabel

Usage
This module is used to remove objects below and/or above a certain eccentricity from a label matrix.

Input Structure Members
MaxEccentricity – Any object with an eccentricity higher than this value will be removed.

MinEccentricity – Any object with an eccentricity lower than this value will be removed.

ObjectsLabel – The label matrix from which the objects will be removed.

Output Structure Members
LabelMatrix – The filtered label matrix.

generateBinImgUsingGlobInt

Usage
This module is used to convert a grayscale image to binary using a global intensity threshold.

Input Structure Members
ClearBorder – If this value is set to *true,* objects that are within *ClearBorderDist* of the image edges will be erased.

ClearBorderDist – Objects that are within this distance from the edges of the image will be erased if *ClearBorder* is set to *true*.
Pixel intensities greater than the threshold intensity are converted to 1 in the binary image.
Image – Grayscale image to be converted.
IntensityThresholdPct – This is a percentage of the intensity range of the image. The threshold intensity value is calculated as *IntensityThresholdPct*double(max_pixel-min_pixel)+min_pixel.*

Output Structure Members

Image – Resulting binary image.

generateBinImgUsingGradient

Usage

This module is used to convert a grayscale image to a binary image using values of the image gradient.

Input Structure Members

ClearBorder – If this value is set to *true,* objects that are within *ClearBorderDist* of the image edges will be erased.
ClearBorderDist – Objects that are within this distance from the edges of the image will be erased if *ClearBorder* is set to *true*.
GradientThreshold – Areas in the gradient image where the gradient is higher than this value will be set to 1 in the binary image.
Image – Grayscale image to be converted.

Output Structure Members

Image – Resulting binary image.

generateBinImgUsingLocAvg

Usage

This module is used to convert a grayscale image to a binary image using local average values.

Input Structure Members

BrightnessThresholdPct – This value indicates what percentage of the local average value will be used to threshold the image. If the pixel intensity is higher than this value times the local average value the corresponding pixel in the binary image will be set to one.
ClearBorder – If this value is set to *true,* objects that are within *ClearBorderDist* of the image edges will be erased.
ClearBorderDist – Objects that are within this distance from the edges of the image will be erased if *ClearBorder* is set to *true*.
Image – Grayscale image to be converted.
Strel – Filter type used to generate the local average image. Currently *'circular'* is the only value supported.

StrelSize – Size of the local neighborhood used to calculate the average for each pixel in the image.

Output Structure Members

Image – Resulting binary image.

getArrayVal

Usage

This module returns a value or set of values from an array.

Input Structure Members

Array – Array from which the value is to be extracted.
Index – The index of the values to be returned.

Output Structure Members

ArrayVal – Set of values extracted from the array.

getCentroids

Usage

This module returns a list of centroids for the objects in a label matrix.

Input Structure Members

LabelMatrix – Label matrix from which the object centroids will be calculated.

Output Structure Members

Centroids – The list of centroids extracted from the label matrix.

getConvexObjects

Usage

This module is used to find and return the index of convex objects in a binary image. The object outlines are simplified using a Douglas-Pecker algorithm to prevent detection of insignificant convexities.

Input Structure Members

ApproximationDistance – This value represents the minimum distance between the approximated outline and the real one. Increasing this value makes the contours simpler but less like the original outlines.
Image – Binary image to be processed.

Output Structure Members
ConvexObjectsIndex – List containing the index of the convex objects. The index of each object is based on performing a *bwlabeln* operation on the binary image.

getCurrentTracks

Usage
Module to extract a subset of tracks out of the tracks matrix starting with the current frame.

Input Structure Members
CurFrame – The current frame index.
FrameStep – How many frames to skip when reading the track subset. If every frame is to be read set this value to 1.
MaxMissingFrames – This value indicates if tracks not present in the current frame should be included in the track subset and if so how many frames away from the current frame a track is allowed to be and still be included in the subset. Setting this value to zero ensures that only tracks present in the current frame are included.
OffsetFrame – The number of frames from the current frame the subset should include. This value can be a positive or negative integer.
TimeCol – Index of the time column in the tracks matrix.
TimeFrame – The amount of time elapsed between each frame.
TrackIDCol – Index of the track ID column in the tracks matrix.
Tracks – Matrix containing the set of tracks from which the subset is to be extracted.

Output Structure Members
Tracks – The subset of tracks extracted from the tracks matrix.

getFileInfo

Usage
This module is used to extract the file name, extension and directory from the absolute path.

Input Structure Members
DirSep – Directory separator (“\” for Windows, “/” for Linux/Unix).
PathName – Absolute path name from which file name, extension and directory will be extracted.

Output Structure Members
DirName – The extracted directory name.
FileName – The extracted file name.
ExtName – The extracted extension name.

getMaxTrackID

Usage
This module is used to return the current maximum track ID from a track matrix.

Input Structure Members
TrackIDCol – Index of the track ID column in the tracks matrix.
Tracks – Matrix containing the set of tracks.

Output Structure Members
MaxTrackID – The maximum track ID.

getObjectIntensities

Usage
This module is used to return the object intensities from objects in a label matrix using the corresponding intensity image.

Input Structure Members
LabelMatrix – The label matrix containing the objects for which the mean intensity data will be extracted.
IntensityImage – Matrix containing the intensity image.

Output Structure Members
MeanIntensities – Array containing the mean intensities for each object in the label matrix.

getObjectsMeanDisplacement

Usage
This module estimates the average displacement of the cells in the frame using the centroids from the previous and current frame.

Input Structure Members
Centroid1Col – The index of the first coordinate of the object centroids in the tracks matrix.
Centroid2Col – The index of the second coordinate of the object centroids in the tracks matrix.
CurrentTracks – Set of tracks belonging to previous frame.
ObjectCentroids – The list of centroids in the current frame.

Output Structure Members
MeanDisplacement – The estimated mean displacement of the objects in the frame.
SDDisplacement – The estimated standard deviation of the objects in the frame.

getShapeParams

Usage

This module is used to extract the shape parameters (Area, Eccentricity, etc.) of objects in a label matrix. Mostly, a wrapper for the MATLAB *regionprops* function. Adds a blob id to the output so that objects that belong to the same blob may be identified at a later time.

Input Structure Members

LabelMatrix – The label matrix from which the shape parameters will be extracted.

Output Structure Members

Centroids – The centroids of the objects in the label matrix.
ShapeParameters – The shape parameters of the objects in the label matrix.

getShapeParamsWithDisconnects

Usage

This module is used to extract the shape parameters (Area, Eccentricity, etc.) of objects in a label matrix. The difference between this module and *getShapeParams* is that *getShapeParamsWithDisconnects* supports objects which are disjointed (spread across multiple blobs) at the cost of execution speed.

Input Structure Members

LabelMatrix – The label matrix from which the shape parameters will be extracted.

Output Structure Members

Centroids – The centroids of the objects in the label matrix.
ShapeParameters – The shape parameters of the objects in the label matrix.

getTrackIDs

Usage

This module is used to retrieve the list of track IDs from the track matrix.

Input Structure Members

TrackIDCol – Index of the track ID column in the tracks matrix.
Tracks – Matrix containing the set of tracks from which IDs will be extracted.

Output Structure Members

TrackIDs – The IDs extracted from the tracks matrix.

holdValue

Usage

The only purpose for this module is to hold a value so that other modules may use it, for example by having the *holdValue* module at a higher level in the hierarchy, thereby preventing modules at lower levels from overwriting the value.

Input Structure Members

ValueToHold – The value to hold.

Output Structure Members

ValueToHold – The held value.

imNorm

Usage

This module is used to normalize an image so that the lowest pixel value is zero and the highest pixel value is the maximum allowed value for the specified integer class.

Input Structure Members

IntegerClass – The integer class of the image. Has to be an integer class supported by MATLAB such as 'int8' or 'uint8'.
RawImage – The image to be processed.

Output Structure Members

Image – The normalized image.

loadAncestryLayout

Usage

This module is used to load a structure containing the order of each column in the ancestry matrix from a MATLAB .mat file.

Input Structure Members

FileName – The name of the .mat file containing the ancestry layout structure.

Output Structure Members

AncestryLayout – Structure containing the order of each column in the ancestry matrix.

loadCellsLabel

Usage

This module is used to load cells/nuclei labels. It looks for a variable named cells_lbl in the .mat data file.

Input Structure Members

MatFileName – The name of the data file.

Output Structure Members

LabelMatrix – The label matrix loaded from the file.

loadTracksLayout

Usage

This module is used to load a structure containing the order of each column in the tracks matrix from a MATLAB .mat file.

Input Structure Members

FileName – The name of the .mat file containing the tracks layout structure.

Output Structure Members

TracksLayout – Structure containing the order of each column in the tracks matrix.

makeAncestryForCellsEnteringFrames

Usage

This module is used to add ancestry records for cells entering the field of view after the first frame (not the result of a mitotic event in the field of view).

Input Structure Members

CellsAncestry – Current cell ancestry records.
FirstFrameIDs – The IDs of tracks starting in the first frame.
SplitCells – The IDs of tracks that are the result of mitosis.
TimeCol – The index of the time column in the tracks matrix.
TrackIDCol – The index of the track ID column in the tracks matrix.
TrackIDs – The IDs of all the tracks.
Tracks – The matrix containing all the tracks.

Output Structure Members

CellsAncestry – The current cell ancestry record with ancestry of cells entering the field of view after the first frame appended to the end.

makeAncestryForFirstFrameCells

Usage

This module is used to create ancestry records for cells present in the first frame of a time-lapse movie. These cells are generation zero.

Input Structure Members

TimeCol – The index of the time column in the tracks matrix.
TrackIDCol – The index of the track ID column in the tracks matrix.
TrackIDs – The IDs of all the tracks.
Tracks – The matrix containing all the tracks.

Output Structure Members

CellsAncestry – The ancestry records for the cells in the first frame.
FirstFrameIDs – IDs of cells in the first frame.
UntestedIDs – IDs of cells in the movie that were not present in the first frame.

makeExcludedTracksList

Usage

This module wraps its input, a list of cell IDs, in a MATLAB cell array.

Input Structure Members

UnassignedCellsIDs – The list of cell IDs.

Output Structure Members

ExcludedTracks – The MATLAB cell array.

makeImgFileName

Usage

This module is used to build the filename of a frame from a time-lapse movie using the root file name, current frame number and a specified number format and file extension.

Input Structure Members

CurFrame – The index of the current frame.
FileBase – The root of the image file name.
FileExt – The file extension.

NumberFmt – The number format to be used. See MATLAB *sprintf* documentation for format documentation.

Output Structure Members

FileName – String containing the resulting file name.

makeUnassignedCellsList

Usage

This module is used to create a list of IDs for each cell centroid in a list.

Input Structure Members

CellsCentroids – List of cell centroids.

Output Structure Members

UnassignedCellsIDs – List of IDs.

manualSegmentationReview

Usage

This module is used to manually correct errors in automatic segmentation. The module loads a GUI for each frame with all the available segmentation corrections. To start the user has to select between blob correction and object correction by clicking on the *"Select Blob"* or *"Select Object"* button. A blob is a contiguous region as defined by the background pixels and it may contain one or more objects. An object is the set of pixels with the same non-zero ID in the label matrix. An object may span multiple blobs.

In general (with the exception of the *"Restore Blob"* operation), an object or blob must be selected before an operation can be performed on it. Selection is performed by clicking on the object or blob of interest. A selected item is indicated by a checkerboard pattern. If the "Select Multiple" box is checked, clicking on an unselected item adds it to the selection.

The types of operations that may be performed on a blob are resegmentation, deletion and restoration. To resegment a blob one needs to indicate how many objects the new blob will contain and their approximate boundaries. Clicking on the selected blob after the *"Resegment Blob"* button has been pressed indicates that the pixels at those locations belong to the first object in the blob. To move to the next object, press the letter *"n"* on the keyboard and click within the blob to indicate rough boundaries. The boundaries do not have to be specified precisely and a blob may be separated into two objects in as few as two clicks. Once all the objects have been defined, press the letter *"d"* on the keyboard and the blob will be resegmented. During resegmentaion all the pixels in the blob are assigned to objects using a nearest-neighbor classifier based on the pixels selected by the user. A blob may be deleted by selecting it and then clicking the *"Remove Blob"* button. To restore a blob removed by mistake, click on the *“Restore Blob”* button. The GUI will display the *“Raw Label”* image which shows all the blobs present after thresholding before any removal by filters or manual deletions. Choose the blob to restore by clicking on it.

Two types of operations can be performed on objects: joining and deletion. To join a number of objects into a single object, click on the *“Join Objects”* button then select the objects you want to join by clicking on them. When you are done with object selection and want to join the objects, press the *“d”* letter on the keyboard. To delete an object, select it, then click on the *“Remove Object”* button.

Once all the corrections have been performed, click on the *“Save Changes & Continue”* button to save your changes and move on to the next frame.

Input Structure Members

Image – The microscopy image for the current *ObjectsLabel*.
ObjectsLabel – The label matrix containing the objects for which the automatic segmentation will be evaluated or corrected.
PreviousLabel – The label matrix containing objects from the previous time step.
RawLabel – The label matrix containing the objects before filtering.

Output Structure Members

LabelMatrix – The label matrix containing manual corrections, if any.

manualTrackingReview

Usage

The manual tracking review module can be used to correct errors in automatic tracking and as a visualization module to select subsets of tracks based on speed, motility and ancestry parameters. For track correction there are six actions that can be performed: continue a track, switch tracks, delete a track, break a track, add a split and remove a split. The GUI displays population statistics at the top under the menu bar and individual cell statistics (if a cell is selected) in a text box on the right side. Detected cell outlines and cell IDs may be displayed by checking the *“Outlines”* and/or *“Labels”* checkboxes.

Track continuation can be used when automatic tracking loses a cell it is tracking and starts a new track for that same cell. To continue the pre-existing track with the new track, select the pre-existing track. Selecting a track is done by clicking on the cell body. After selecting the pre-existing track, click on the *“Continue Track”* button, navigate to a frame where the new track exists and click on the cell body again to select it. The new track is then appended to the pre-existing track. To see the updated track navigate to another frame.

Switching tracks can be used to correct errors resulting from tracks being switched from one cell to the other during automatic tracking. To switch the tracks, navigate to the frame where the error occurs, click on the first cell, click on the *“Switch tracks”* button and finally click on the second cell. To see the updated tracks navigate to another frame.

To erase a track, select it, then click on the *“Delete Track”* button.

Breaking a track splits the track in two at the current frame. The newly created track starts at the current frame and the old track ends at the frame before that. This can be used to correct complex errors by separating a track into smaller segments than using the other actions to fix the automatic tracking errors.

Errors in mitotic event detection can be corrected using the “Add Split” and “Remove Split” buttons. To correct a missed mitotic event click on the parent cell (the cell with the older track), click on the “Add Split” button, then click on the second cell that is part of the mitotic event. The older track is broken at the current frame, a new track is created and the ancestry records are updated for all three tracks. To remove a spurious mitotic event select the track you want to use to continue the parent track, and then click on the “Remove Split” button. The new track is merged with the parent track and the ancestry records are updated for both the parent track and the other remaining track.

The visualization part of the GUI is controlled using the “Manage Selection Layers” button. A selection layer is a transparency overlaid on the original image that highlights certain cells based on a user-defined criterion. The criterion for comparison may be an exact value (such as all cells with an area larger than 500 square pixels) or a percentage (cells with an area in the top 20%). Any combination of shape, motility and ancestry parameters may be used alone or in combination to define a layer. This allows the user to define layers that are either very broad in scope, such as all cells that are larger than average in a movie, or extremely tailored, such as selection for small, rounded, fast cells with a specific parent ID. Multiple layers may be present at one time and, due to the use of transparencies and a broad selection of layer colors, cells that are part of multiple layers can be detected. The layers are automatically updated as the user moves backward or forward through the timelapse sequence, and the resulting images themselves may be saved.

To define a selection layer, click on the “Manage Selection Layers” button, then click the “Add Layer” button. Type in the name of the layer, select a layer color from the dropdown box, then add a number of conditions. Conditions may be combined using “AND” and “OR” logical connectors. A condition consists of a property such as “Area” or “Cell ID”, an operation (“=”,”<”,”>” are supported) and a value. The value can be either an absolute number or a percentage. Once all the conditions have been set, click the “Save Layer” button, then close the selection layers GUI to apply the layer. To delete a layer, click on “Manage Selection Layers”, then select the layer to be removed and click on the “Remove Layer” button.

Input Structure Members

AncestryLayout – Matrix describing the order of the columns in the tracks matrix.
CellsAncestry – Matrix containing cell ancestry records.
ColorMap – Color map to be used in drawing the cell outlines for each generation. Each generation will use the next color in the color map until all colors have been used. Afterwards, the colors in the map are recycled.
FirstFrameIDs – The IDs of tracks starting in the first frame.
FrameCount – The number of frames to track.
FrameStep – Read one out of every x frames when reading the image set. Default value is one, meaning every frame will be read.

ImageFileBase – The root file name of the images in the sequence. For example, if the image names in the time-lapse sequence are “Experiment-0002_Position(8)_t001.jpg”,”Experiment-0002_Position(8)_t002.jpg”, etc., the root image file name is “Experiment-0002_Position(8)_t”.
ImgExt – String indicating the image file extension. Usually, “.jpg” or “.tif”.
MaxMissingFrames – This value indicates if tracks not present in the current frame should be included in the track subset and if so how many frames away from the current frame a track is allowed to be and still be included in the subset.
NumberFormat – A string indicating the number format of the file name to be used when saving the overlayed image.
SegFileRoot – The root of the data file name containing the segmented objects.
StartFrame – Integer indicating at which frame the tracking should start.
SplitCells – The IDs of tracks that are the result of mitosis.
TimeCol – The index of the time column in the tracks matrix.
TimeFrame – Time interval between consecutive frames.
TrackIDCol – The index of the track ID column in the tracks matrix.
TrackIDs – The IDs of all the tracks.
Tracks – The matrix containing all the tracks (track IDs and shape parameters for every cell at every time point).
TracksLayout – Matrix describing the order of the columns in the tracks matrix.

Output Structure Members

CellsAncestry – Matrix containing corrected cell ancestry records.
Tracks – The matrix containing all the corrected tracks (track IDs and shape parameters for every cell at every time point).

mergeTracks

Usage

This module is used to merge a list of track pairs. This module is not used to determine whether a set of tracks *should* be merged. Other modules are provided for that purpose, such as *detectMergeCandidatesUsingDistance*.

Input Structure Members

FrameCount – The number of frames that are being processed.
FrameStep – How many frames to skip when reading frames. Setting this value to one will cause all the frames to be read.
NumberFormat – A string indicating the number format of the file name to be used when saving the overlayed image.
SegFileRoot – The root of the data file name containing the segmented objects.
StartFrame – The first frame in the processed set.
TimeFrame – Time interval between consecutive frames.
Tracks – The current set of tracks.
TracksLayout – Matrix describing the order of the columns in the tracks matrix.

TracksToBeMerged – The matrix of track IDs to be merged. Primary IDs (the IDs which will remain after the merge) are listed in the first column and secondary IDs (the IDs which will be removed after the merge) are listed in the second column of the matrix.

Output Structure Members
Tracks – The new track set with the results from the merge incorporated.

negativeImage

Usage
This module returns the negative of the image provided as an argument.

Input Structure Members
Image – Image to be processed.

Output Structure Members
Image – Negative image.

overlayAncestry

Usage
This module is used to overlay the cell outlines (color-coded according to generation number) and the track ids on top of the original cell image.

Input Structure Members
AncestryLayout – Matrix describing the order of the columns in the ancestry matrix.
CurrentTracks – The set of tracks for the current image.
CellsLabel – The label matrix containing the detected cell shapes for the current image.
CellsAncestry – Matrix containing the ancestry records for the cells in the time-lapse movie.
ColorMap – Color map to be used in drawing the cell outlines for each generation. Each generation will use the next color in the color map until all colors have been used. Afterwards, the colors in the map are recycled.
Image – This is the original cell image.
ShowLabels – Boolean value. If set to false the cell IDs will not be overlayed.
ShowOutlines – Boolean value. If set to false the cell outlines will not be overlayed.
TracksLayout – Matrix describing the order of the columns in the tracks matrix.

Output Structure Members
Image – The overlayed image.

percentageForeground

Usage
This module calculates the percentage of foreground pixels in a binary image.

Input Structure Members
Image – Binary image for which the percentage of foreground pixels is to be calculated.

Output Structure Members
PercentageForeground – The percentage of foreground pixels.

polygonalAssistedWatershed

Usage
This module is used to prevent a watershed segmentation module (can be another type of segmentation module) from splitting convex objects. The assumption is that convex objects are atomic and should not be split. This assumption works well for nuclear stains.

Input Structure Members
ConvexObjectsIndex – List containing the index of convex objects. This list may be generated using the *getConvexObjects* module.
ImageLabel – Label matrix containing the original objects before segmentation.
MinBlobArea – Objects resulting from segmentation that have an area smaller than this value will be unsegmented.
WatershedLabel – Label matrix containing the objects after segmentation.

Output Structure Members
LabelMatrix – A label matrix containing the objects segmented according to the segmentation in *WatershedLabel* with the exception of convex objects, which are left unaltered.

refineSegmentation

Usage
This module is used to retain only objects in a label matrix that are nearest to objects in another matrix.

Input Structure Members
CurrentLabel – The label matrix from which objects may be removed if they don't have an object to which they are nearest in the *PreviousLabel* matrix.
PreviousLabel – The objects in this label will determine the objects that will be retained in the current label.

Output Structure Members
LabelMatrix – The filtered label matrix.

removeShortTracks

Usage

This module is used to erase tracks with a lifespan shorter than a specified period of time.

Input Structure Members

AncestryLayout – Matrix describing the order of the columns in the ancestry matrix.
Tracks – The tracks matrix to be processed.
CellsAncestry – Matrix containing the ancestry records for the cells in the time-lapse movie.
TracksLayout – Matrix describing the order of the columns in the tracks matrix.
MinLifespan – Tracks with a lifespan shorter than this value will be erased.

Output Structure Members

Tracks – The filtered tracks matrix.

saveAncestry

Usage

This module is used to save the cells ancestry matrix. The matrix is saved with the variable name *cells_ancestry*.

Input Structure Members

AncestryFileName – The file name to which the cell ancestry matrix should be saved.
CellsAncestry – The matrix containing the cells ancestry records.

Output Structure Members

None

saveAncestrySpreadsheets

Usage

This module is used to save the tracks and ancestry records spreadsheets.

Input Structure Members

ProlXlsFile – The desired file name for the spreadsheet containing the ancestry records.
CellsAncestry – Matrix containing the ancestry records for the cells in the time-lapse movie.
ShapesXlsFile – The desired file name for the spreadsheet containing the tracks and shape parameters data.
Tracks – The tracks matrix to be processed.
TracksLayout – Matrix describing the order of the columns in the tracks matrix.

Output Structure Members
None.

saveCellsLabel

Usage
This module is used to save a MATLAB label matrix containing cell objects.

Input Structure Members
CellsLabel – The label matrix containing cell objects.
CurFrame – The index of the frame to which the label matrix corresponds.
FileRoot – String containing the root of the file name to be used when saving the label matrix.
NumberFormat – String indicating the number format to be used when formatting the current frame number to be concatenated to the file root string. See the MATLAB sprintf help file for example number format strings.

Output Structure Members
CellsLabel – The label matrix containing cell objects.

saveRegionPropsSpreadsheets

Usage
This module is used to save the shape parameters extracted using the *getRegionProps* wrapper module.

Input Structure Members
RegionProps – The matrix containing the shape parameters.
SpreadsheetFileName – The desired file name for the saved file.

Output Structure Members
None.

saveTracks

Usage
This module is used to save the tracks matrix.

Input Structure Members
Tracks – Matrix containing the tracks to be saved.
TracksFileName – The desired file name for the saved tracks data.

Output Structure Members
None.

segmentObjectsUsingClusters

Usage
This module is used to segment objects in a label matrix using hierarchical clustering.

Input Structure Members
ClusterDistance – The height threshold for the cluster tree. All leaves below this value will be grouped in a cluster. See documentation for the MATLAB function *cluster* for more details.
MinimumObjectArea – Objects with an area smaller than this value will be unsegmented and distributed between the neighboring objects.
ObjectsLabel – The label matrix containing the objects to be segmented.
ObjectReduce – Used to reduce the size of the objects. If the objects are too large the clustering function will run out of memory. When this happens set *ObjectReduce* to a value lower than one.

Output Structure Members
LabelMatrix – The label matrix containing the segmented objects.

segmentObjectsUsingMarkers

Usage
This module is used to segment objects in a label matrix using markers from another label matrix.

Input Structure Members
MarkersLabel – The label matrix containing the marker objects.
ObjectsLabel – The label matrix containing the objects to be segmented.

Output Structure Members
LabelMatrix – Label matrix containing the segmented objects.

setArrayVar

Usage
This module is used to set a set of values in an array.

Input Structure Members
Array – The array where the values will be entered.
Index – The index in the array where the values will be entered.

Var – The set of values that will be entered in the array.

Output Structure Members

Array – The array with the new set of values.

showImageAndPause

Usage

This module is used to show an image and pause execution.

Input Structure Members

FigureNr – The handle number of the MATLAB figure. If it doesn't exist it will be created.
Image – Matrix containing the image to be shown.

Output Structure Members

None.

showLabelMatrixAndPause

Usage

This module is used to show a MATLAB label matrix and pause execution.

Input Structure Members

FigureNr – The handle number of the MATLAB figure. If it doesn't exist it will be created.
LabelMatrix – The label matrix to be displayed.

Output Structure Members

None.

solidityFilter

Usage

This module is used to remove objects below or above a threshold solidity from a binary image.

Input Structure Members

Image – The binary image from which objects will be removed.
MaxSolidity – Objects whose solidity is above this value will be removed from the image.
MinSolidity - Objects whose solidity is below this value will be removed from the image.

Output Structure Members

Image – The filtered binary image.

solidityFilterLabel

Usage
This module is used to remove objects below or above a threshold solidity value from a MATLAB label matrix.

Input Structure Members
ObjectsLabel – The label matrix from which objects will be removed.
MaxSolidity – Objects whose solidity is above this value will be removed from the image.
MinSolidity - Objects whose solidity is below this value will be removed from the image.

Output Structure Members
LabelMatrix – The filtered label matrix.

startTracks

Usage
This module is used to start a tracks matrix.

Input Structure Members
CellsLabel – The label matrix identifying the objects in the first frame of the time-lapse.
CurFrame – The index value of the current frame.
ShapeParameters – The shape parameters for the objects in the first frame (Area, Eccentricity, etc.).
TimeFrame – The time interval between consecutive frames in the time-lapse.

Output Structure Members
Tracks – The new tracks matrix.

subtractFunction

Usage
This module subtracts the first variable from the second one.

Input Structure Members
Number1 – The first variable.
Number2 – The variable to be subtracted.

Output Structure Members
Difference – The result of the subtraction.