

From imagination to impact

Track Assist User Guide

National ICT Australia
Version 2.2.0

Research Excellence in ICT Wealth Creation for Australia



Australian Government

Department of Broadband, Communications
and the Digital Economy

Australian Research Council

NICTA Funding and Supporting Members and Partners



Australian
National
University

UNSW
THE UNIVERSITY OF NEW SOUTH WALES



Trade &
Investment



State Government
Victoria



THE UNIVERSITY
OF MELBOURNE
AUSTRALIA



THE UNIVERSITY
OF SYDNEY

Queensland
Government

Griffith
UNIVERSITY

QUT
Queensland University of Technology

Introduction



- What does TrackAssist do?
 - TrackAssist is designed to enable labour-efficient **tracking** of populations of cells in time-lapse microscopy imaging. It also models the **lineages** of cells.
- Who produced TrackAssist?
 - TrackAssist was created by **NICTA** and the Immunology group at the Walter & Eliza Hall Institute (**WEHI**).
- Why was TrackAssist created?
 - New experimental protocols in Immunology indicate that the fate of B-lymphocytes is partly determined by their lineage; sibling cells have similar fates at similar times. Attempts to understand and model immune response need to consider the lineage of individual cells to explain their behaviour. The only way to determine a cell's lineage is to track generations of cells using optics.

Introduction – Inputs and Outputs



- What sort of images does TrackAssist use?
 - TrackAssist uses 8 or 16-bit time-lapse microscopy images of mobile cells
 - Data can be exported from digital confocal microscopes and loaded immediately into TrackAssist
 - The primary outputs of TrackAssist are:
 - Details of cell detections (e.g. image, coordinates, contours & bounding-box)
 - Details of cell tracks (i.e. sequences of observations of cells over time)
 - Details of cell lineages (a tree of cell tracks)
 - Labelled images in which pixel values indicate the identity of cells. For example, all pixels of cell X will have the same intensity value. This allows researchers to compute functions of segmented cell appearance in 3rd-party applications such as Matlab

Introduction – How it works

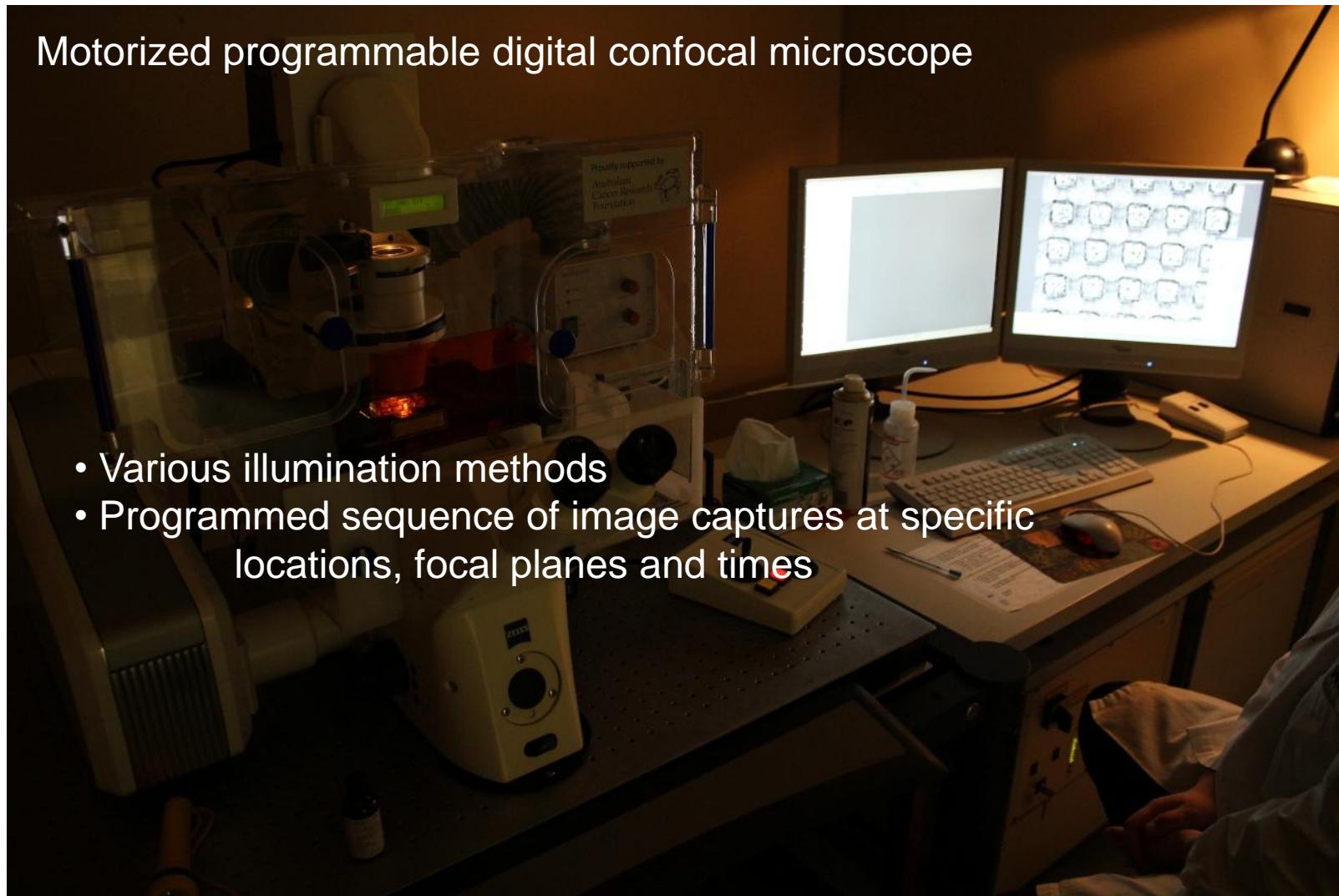


- TrackAssist is best imagined as a specialized combination of several software programs:
 - Like a video player, TrackAssist allows you to view time-lapse microscopy as a video
 - Like other biomedical imaging software, TrackAssist supports high dynamic range images, megapixel resolutions and a wide variety of image formats
 - TrackAssist uses machine vision algorithms to detect cells
 - Advanced Bayesian tracking algorithms are used to track cells
 - Cell detections are “painted” onto the images using a variety of graphical tools, some manual and some automated. These tools are similar to photo-editing software
- TrackAssist uses a combination of automatic, manual and semi-automated methods to achieve perfect tracking with minimum effort.

Introduction – Typical equipment



Motorized programmable digital confocal microscope



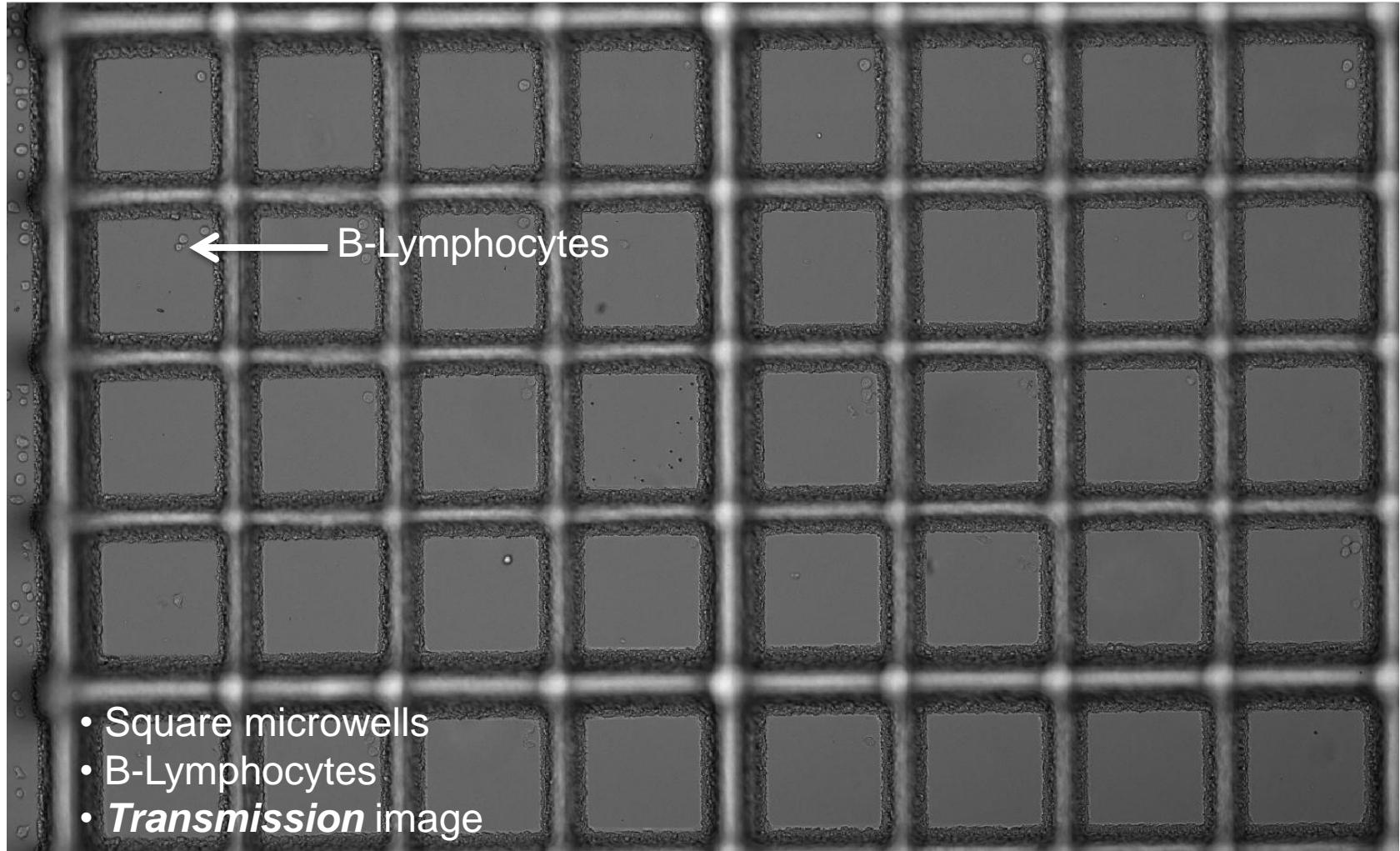
- Various illumination methods
- Programmed sequence of image captures at specific locations, focal planes and times

Introduction - Imagery

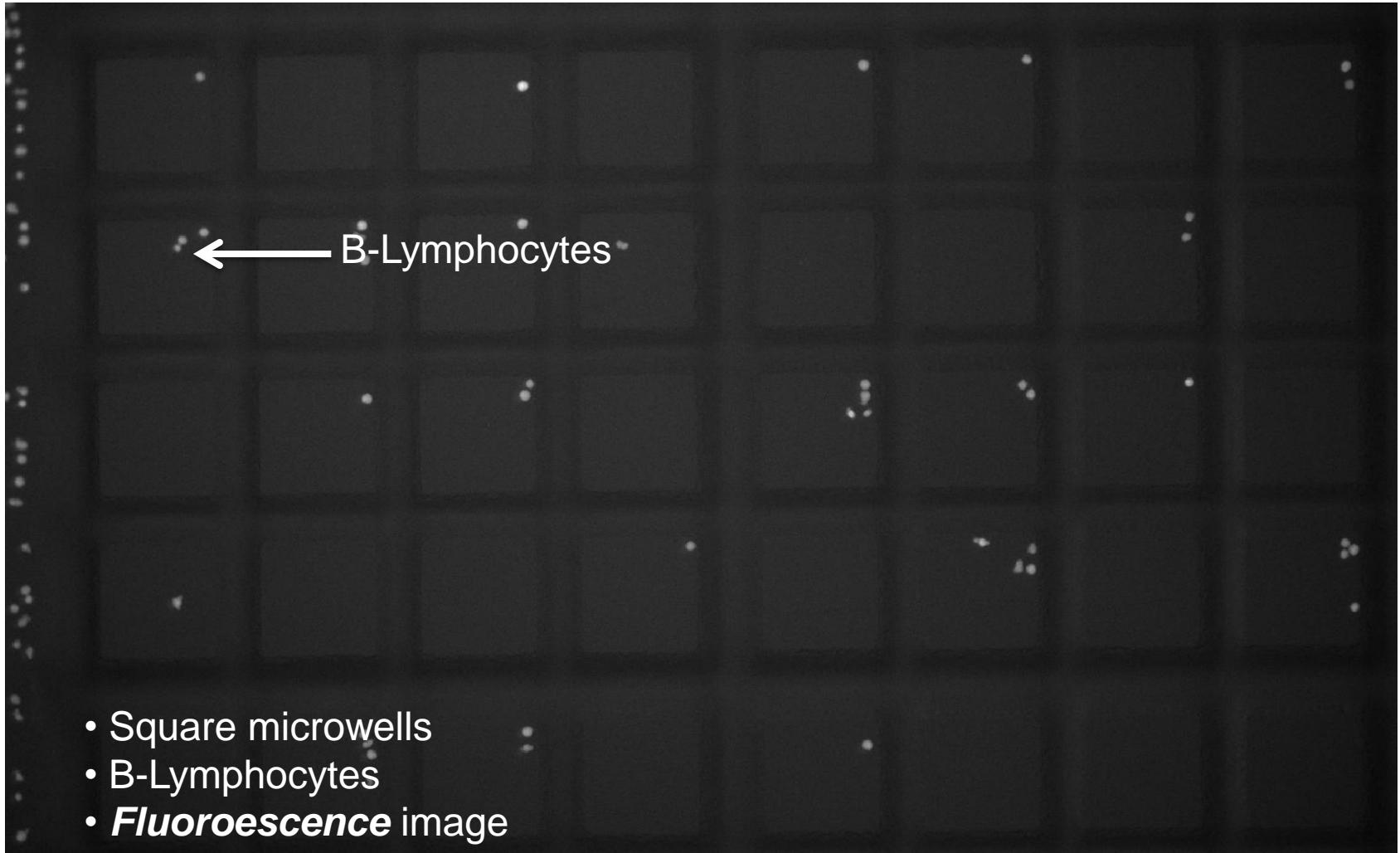


- TrackAssist supports N-dimensional image sets with any systematic naming convention
 - e.g. Position x Channel x Time
 - Often researchers capture images of cells under different illumination conditions – for example, fluorescent and transmission images
- TrackAssist assumes that the observed cells are:
 - Physically constrained into small groups by barriers called “microwells”
 - Generally arranged onto a single plane co-planar with the focal plane
 - In low densities (typically ~1 to 50 cells per microwell)
 - Mobile, i.e. they move around
 - The cells may clump together, or not
 - Generally the cells will be convex in shape

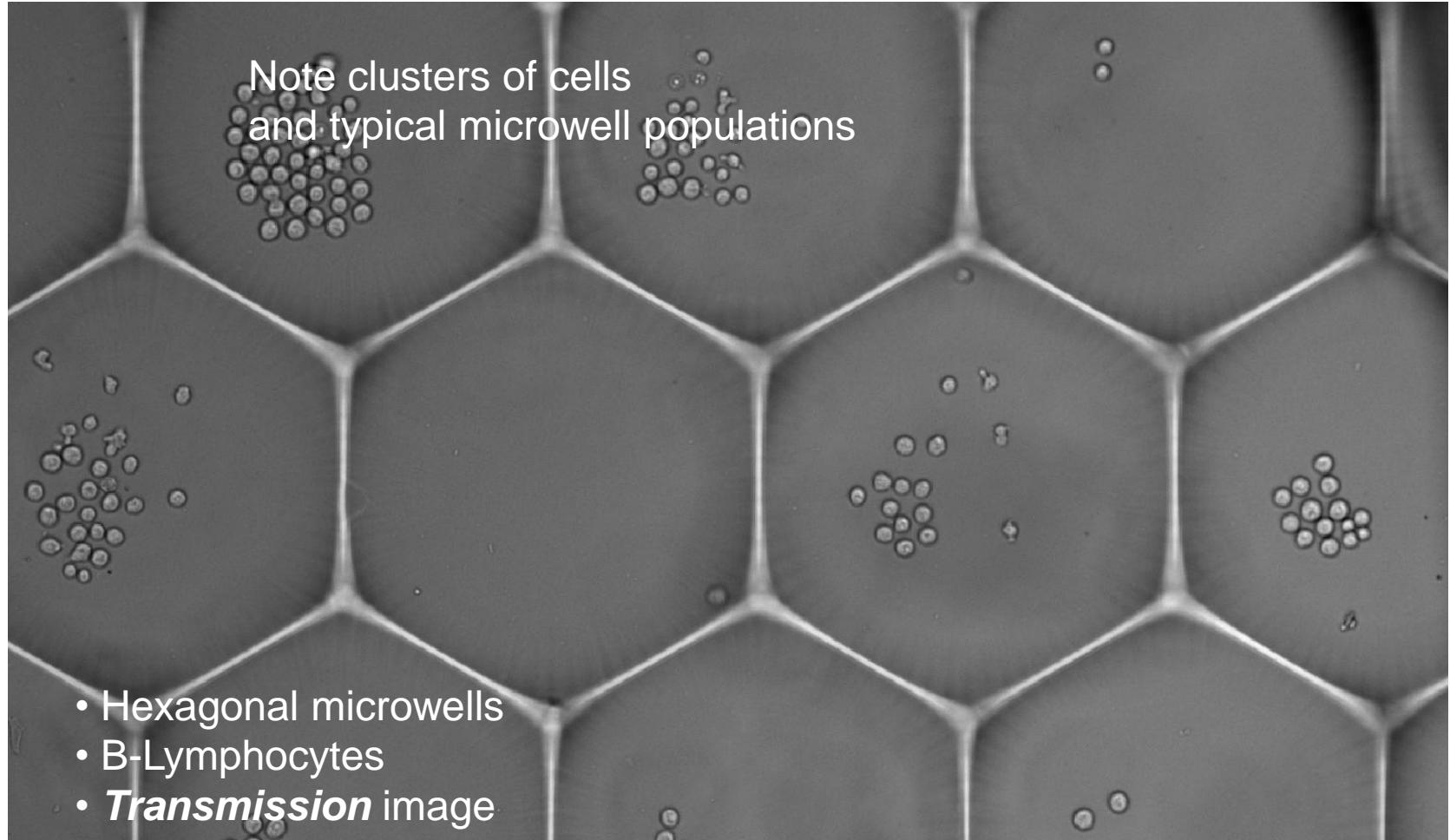
Introduction – Typical imagery



Introduction – Typical imagery



Introduction – Typical imagery



Introduction – Typical imagery



This image was captured at approximately the same time, position and focal depth as the previous one, but using a different method of illumination.

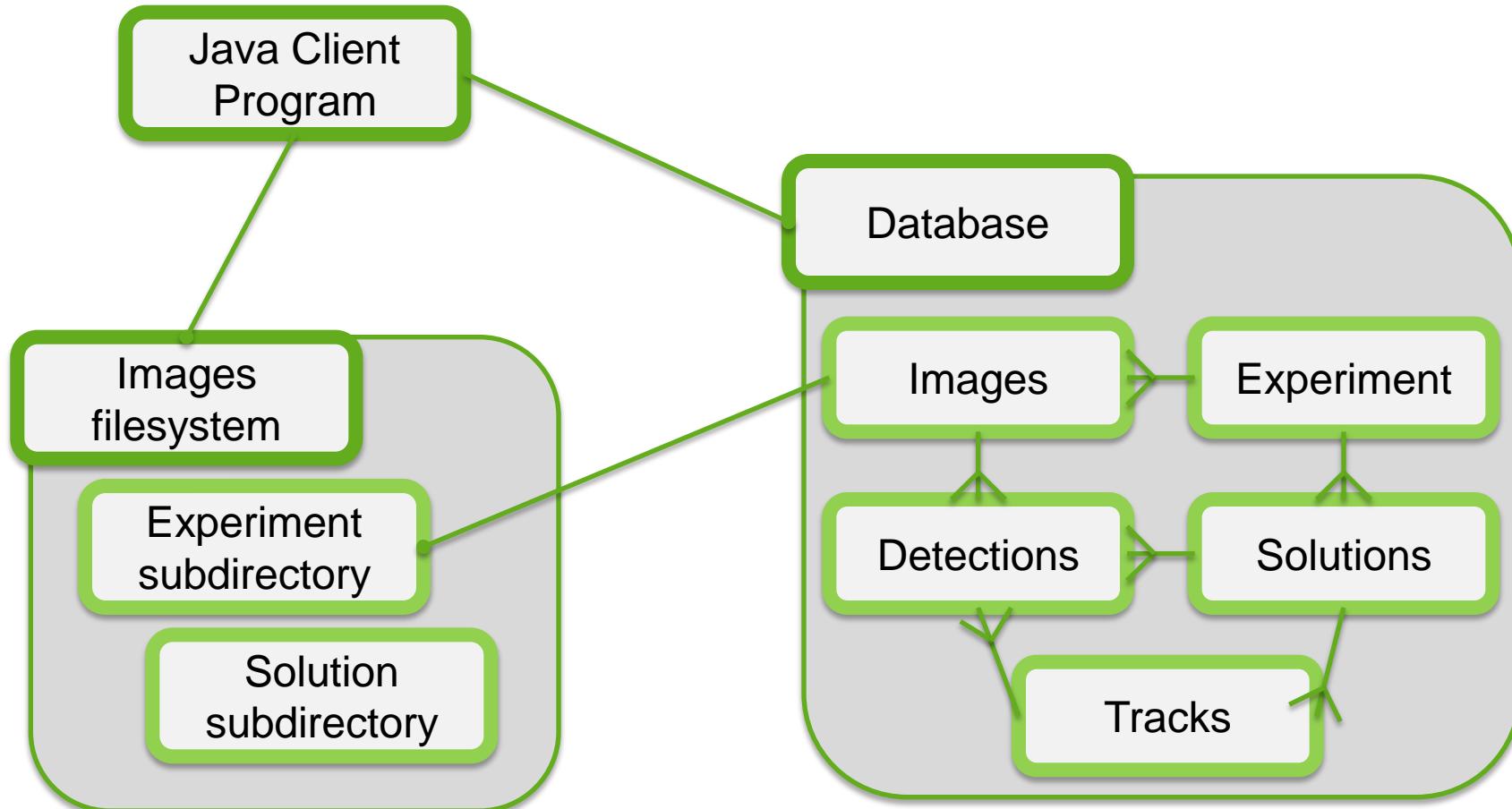
- Hexagonal microwells
- B-Lymphocytes
- ***Fluoroescence*** image

Architecture

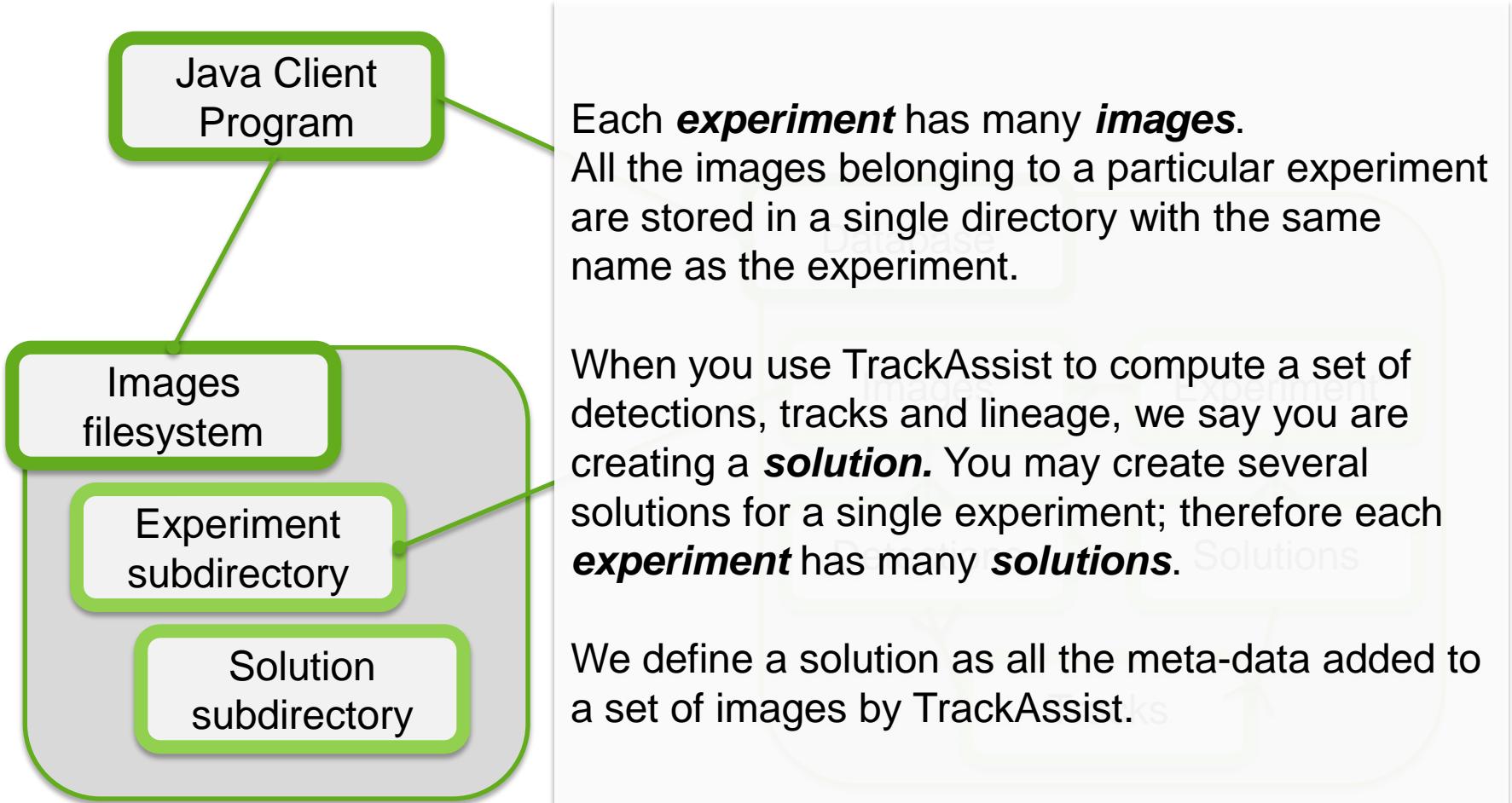


- TrackAssist has 3 components:
 - A SQL **database** that stores all meta-data about cells, detections, tracks, lineages generated by the program. All persistent information is stored here. Users are welcome to extract data directly from the database. The database is PostgreSQL by default, but any SQL RDBMS can be used.
 - A client **program**, written in Java, that implements the user interface and algorithms in TrackAssist. Users only need to use the client program directly.
 - A managed **filesystem**. Typical time-lapse biomedical imagery datasets may be several GB in size per experiment. To store such a large quantity of data, a directory on the computer is dedicated to TrackAssist. The files in here should not be modified or deleted by users.
- The client program manages both the filesystem and the database contents.

Architecture



Architecture

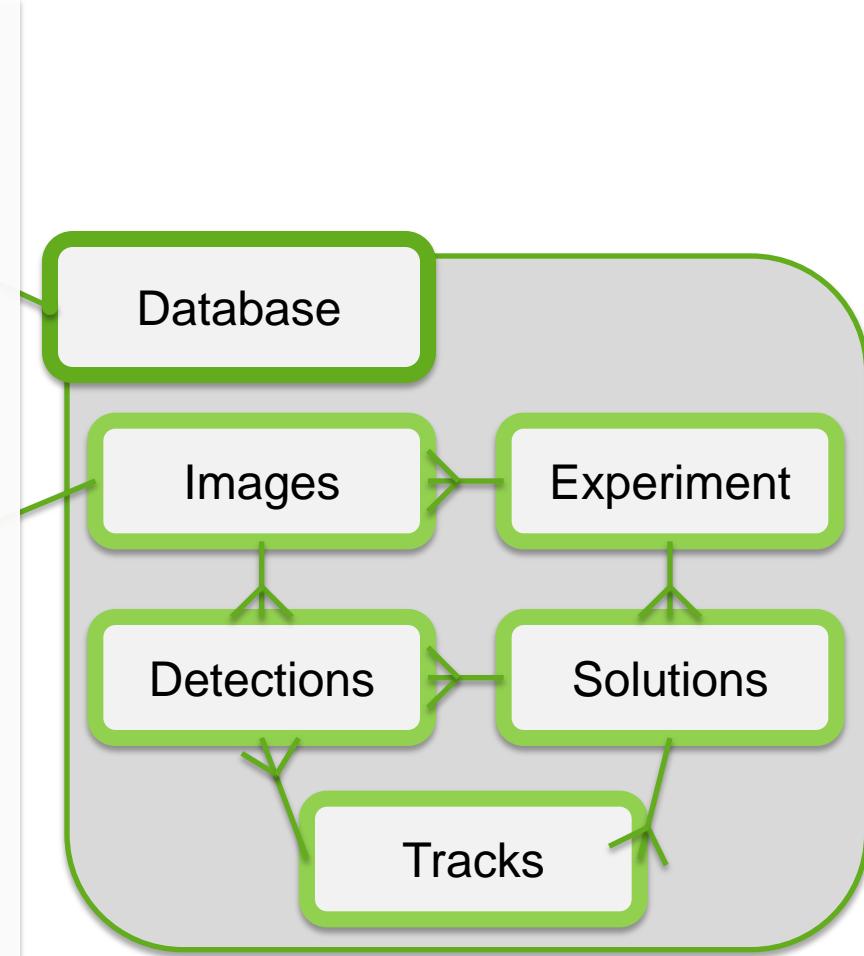


Architecture

Within the database, the same structure is reflected. Each experiment has many images; every detection is linked to the image it was found in.

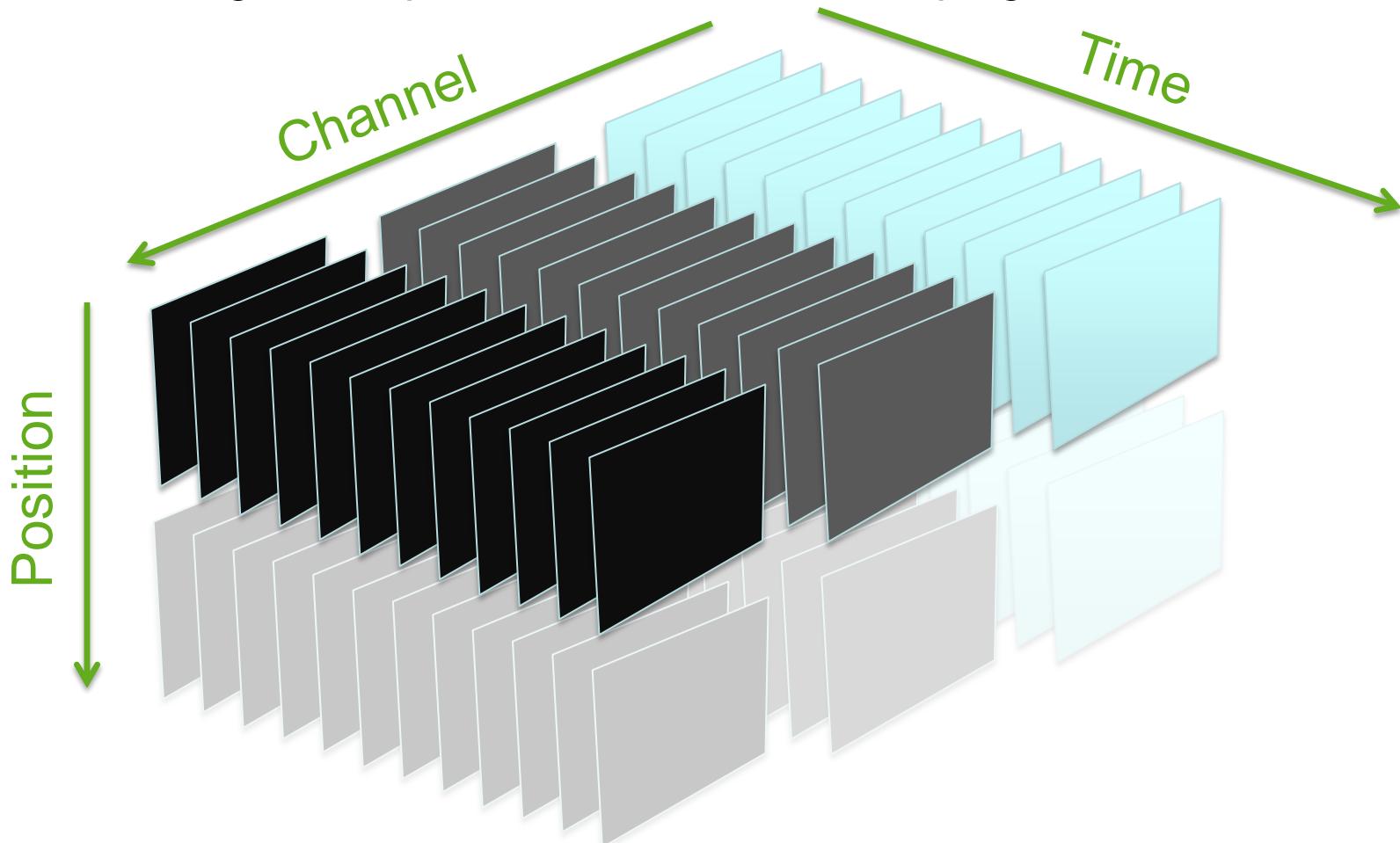
Every experiment has zero or more solutions. Each solution consists of sets of detections and tracks. A track is defined as a set of detections.

Lineages are implicitly represented in the database by detections that appear in more than one track (i.e. the final detection in a parent track is also the first detection in daughter tracks).



N-Dimensional Imagery

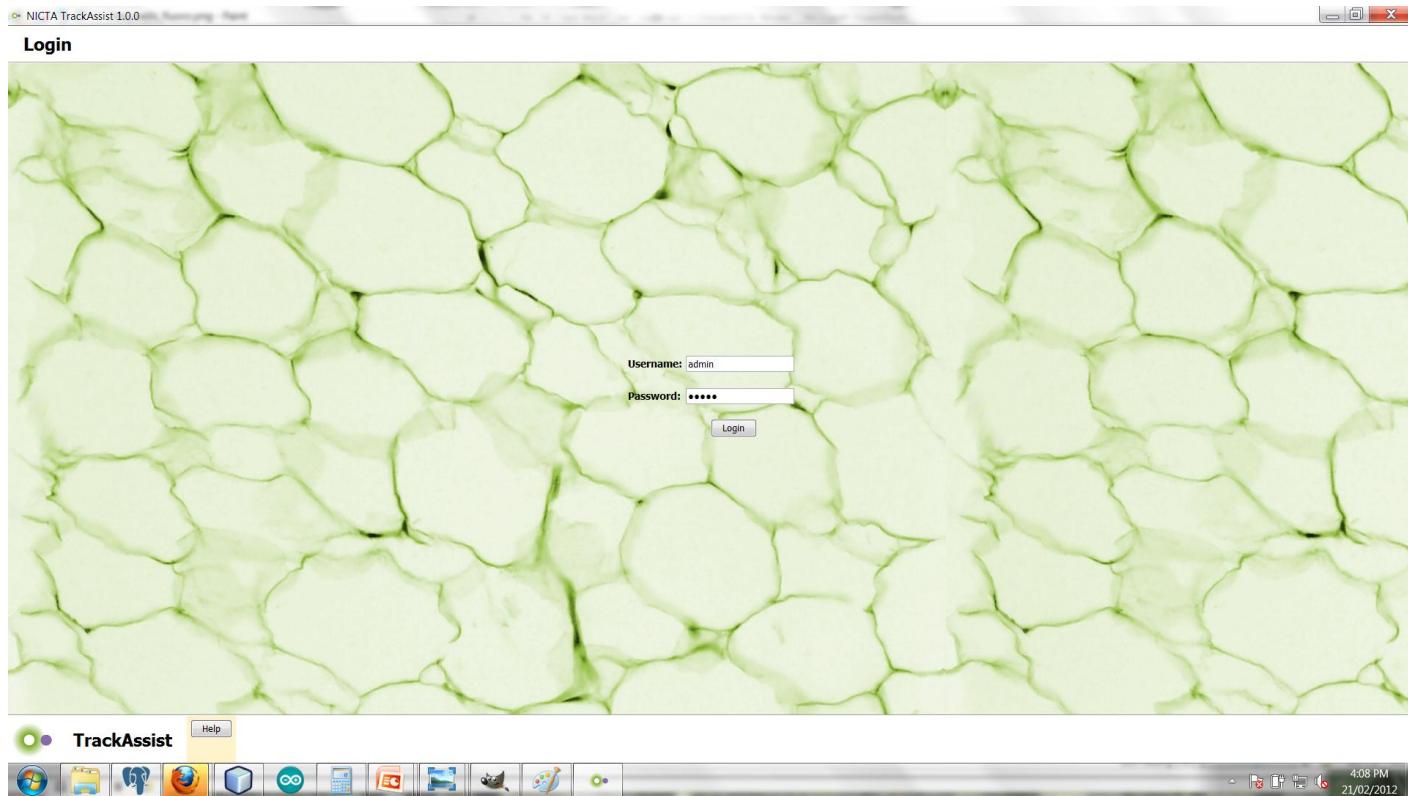
The images captured for each experiment are typically a hypercube defined by N dimensions – for example, collecting 3 channels of different images at 2 positions over 12 time steps gives 3 dimensions:



Program Walkthrough - Login



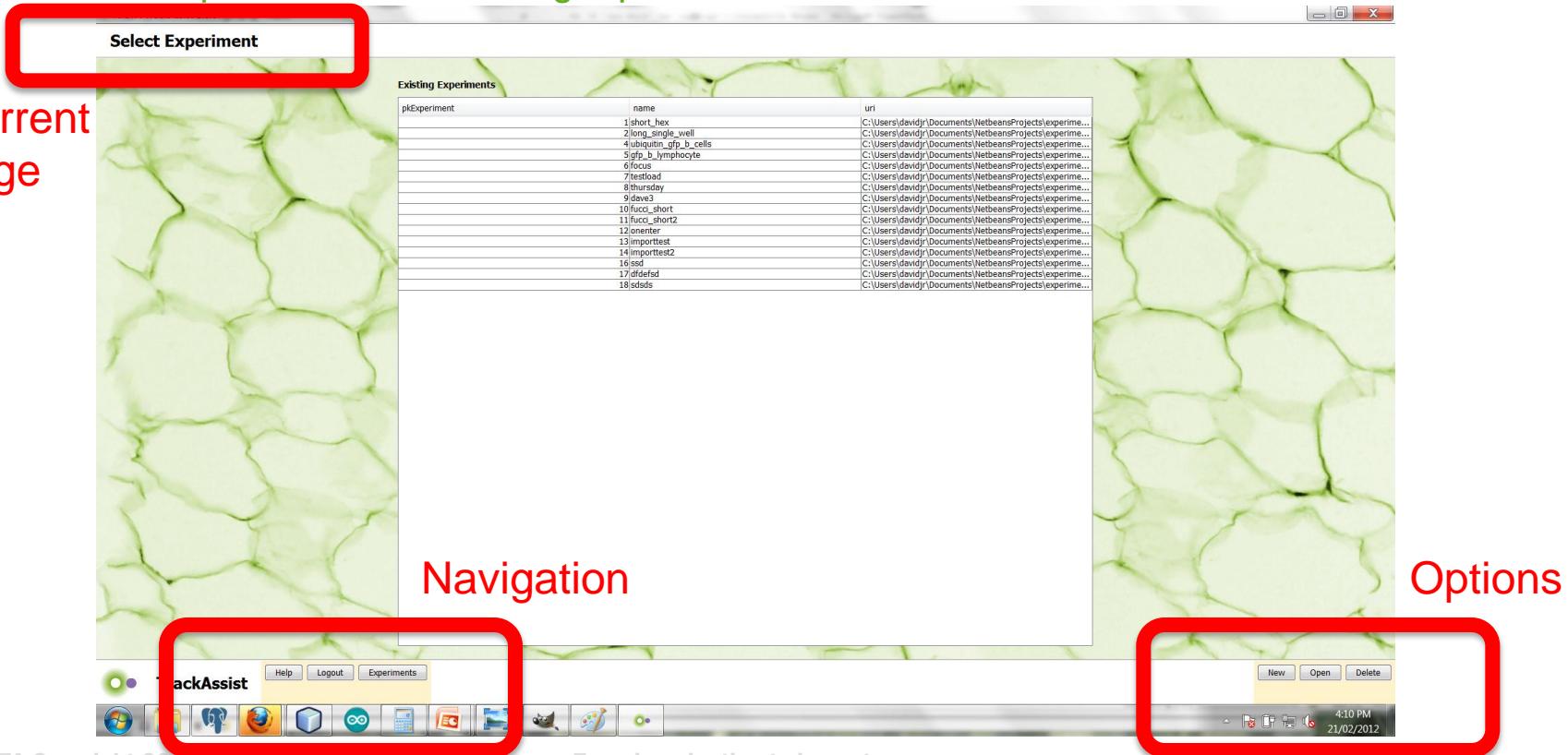
- When opened, TrackAssist will show the Login screen.
- Login to the program with your username and password. These are stored in the database.
 - The default username/password is admin/admin



Program Walkthrough - Experiments

- You will then be shown a list of existing experiments.
 - Select an existing experiment then click “Open” or click “New” to import new imaging data
 - Generally, TrackAssist shows navigation buttons in the lower-left panel, and options in the lower-right panel...

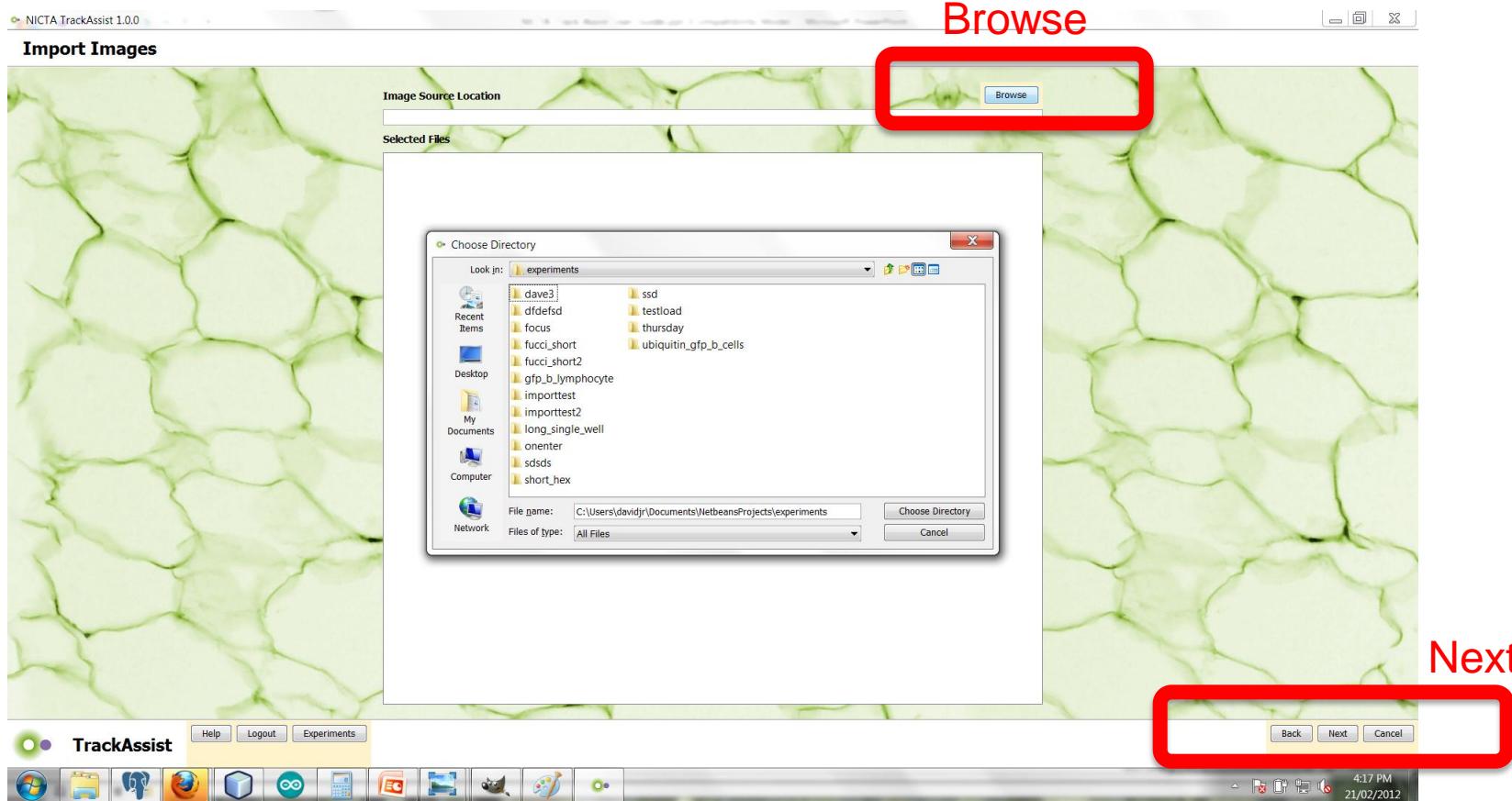
Current page



Program Walkthrough – Importing Data



- Click “browse” to select a folder containing images to import
 - These images will all be copied to TrackAssist’s managed filesystem. This may take some time if the images are many GB in size.
 - Click “Next” to continue.



Program Walkthrough – Importing Data



- You need to explain the naming of your images to TrackAssist.
 - Break down the filenames by delimiters or fixed-width fields
 - Assign all varying parts of the filenames to a coordinate then click “Next”

The screenshot shows the NICTA TrackAssist 1.0.0 software interface. The main window title is "Parse Filenames".

Example File Name: 20090201_ubiquitin-0005_d3_8_p000001t0000001z001c01.tif

File Name Breakdown:

Name Part	Width/Delimiter	Fixed Width?
20090201		
ubiquitin	-p	
0005		
d3		
8		
p000001	t	
0000001	z	
001c	c	
01		
.tif		

Coordinates: A red box highlights this section.

Coordinate	Coordinate Type	Parsed Result
N/A		20090201_ubiquitin-0005_d3_8
x	Number	1
time	Number	1
z	Number	1
channel	Number	1
N/A		.tif

Buttons: "Reset", "Next", "Back", "Cancel". A red box highlights the "Next" button.

Program Walkthrough – Importing Data



- For example, we have extracted 4 coordinates from our images' filenames: **x** (position), **time**, **z** (position), and **channel**.
 - These are all numeric fields

The screenshot shows the NICTA TrackAssist 1.0.0 software interface. The main window title is "Parse Filenames". On the left, there is a "File Name Breakdown" table where file parts are mapped to width/delimiter. On the right, there is a "Coordinates" table showing parsed results. A red box highlights the "Coordinates" table, and the word "Coordinates" is written in red above it. Another red box highlights the "Next" button at the bottom right of the window, and the word "Next" is written in red to its right.

Name Part	Width/Delimiter	Fixed Width?
20090201_ubiquitin-0005_d3_8_p	_p	<input type="checkbox"/>
000001t	t	<input type="checkbox"/>
00000001z	z	<input type="checkbox"/>
001c	c	<input type="checkbox"/>
01	.	<input checked="" type="checkbox"/>
tif		<input type="checkbox"/>

Coordinates	Coordinate Type	Parsed Result
N/A		20090201_ubiquitin-0005_d3_8
x	Number	1
time	Number	1
z	Number	1
channel	Number	1
N/A		tif

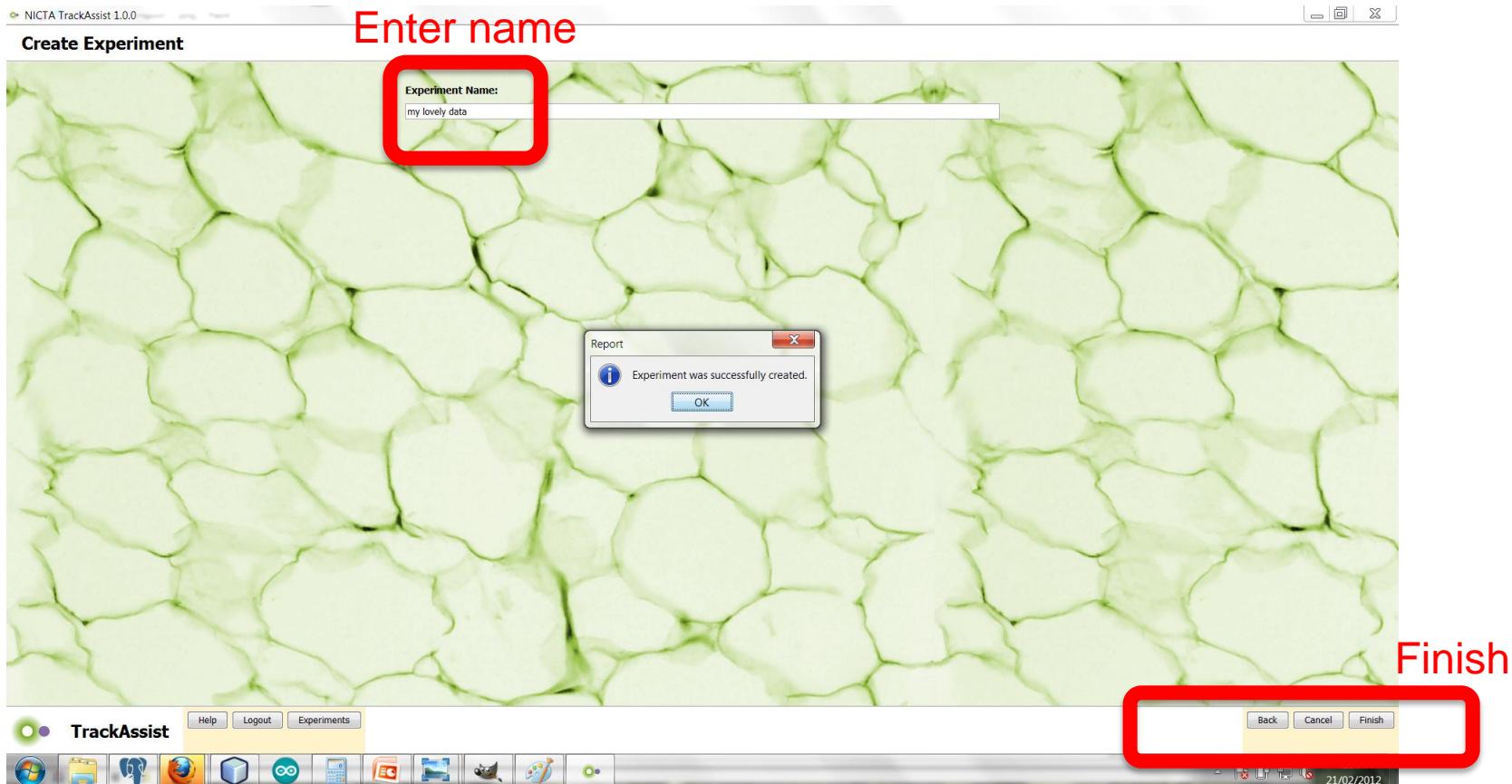
Coordinates

Next

Program Walkthrough – Importing Data

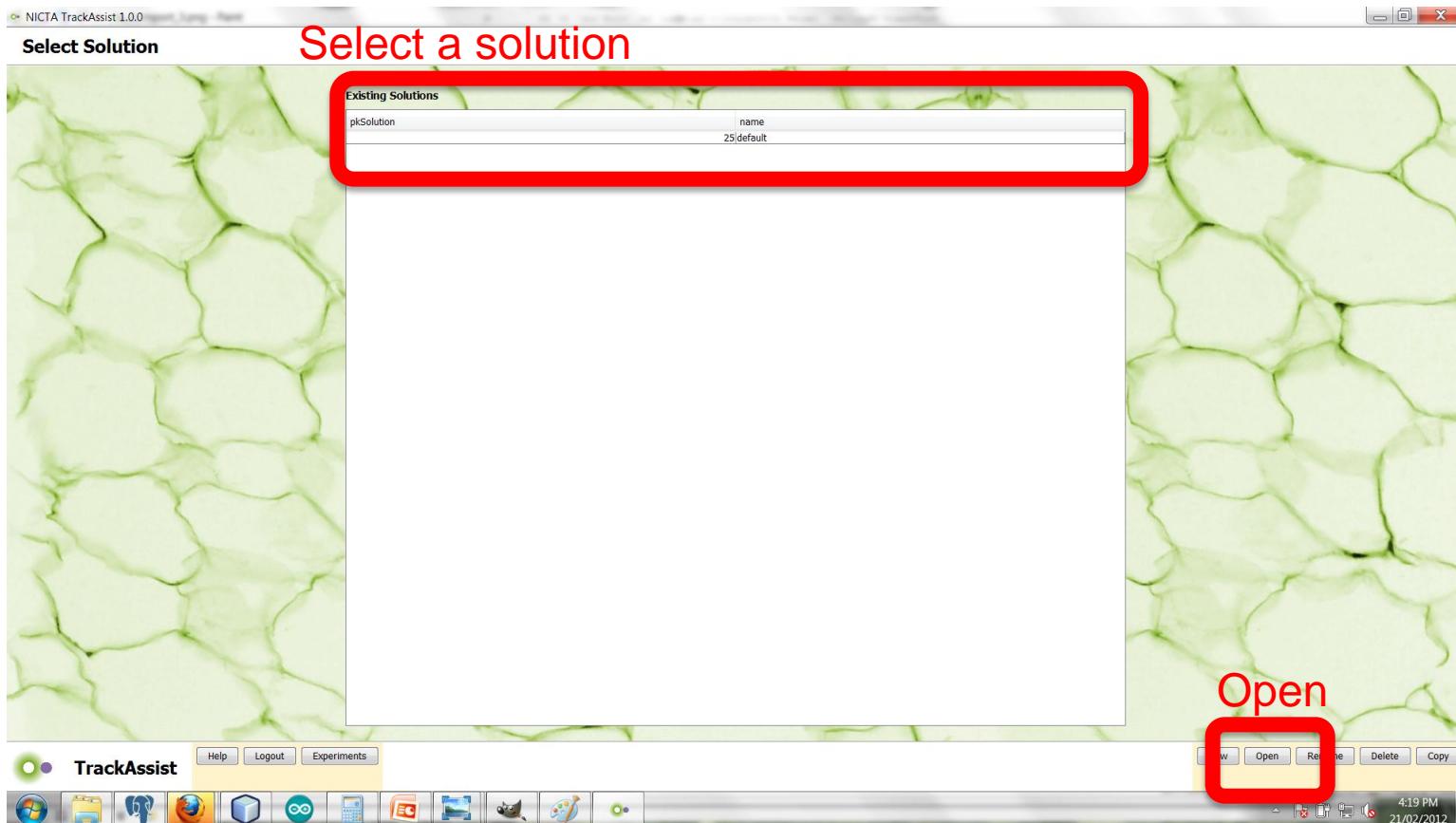


- Enter a name for your new experiment and click “Finish”
 - The name should not contain characters that are not permitted on your filesystem.



Program Walkthrough – Solutions

- After either importing data, or opening an experiment, you need to choose a solution to work on.
 - A default solution is created in all experiments. Double-click or click “Open”



Program Walkthrough – Solutions

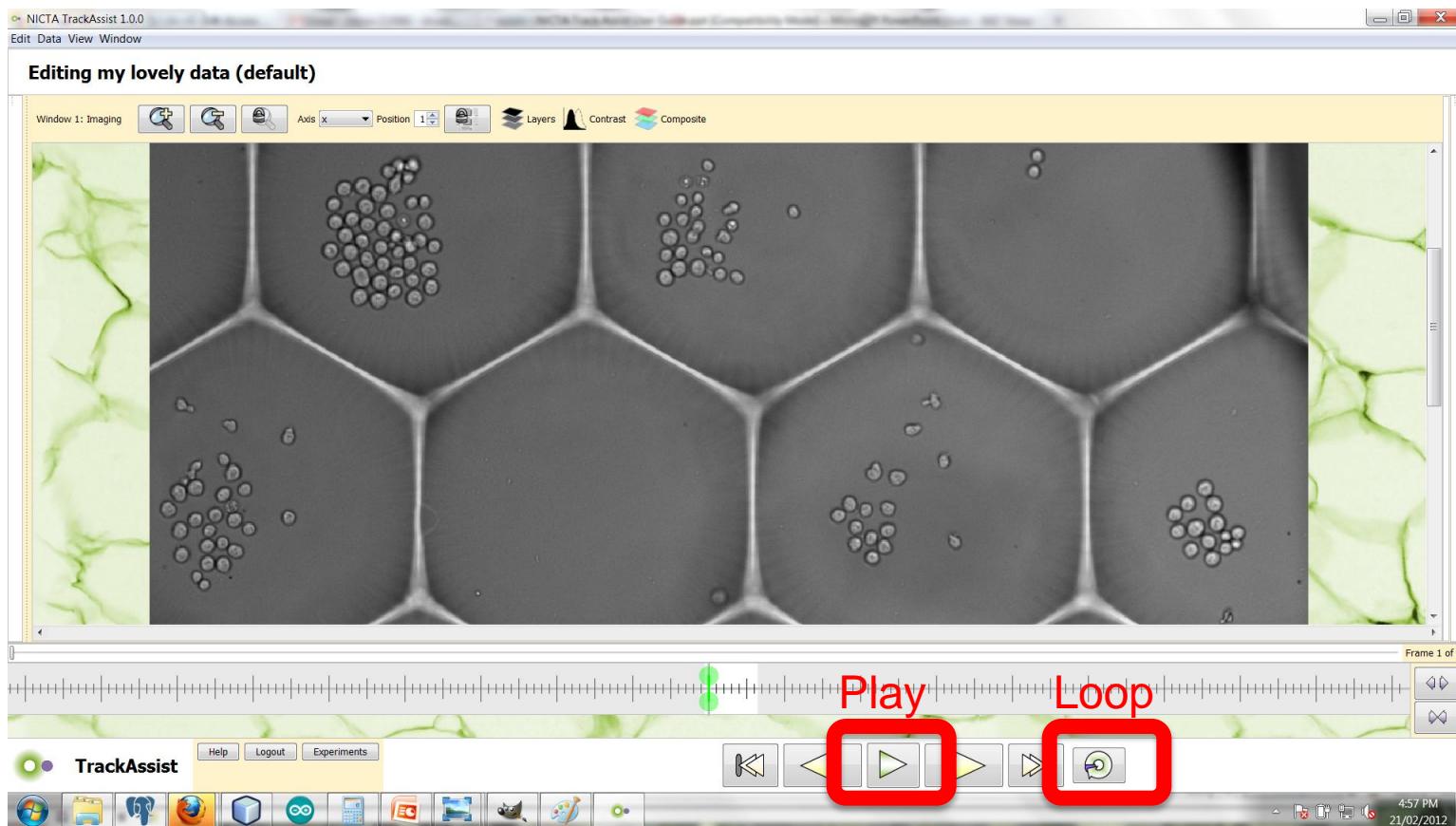


- Note that when opening experiments and solutions, TrackAssist preloads a lot of data including images, detections, and tracks. It also precomputes models based on this data, such as lineage. If your data contains thousands of images totalling many GB, this will take some time. Once loaded, the program is more responsive due to this caching.

Program Walkthrough – Imaging View



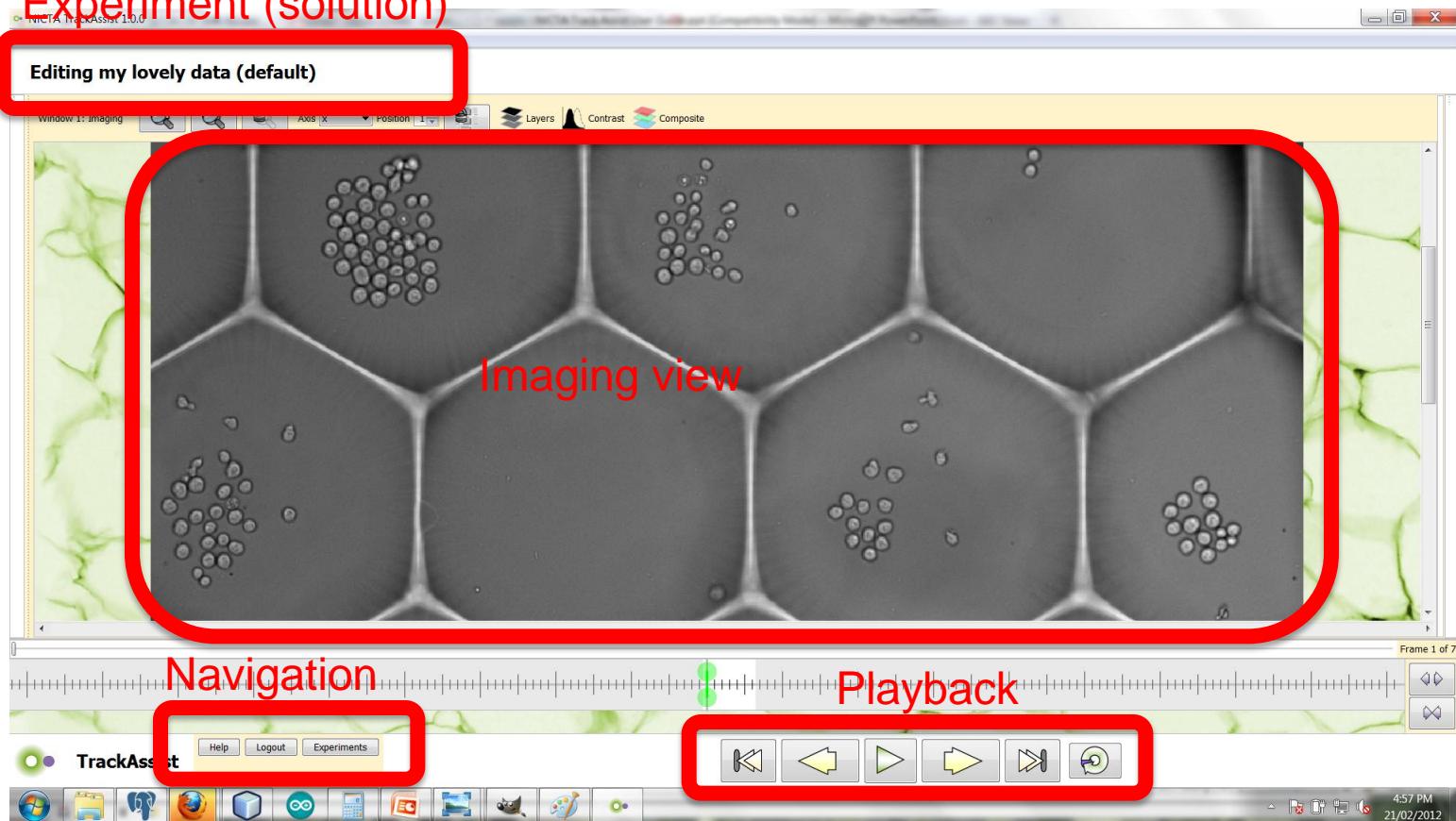
- When a solution is opened, a single imaging view appears.
 - Click the “Play” button or scroll the mouse wheel to advance through time. The images will be shown like a video. Click “Loop” to make the video loop.



Program Walkthrough – Navigation

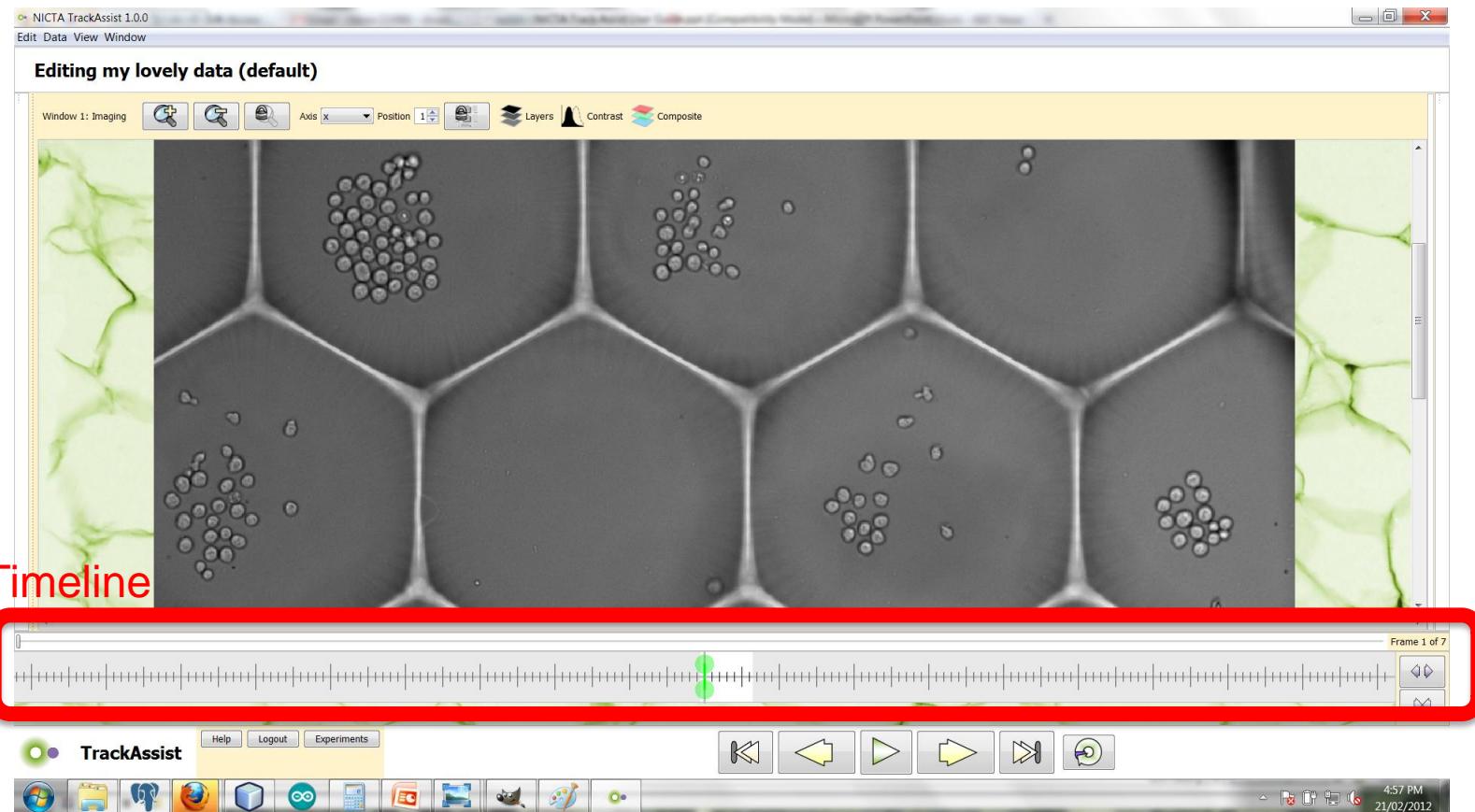
- The title bar shows the current experiment and solution
 - Navigation and Playback controls allow you to navigate to other datasets or within this experiment/solution.

Experiment (solution)



Program Walkthrough – Timeline

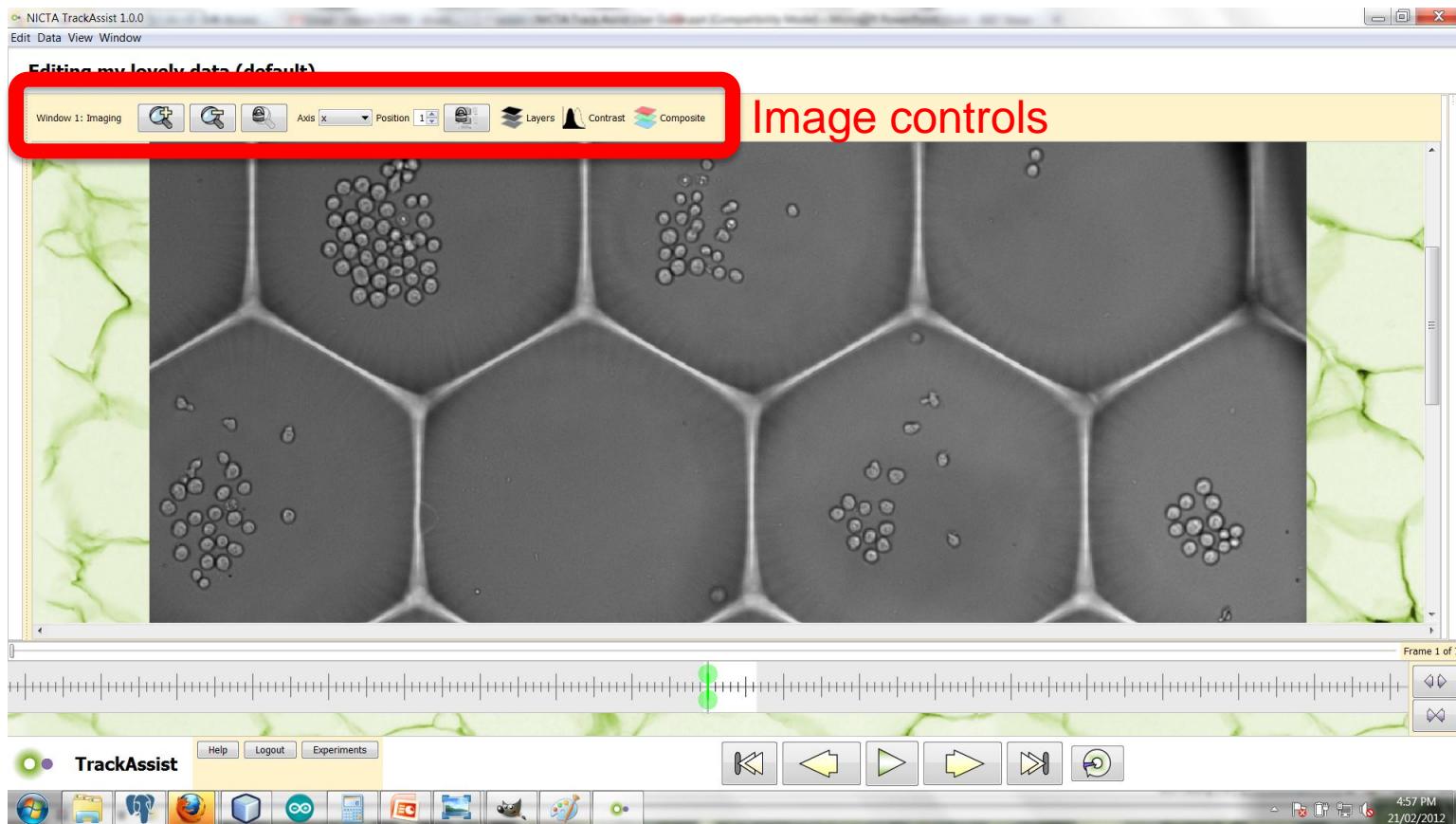
- The timeline panel allows gross and precise time control
 - The white area represents the valid image sequence. Grey areas have no images.
 - The green dots indicate the current time (displayed image).



Program Walkthrough – Image Controls



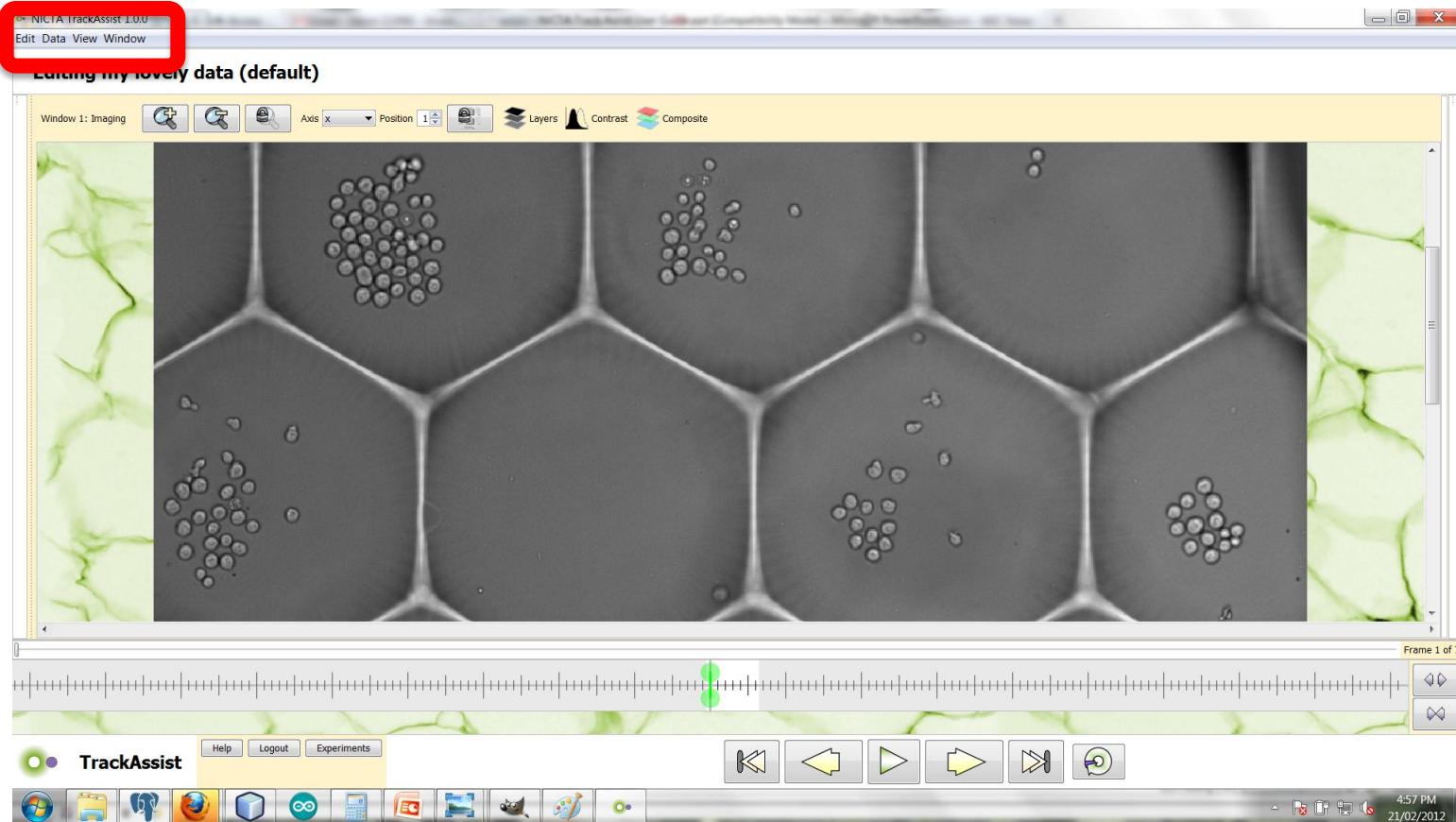
- Each image view can be controlled separately
 - The image can be panned and zoomed.
 - Related images can be displayed, e.g. other channels



Program Walkthrough – Menu

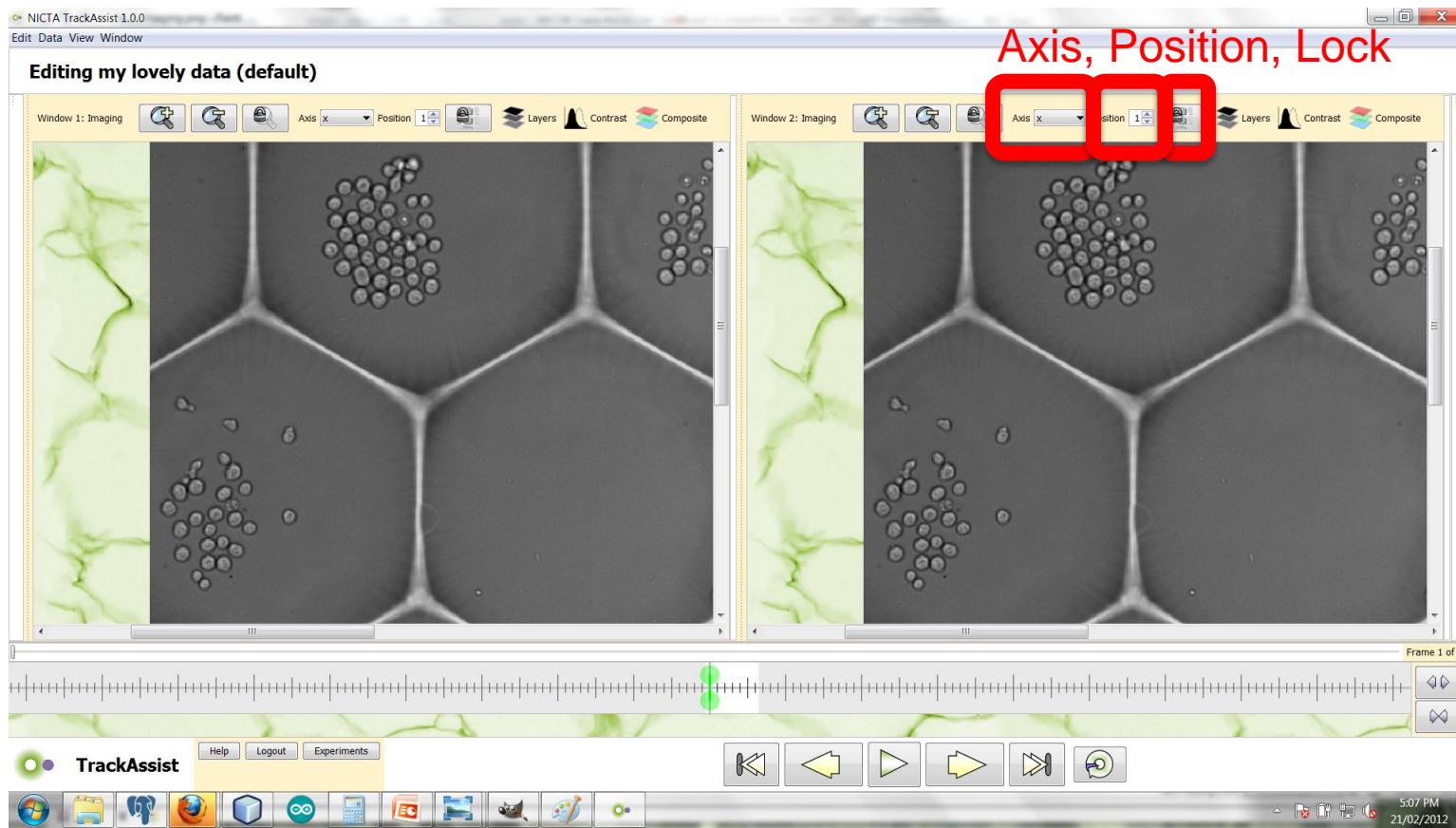
- The views and data options are controlled by a menu
 - Select Window → Add → Imaging Window to add another window.

Menus



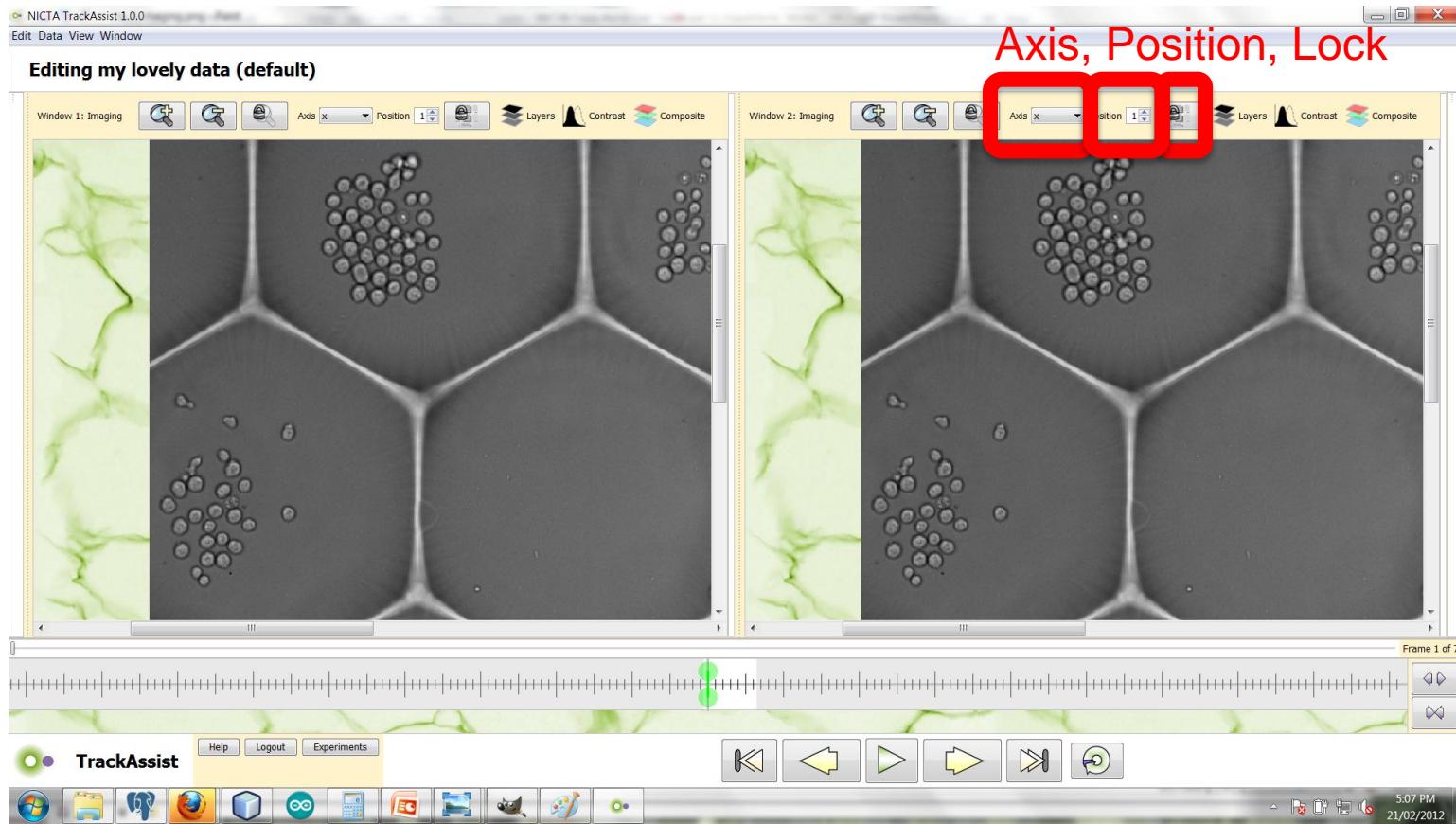
Program Walkthrough – Views

- You now have 2 views. In one of them:
 - Change the “Axis” combo-box to e.g. Channel and click “Lock”.
 - Then change the position to a different value. Two related images are now shown.



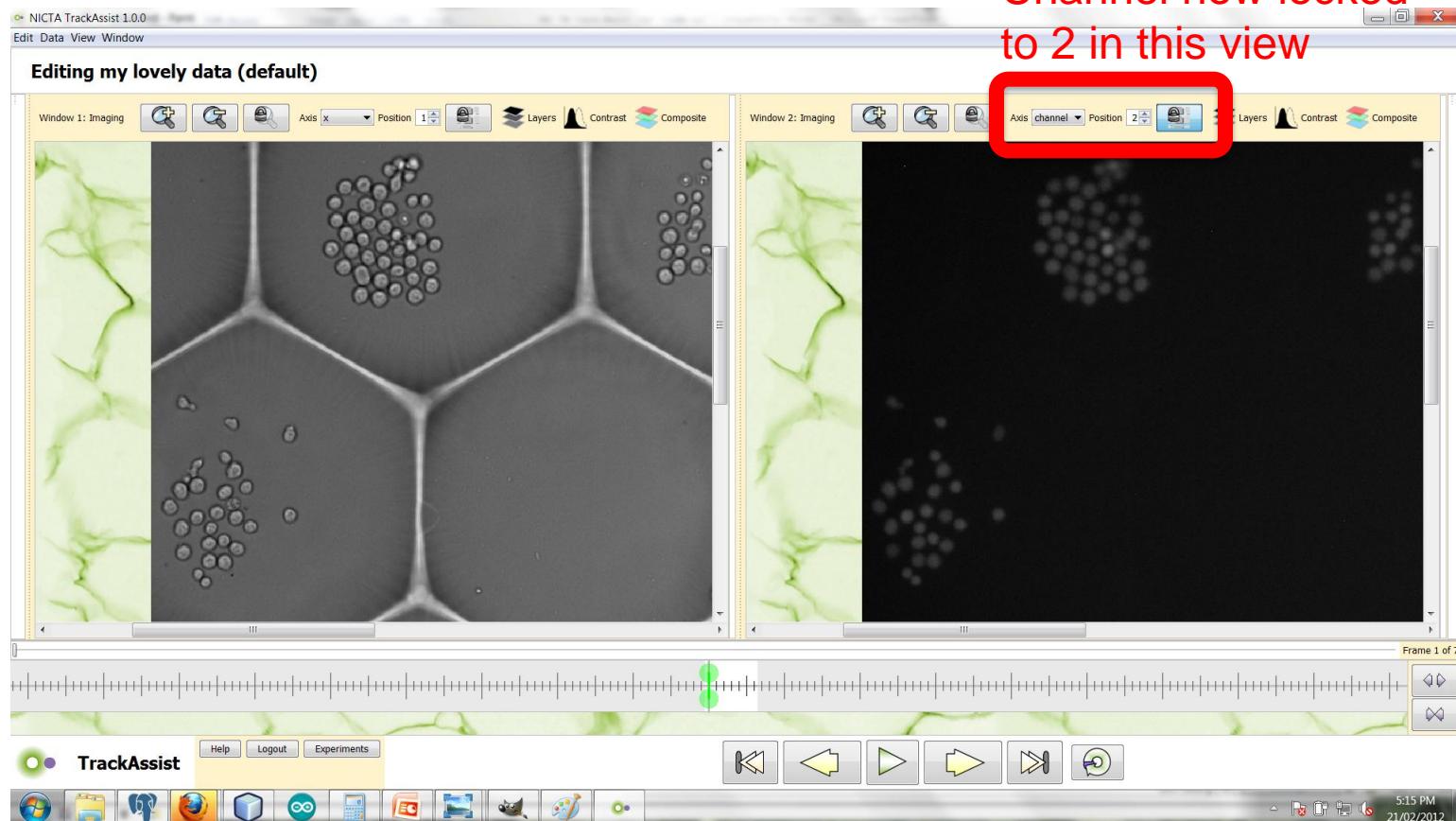
Program Walkthrough – Views

- You now have 2 views. In one of them:
 - Change the “Axis” combo-box to e.g. Channel and click “Lock”.
 - Then change the position to a different value. Two related images are now shown.



Program Walkthrough – Views

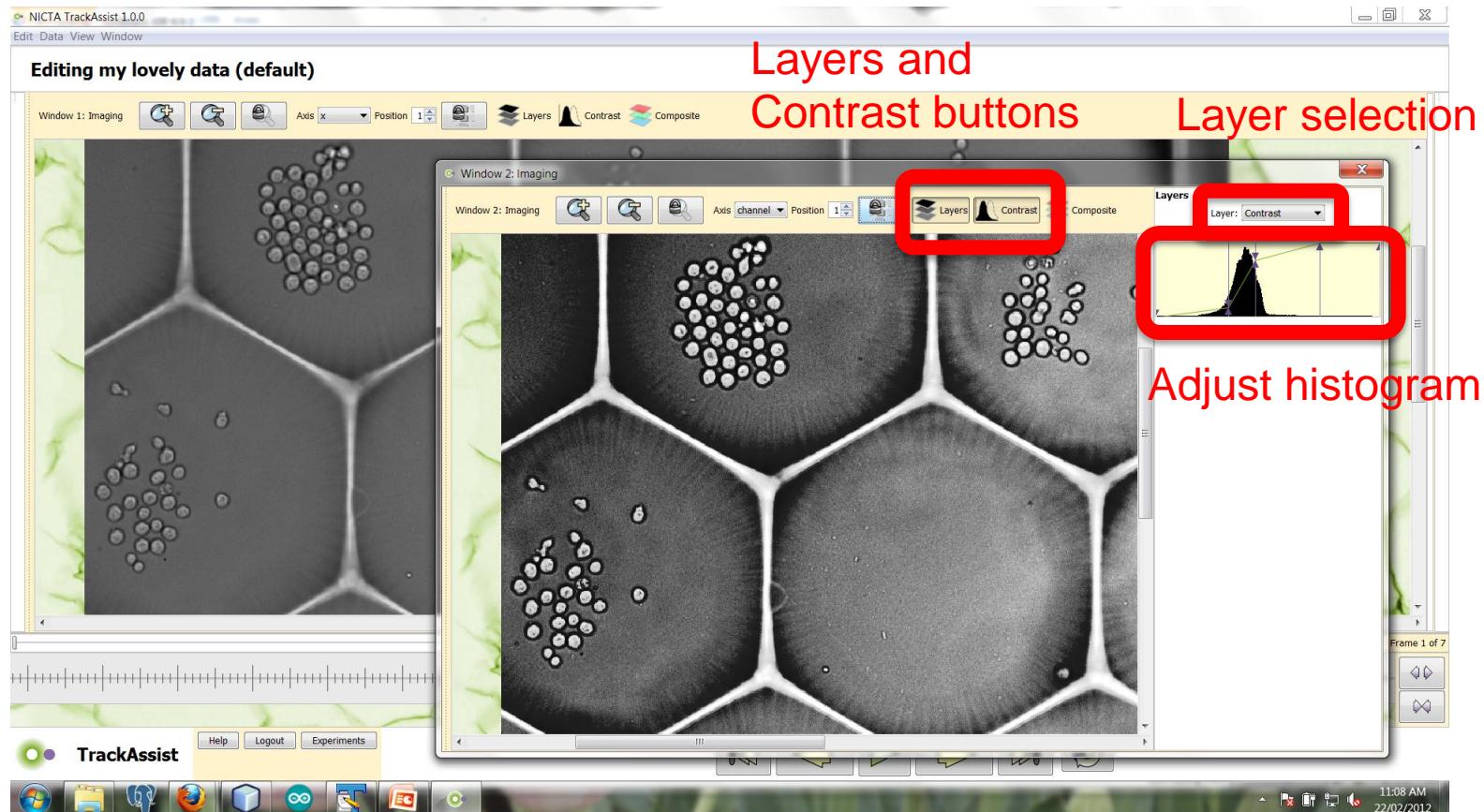
- One of the views is now a different channel.
 - The views are still locked together in other coordinates, pan and zoom.
 - Try moving around the images, and through time.



Program Walkthrough – Contrast



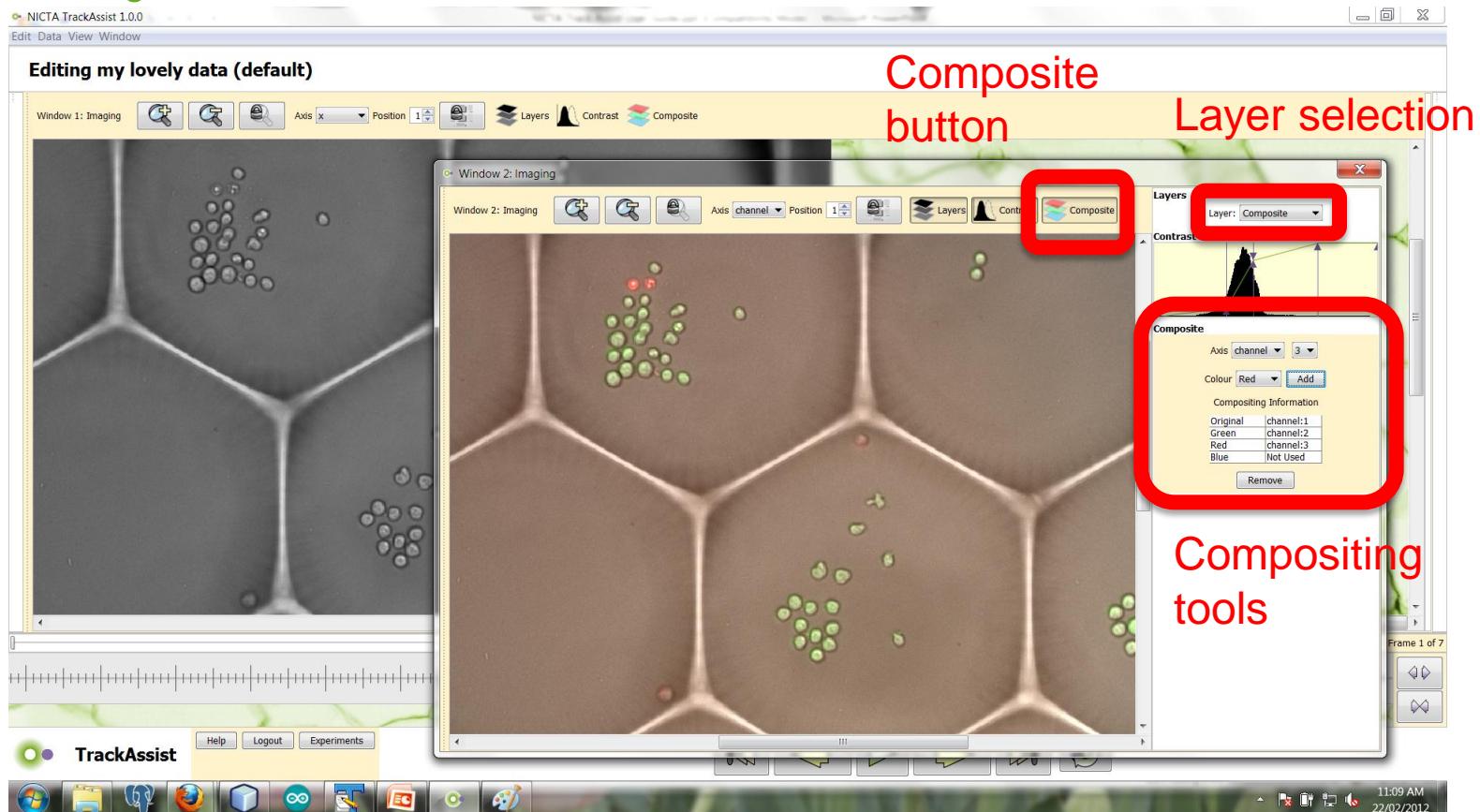
- You can enhance the contrast of the images.
 - Click the “Layers” and “Contrast” buttons then select the “Contrast” layer. Adjust the points on the histogram to modify the contrast of the displayed image.



Program Walkthrough – Compositing



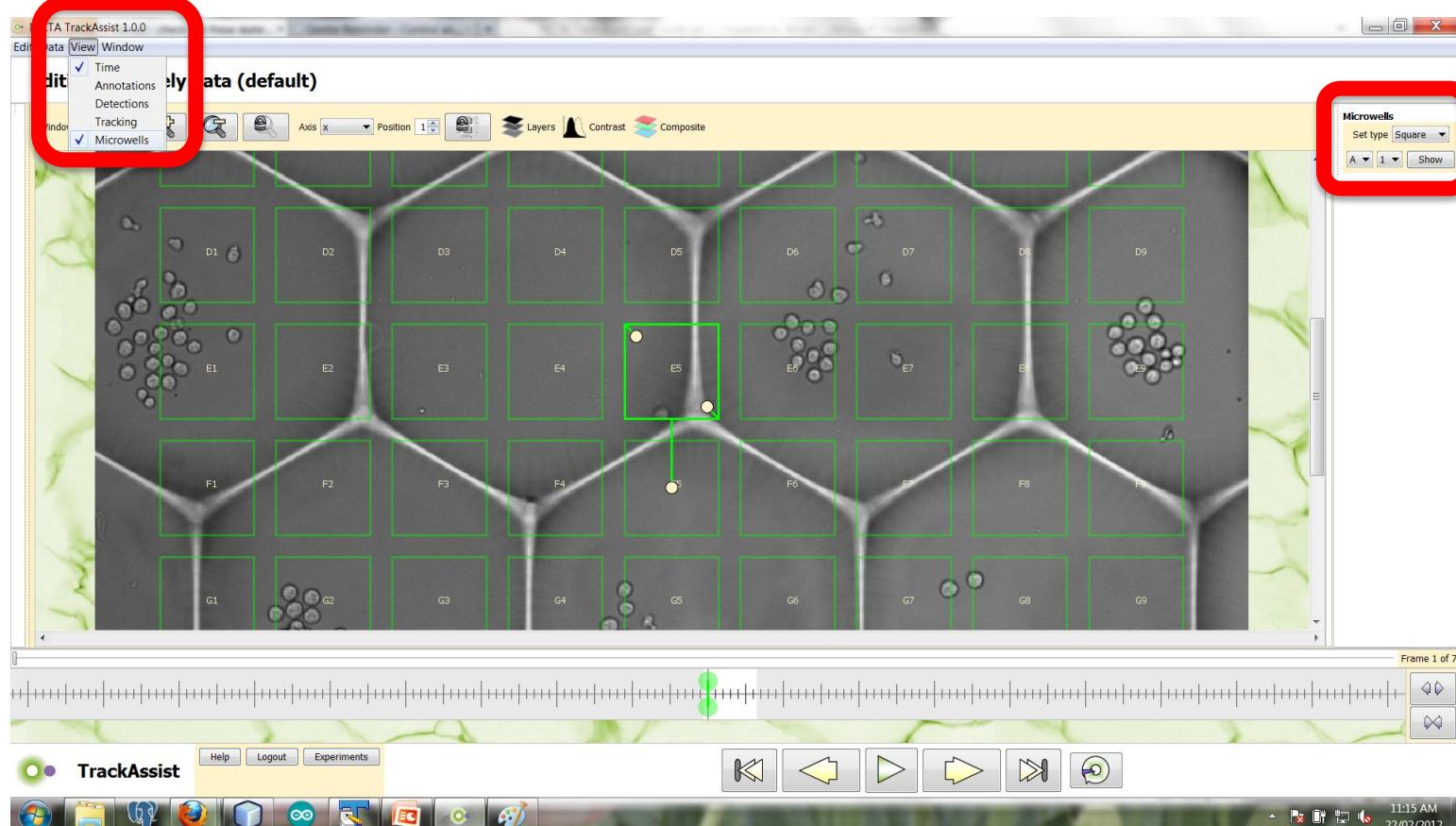
- It is also possible to “composite” multiple channels together.
 - Click “Composite” and then select axis and value (e.g. Channel). Choose a colour and click “Add”. Here we display 2 fluorescent channels on top of the transmission image, as Red and Green.



Program Walkthrough – Microwells

- You can define regions within the images using microwells.
 - From the menu select View→Microwells. Some tools appear on the right.

Menu

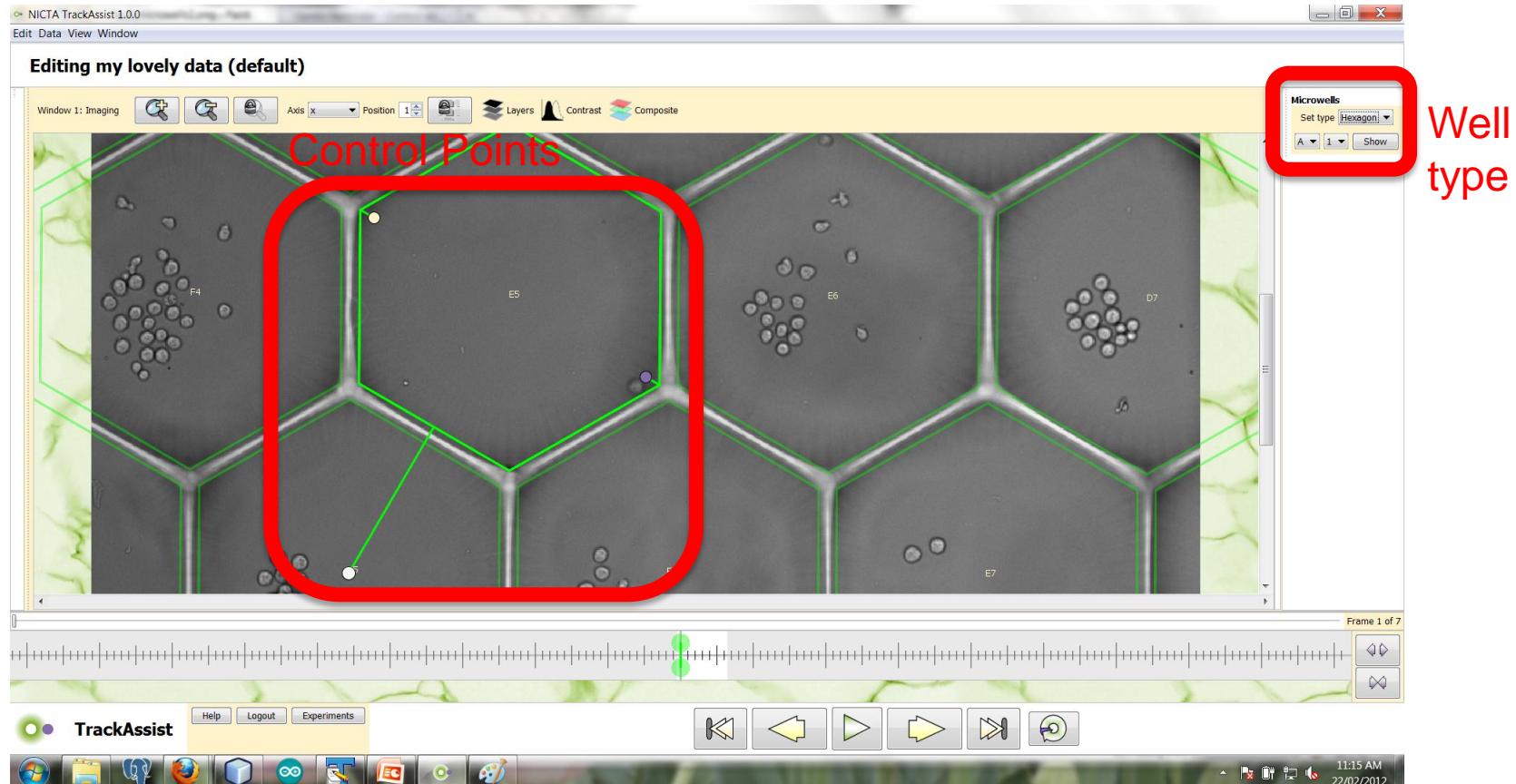


Tools

Program Walkthrough – Microwells



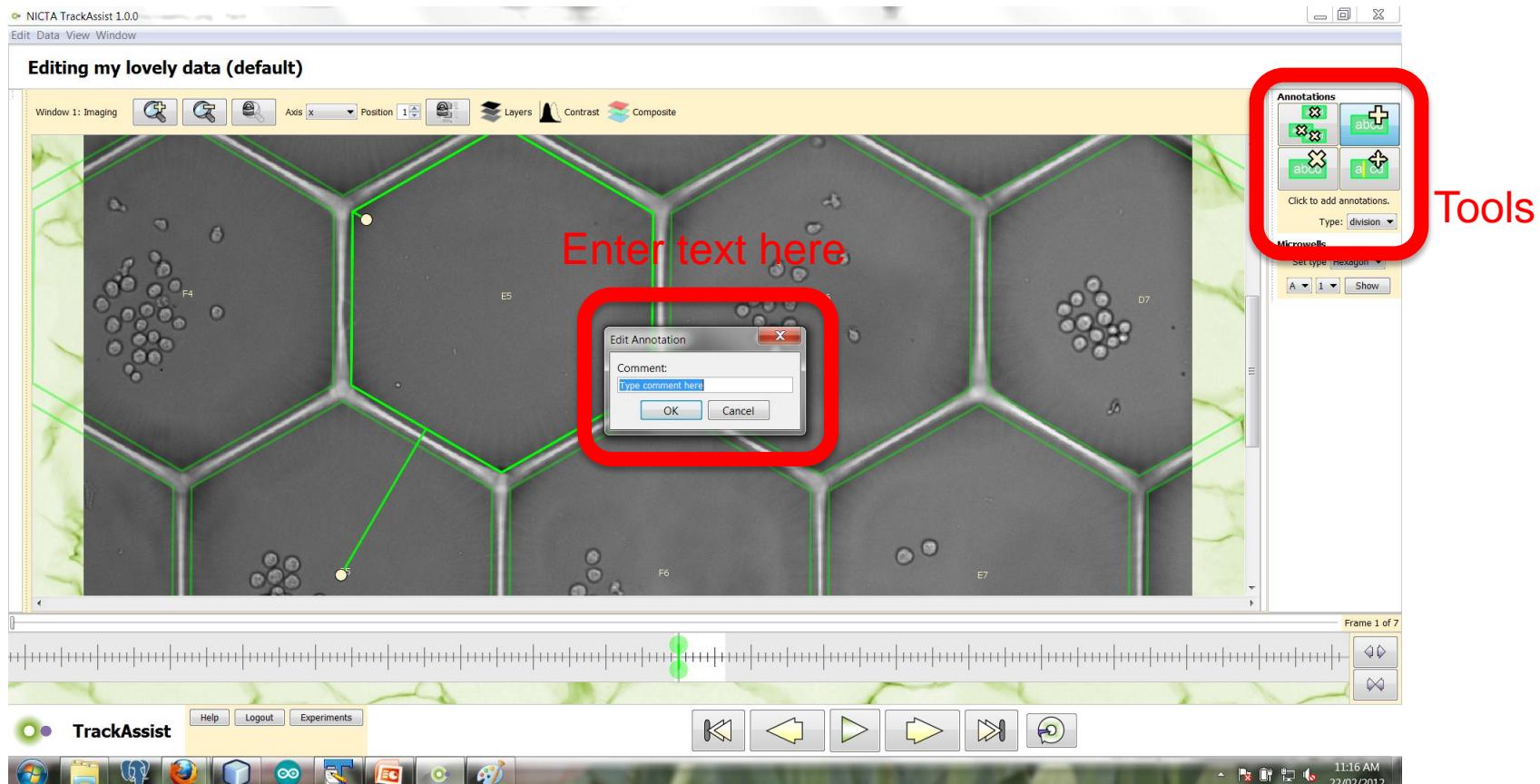
- You can change the type of microwell and move them to fit
 - Currently only hexagonal and square wells are supported.
 - Drag the 3 control points to adjust the microwells



Program Walkthrough – Annotations



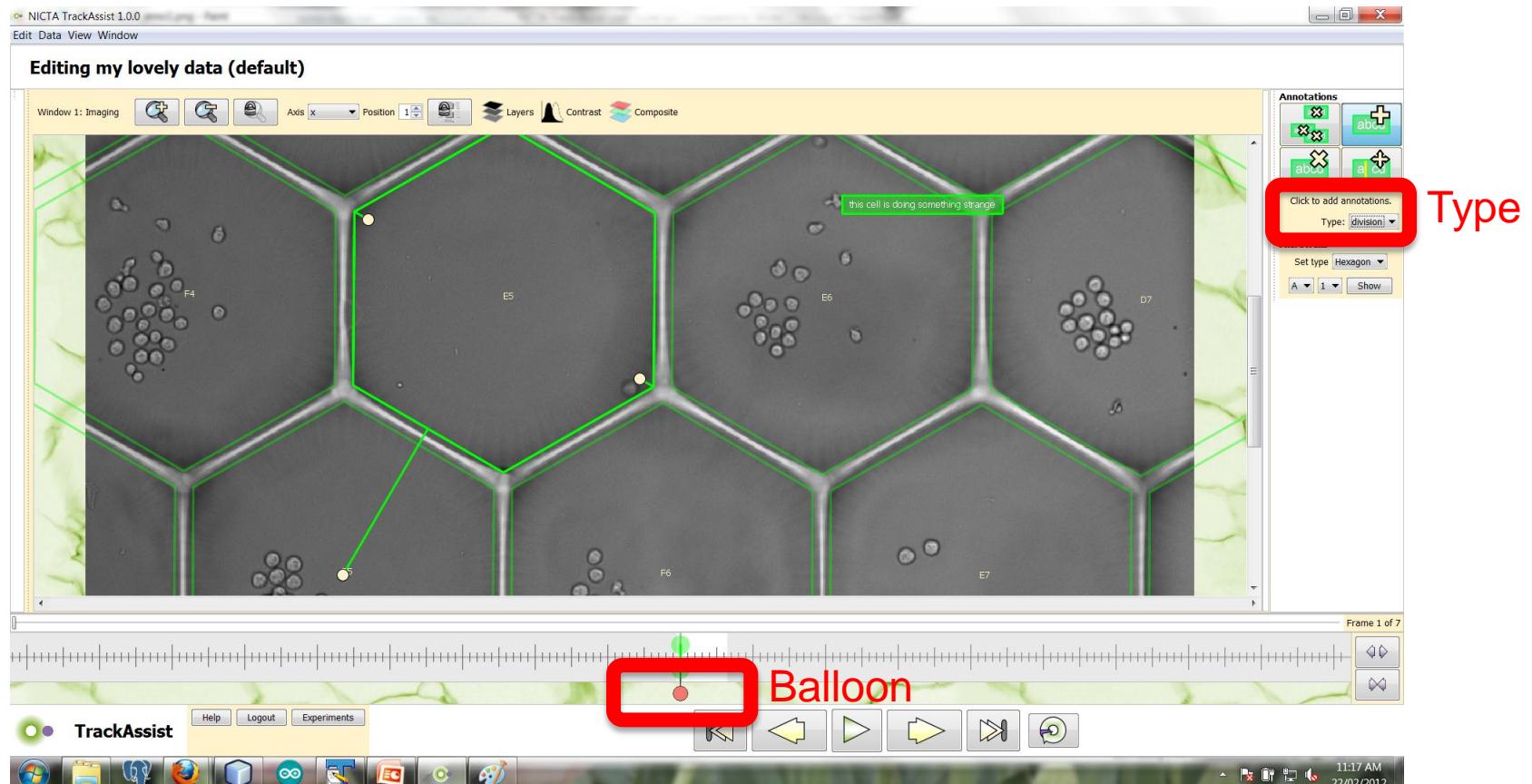
- You can put comments – annotations – anywhere in your images.
 - From the Menu select View→Annotations. Select the “Create” tool (+) and click in the image to add annotations at this position. Annotations can be moved, edited and deleted.



Program Walkthrough – Annotations



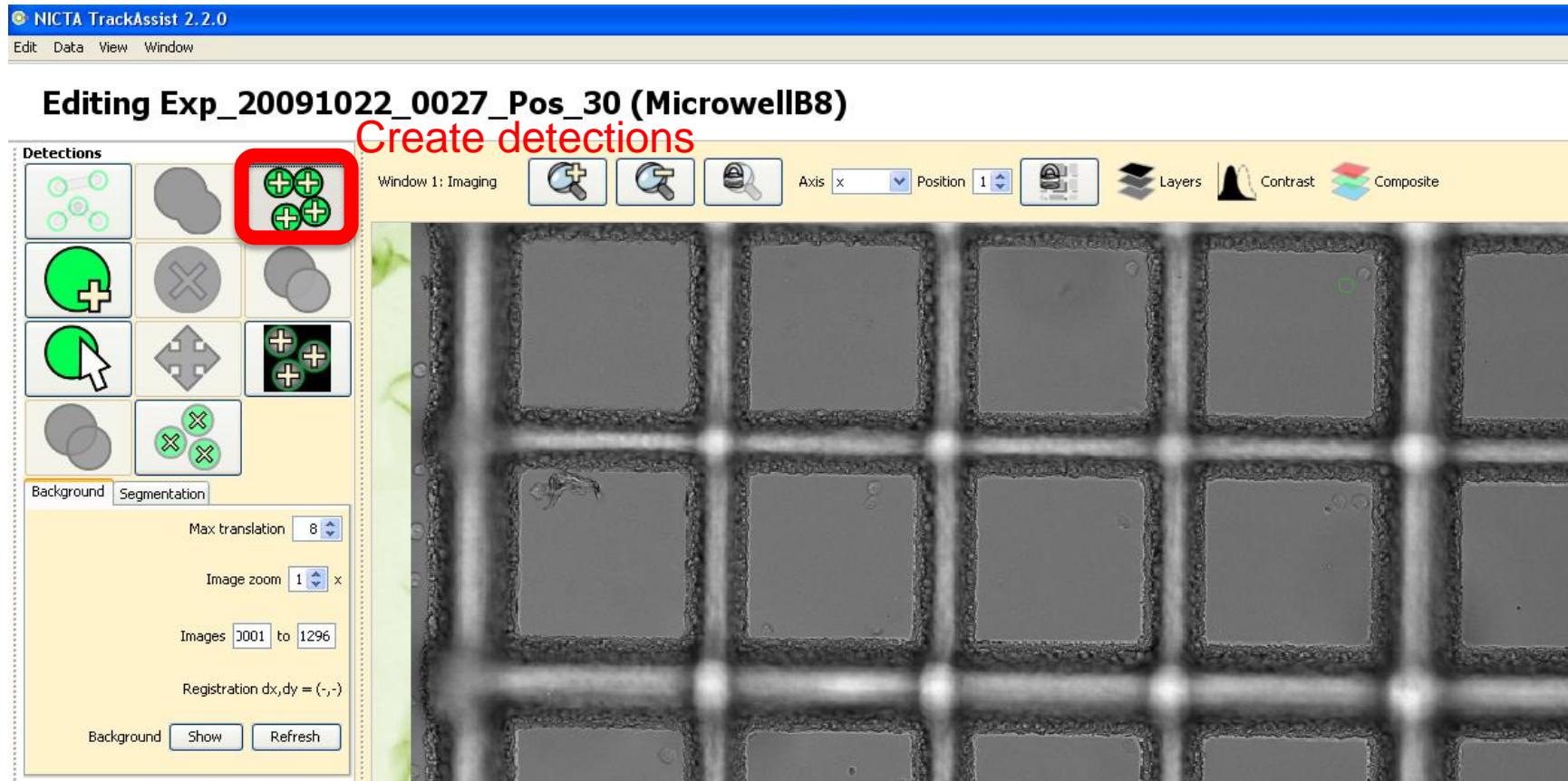
- Annotations appear on the timeline.
 - Click the coloured balloons to show the annotation[s].
 - Colour indicates annotation type.



Program Walkthrough – Detections



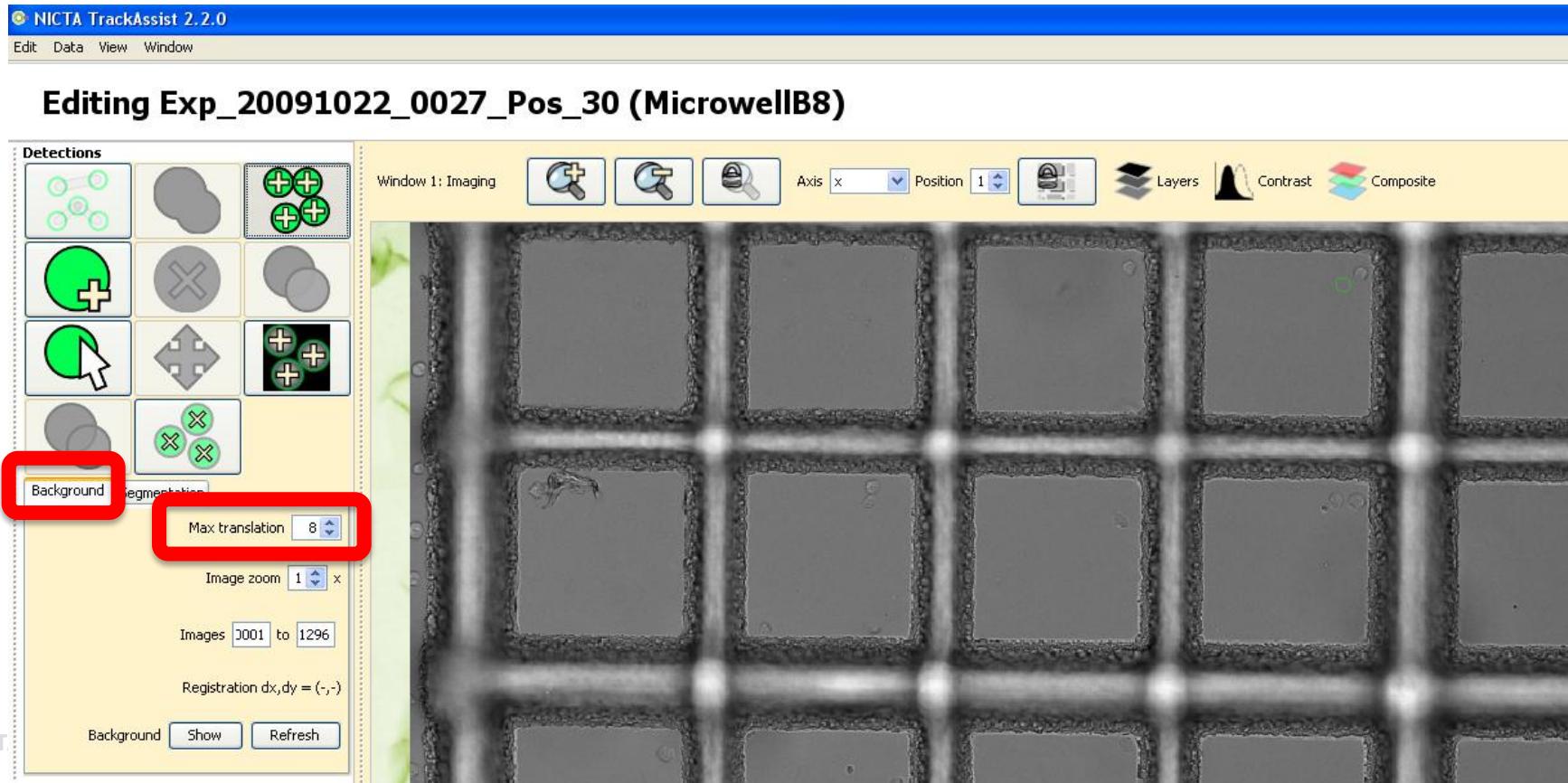
- We will first explain automation cell segmentation by motion detection.
 - From the Menu select View→Detections
 - Click the “Create multiple detections” tool. The tool’s controls will appear.



Program Walkthrough – Detections



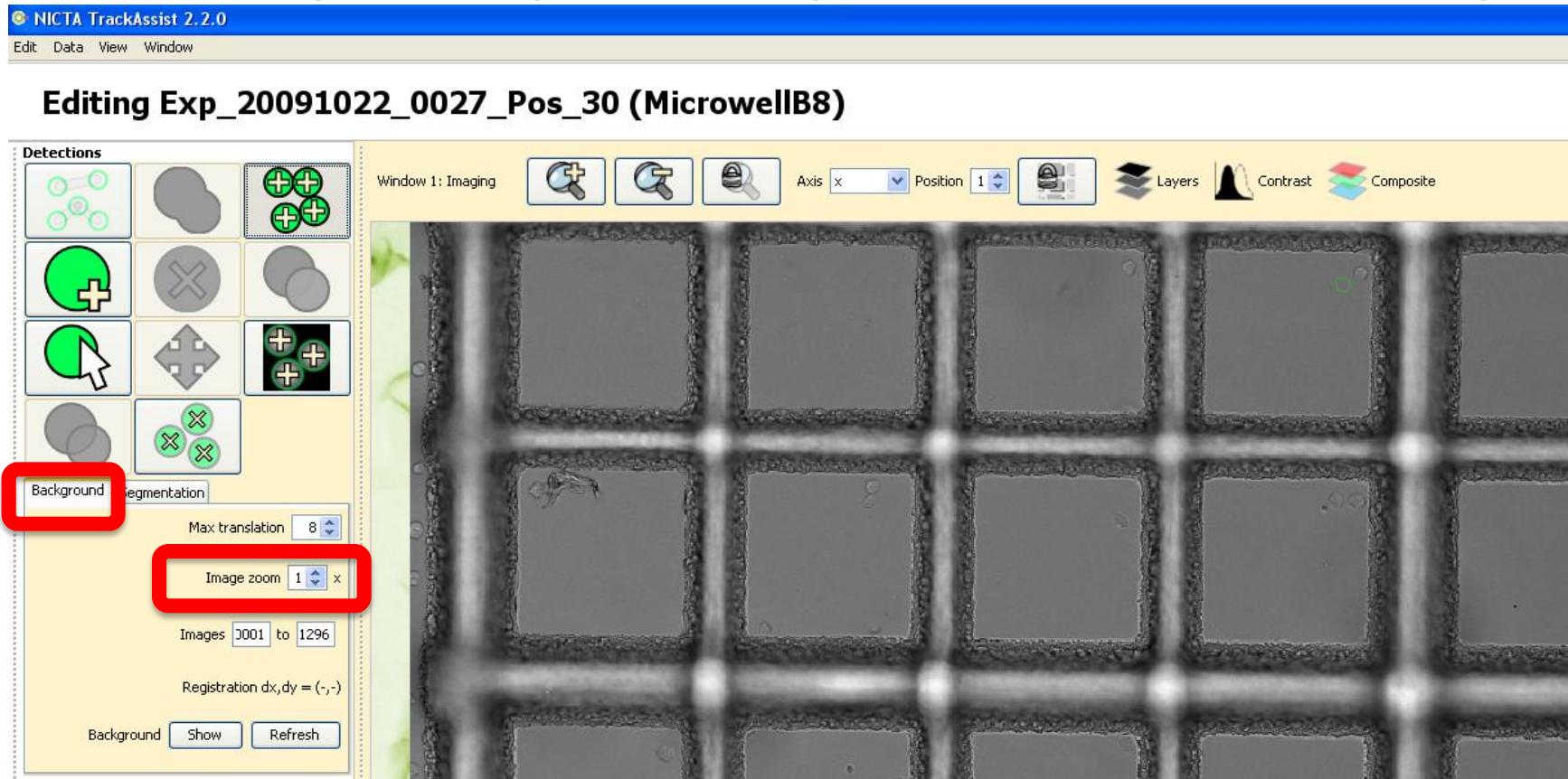
- We need to build a model of the appearance of the image background
 - There are several parameters for creation of the background model.
 - Max translation : Max. translation (pixels) of images when trying to align them to the reference (background) image. This caters for play in the microscope platform. Larger values mean that background modelling takes a lot longer; typically 1-4 is OK



Program Walkthrough – Detections

- Registration at Super-Resolution

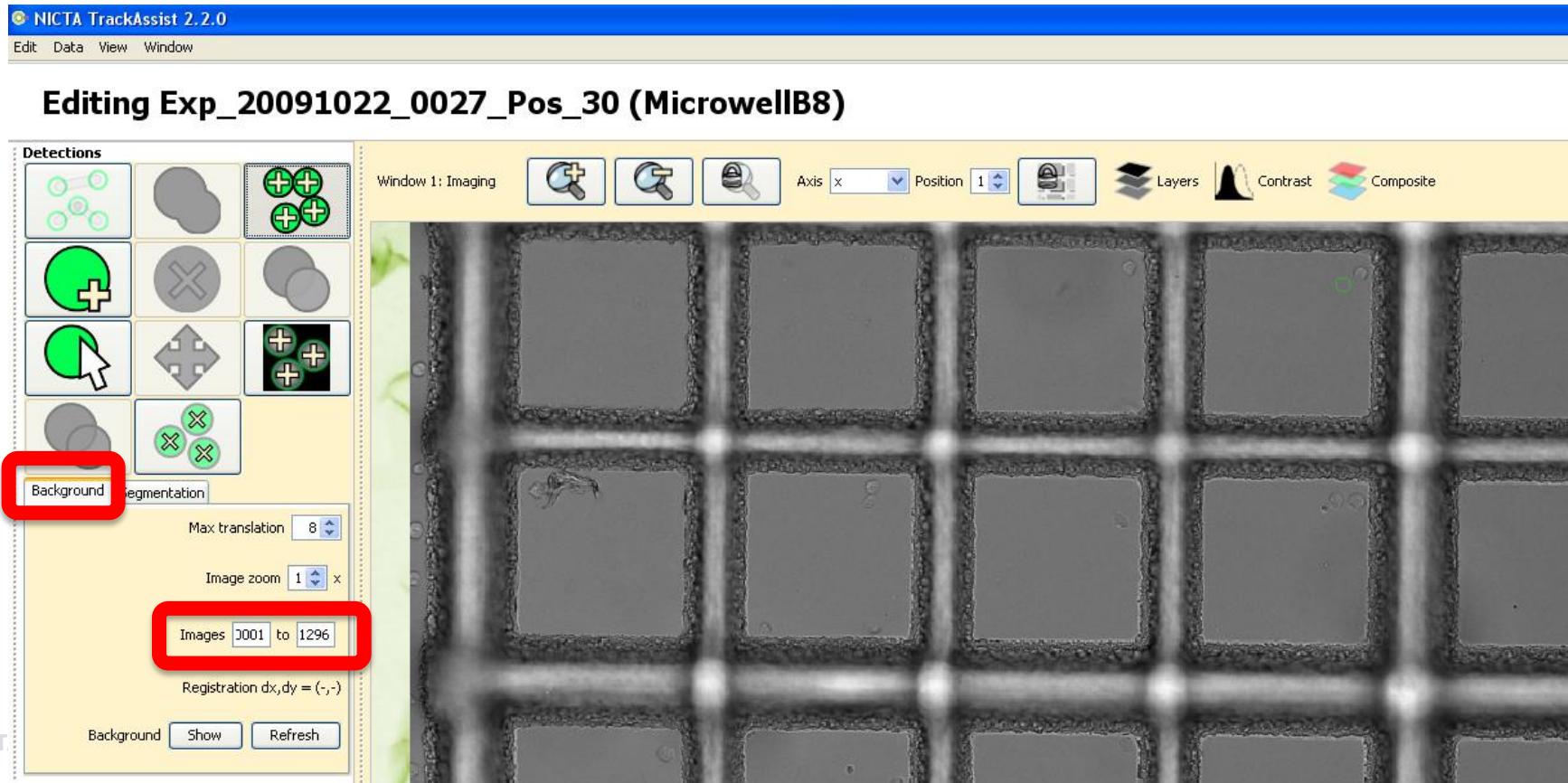
- Image Zoom: Sometimes the ideal alignment of the image is a translation of less than 1 pixel. We allow this by allowing the image to be enlarged (to e.g. 2x size) prior to finding the best alignment. Values greater than 1x therefore slow modelling.



Program Walkthrough – Detections



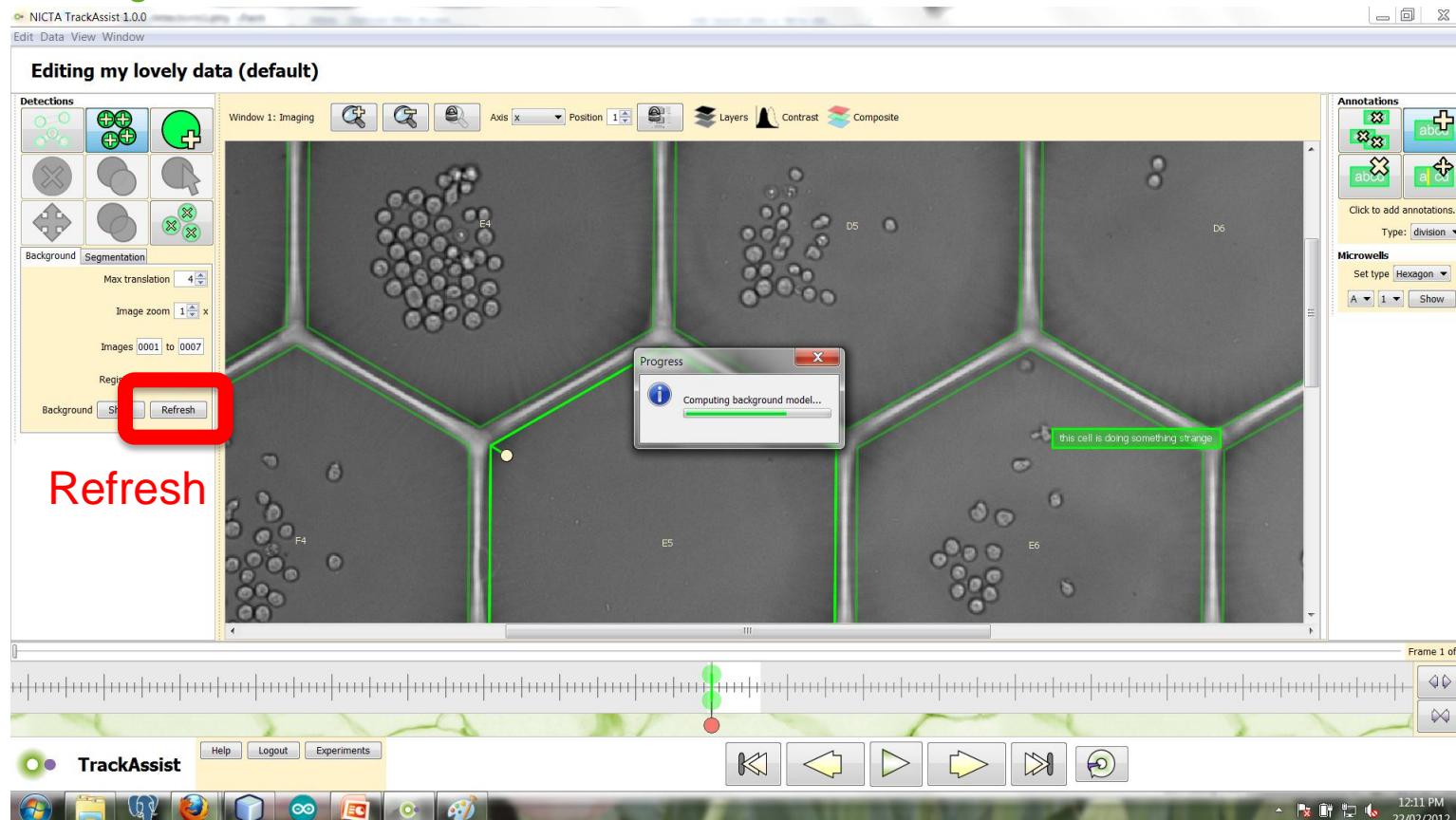
- Image Range to include in Background Model
 - Some consideration should be given to the number of images used to build the background model (ideally as many as possible). To ignore moving cells, we need them to move around, exposing the background underneath them. We therefore prefer time periods where the cells are highly motile. Sometimes, at the end of sequences there is little background visible. In this case, ignore the frames from the end of the sequence.



Program Walkthrough – Detections



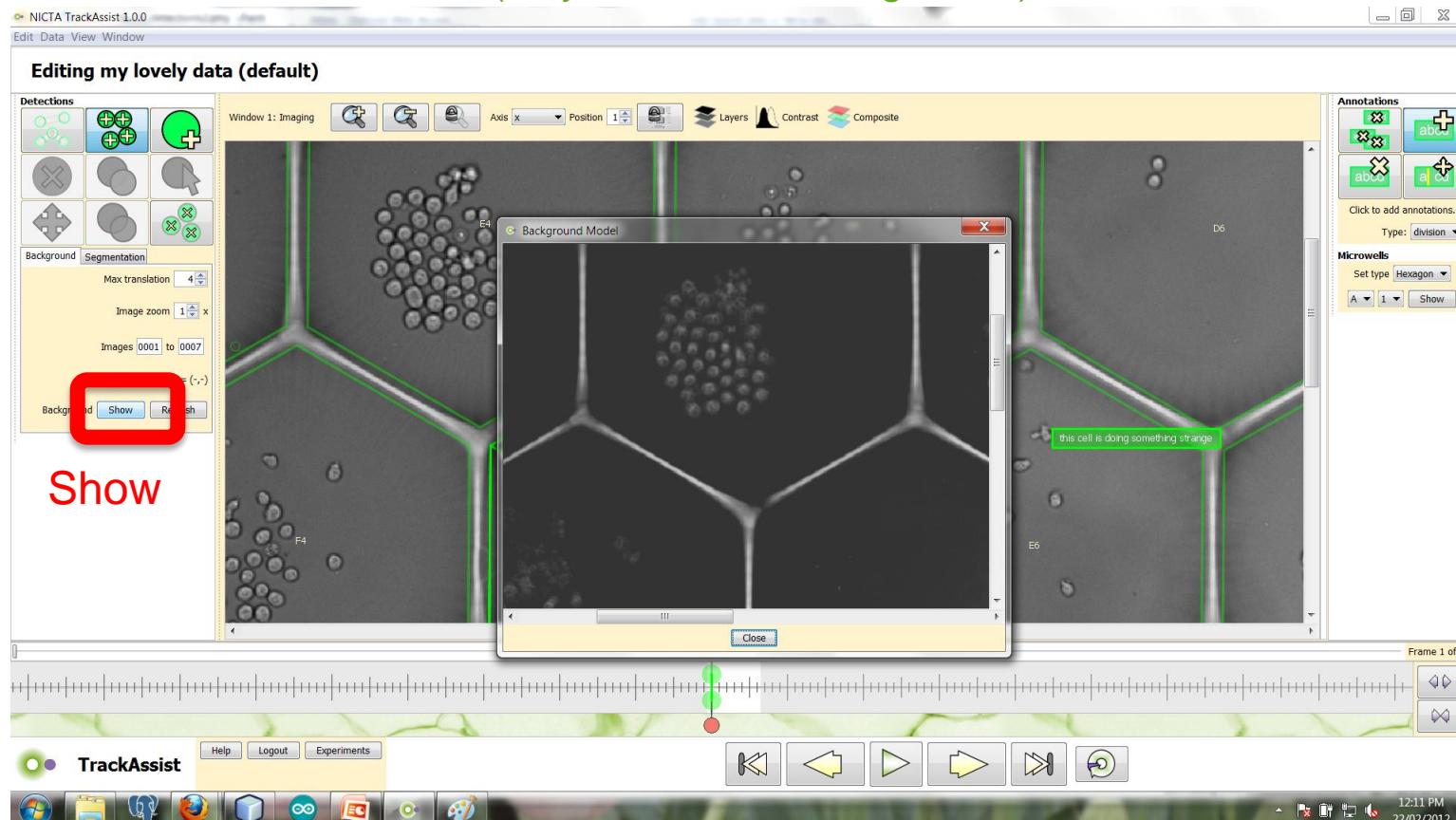
- We need to build a model of the non-moving appearance of the image
 - Choose the parameters (e.g. not too many images perhaps) and click “Refresh” to cause the model to be built. This may take many minutes to process several GB of images...



Program Walkthrough – Detections



- When complete, click “Show” to display the background image.
 - You can make the window larger, and pan and zoom within the image. Zoom by scrolling the mouse wheel. Pan by dragging the mouse. The background should be visible, but not the cells (they should be averaged-out).



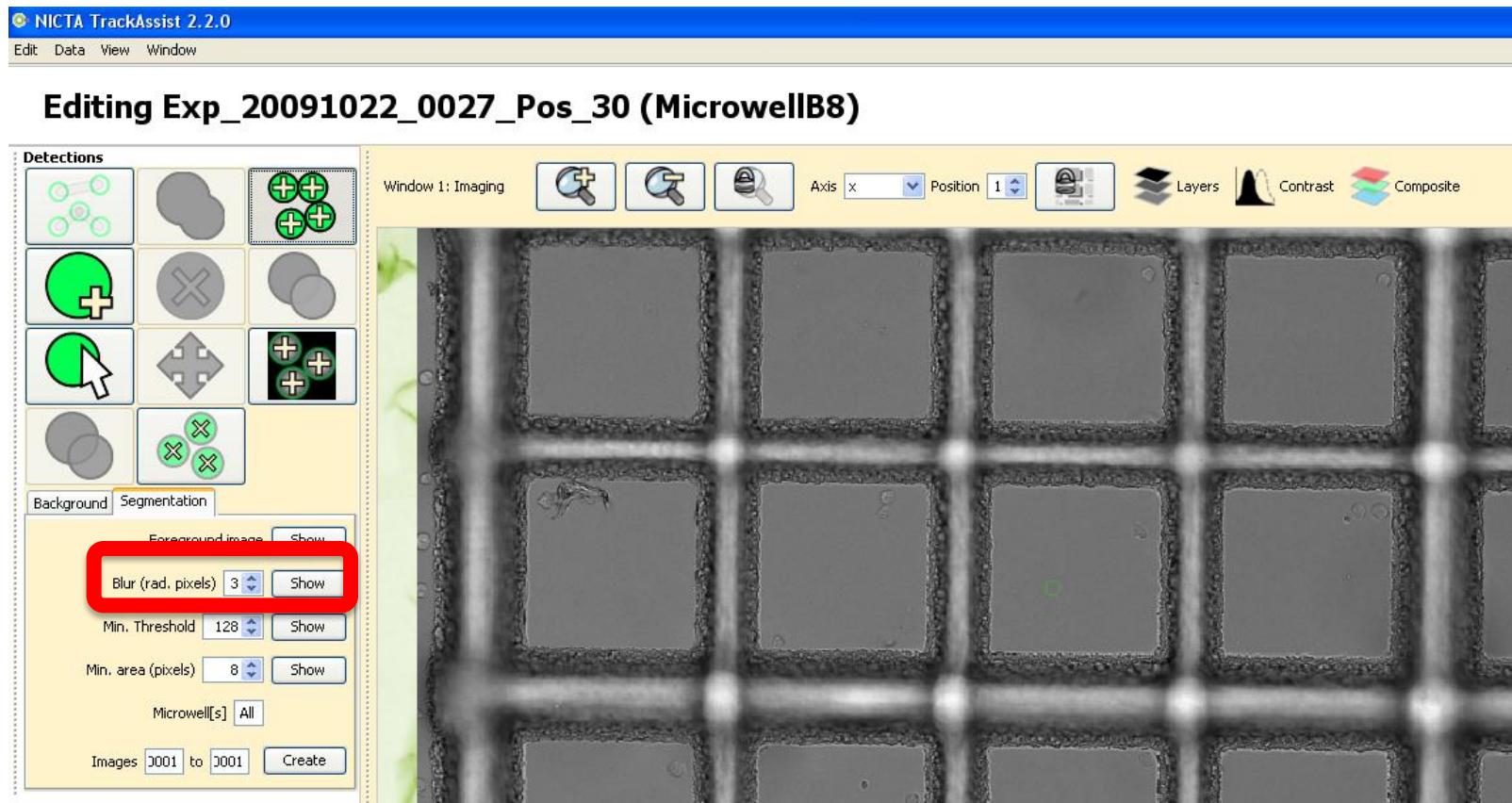
Program Walkthrough – Detections

- We will next use the background image to segment cells.
 - Click the “Segmentation” tab. Click the “Show” buttons to show each step of the image processing. Adjust parameters as necessary. When satisfied, click “Create” to create detections by segmenting cells.



Background Segmentation Parameters

- When using a background image to segment cells:
 - Blur : The kernel radius of a Gaussian smoothing filter. This removes high frequency noises and smoothes the image. This reduces oversegmentation (erroneously dividing cells). Values typically 3-7. Use “Show” to see the effect of the blur.



Background Segmentation Parameters

- When using a background image to segment cells:
 - Min threshold : The pixel intensity value (between 0 and 255) to define a pixel to be foreground. Values < threshold are background (ignored).
 - Default is 128, however, click the “Show” button and check the result. Note that each “Show” button shows cumulative processing i.e. foreground, blur, & threshold.



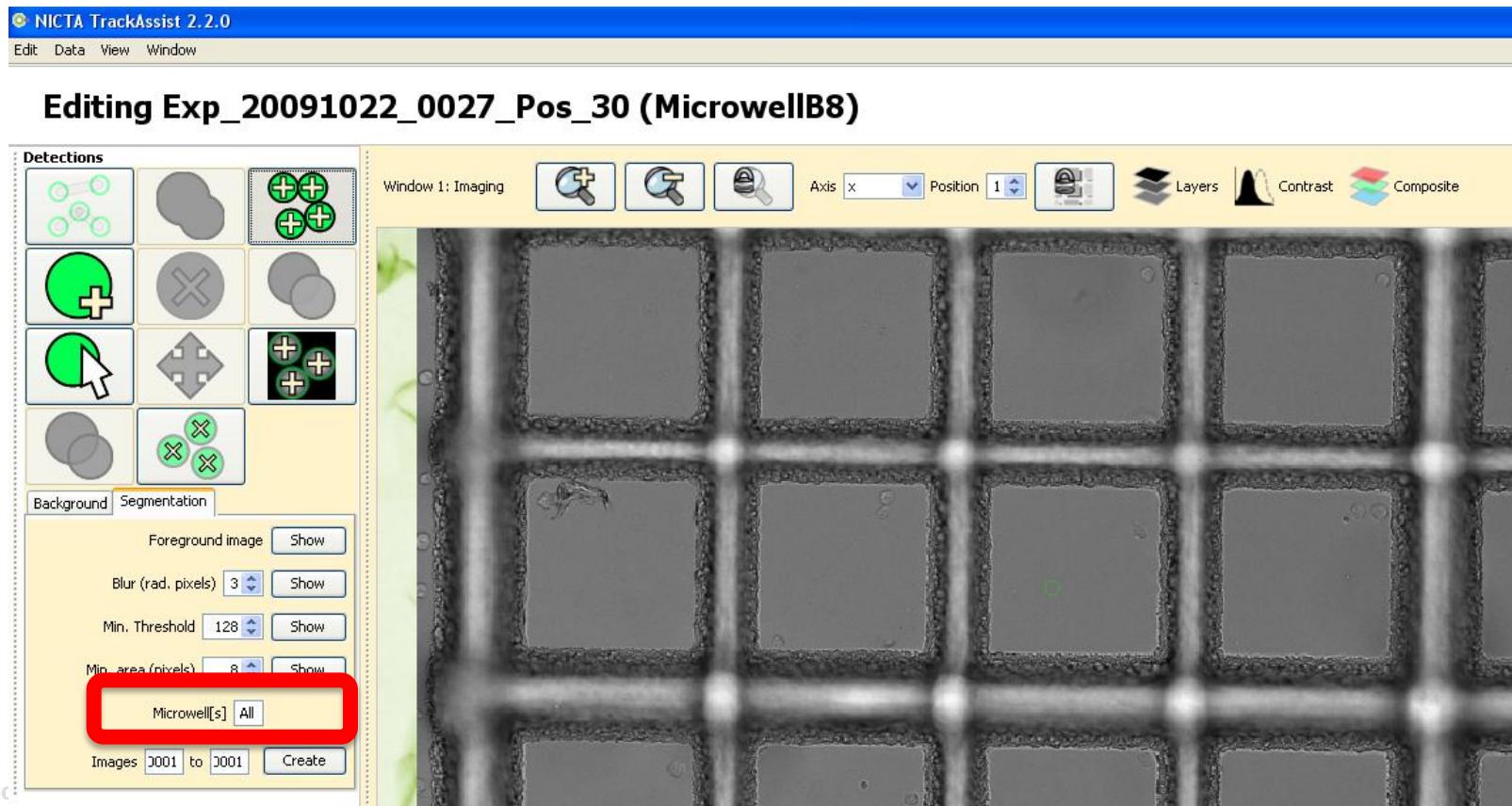
Background Segmentation Parameters

- When using a background image to segment cells:
 - Min area : Minimum area (in pixels) of valid cells. Any foreground component below this size will be ignored, and will not become a detection.
 - Use the show button to see the result of cumulative processing and resulting filtered foreground components.



Background Segmentation Parameters

- When using a background image to segment cells:
 - Microwells: Default is “All”, i.e. whole image. To select individual microwell[s], give them by name in a comma-separated list e.g. B3,B4. We recommend working with a single well at a time. Foreground components in other microwells will be ignored and will not become detections.



Background Segmentation Parameters

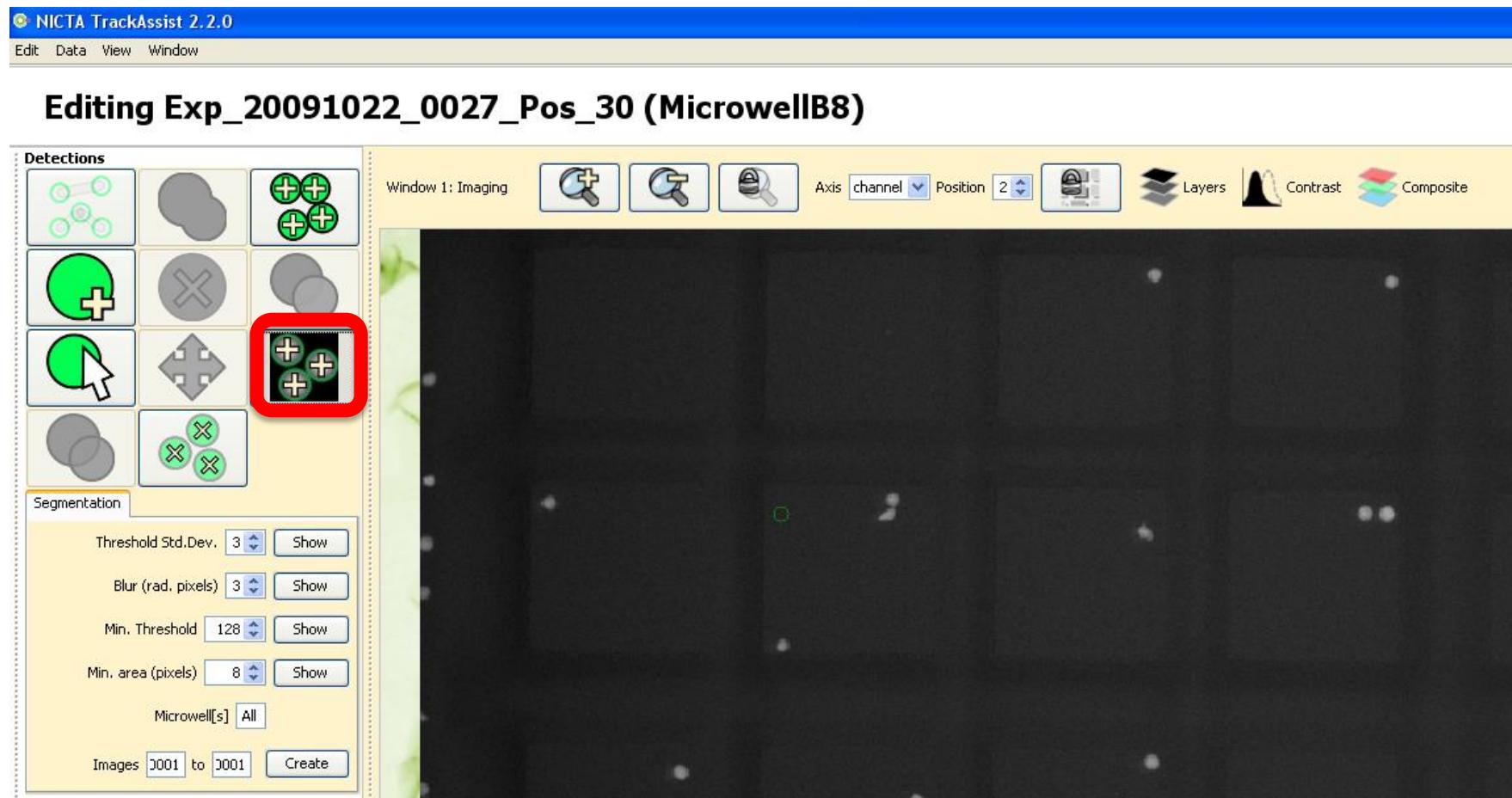
- Filtering by image range:

- The parameters may sometimes need to vary throughout an experiment due to lighting drift or the growth of cells. The image-range defaults to the current image. We recommend testing parameters on a single image, then applying in ranges of 300-500 frames and checking periodically that the results are still acceptable.



Constant Fluorescence Parameters

- The images are assumed to be mostly dark with lighter cells.



Constant Fluorescence Parameters

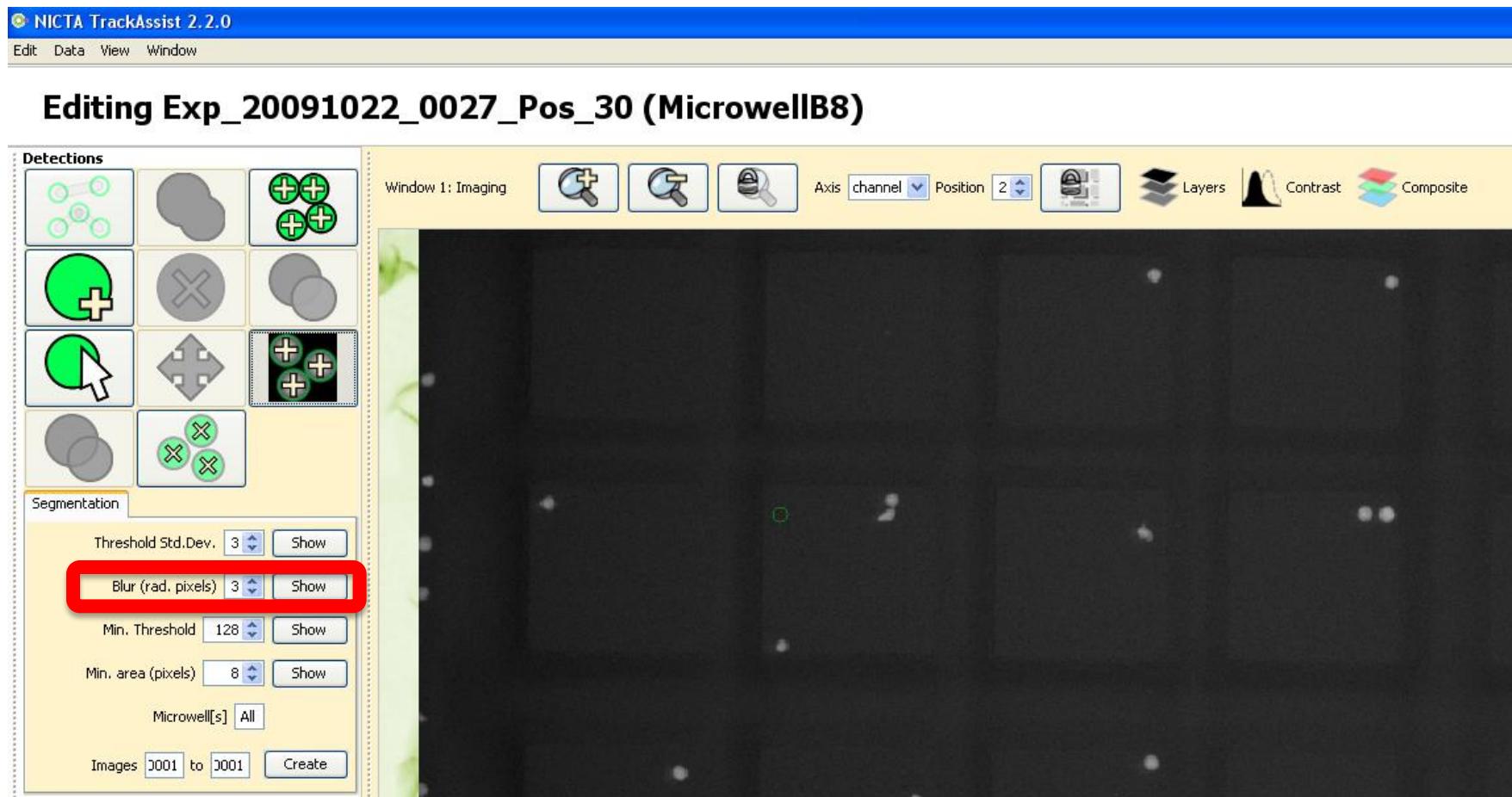
- Threshold Std. Dev. : We normalize the intensity of the original image and place a threshold defined in terms of standard deviations from the image mean intensity. This parameter is the spread or standard-deviation of the normalized image.



Constant Fluorescence Parameters



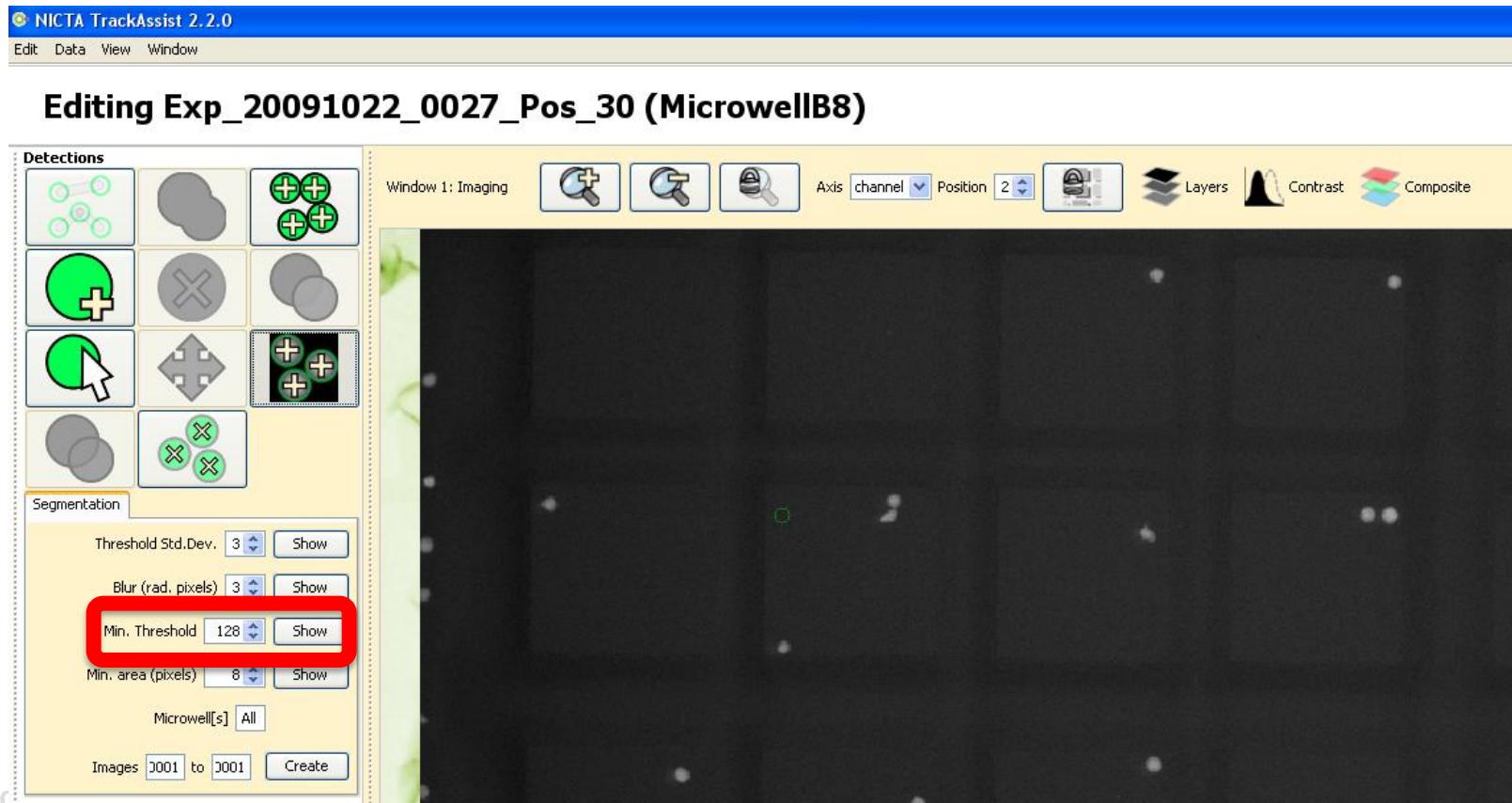
- Blur : The radius of a Gaussian kernel used to smooth the image and prevent over-segmentation. Value typically 3-7. Ideally, after the blur operation, the cells will have a smooth internal texture peaking in the centre of the cells.



Constant Fluorescence Parameters



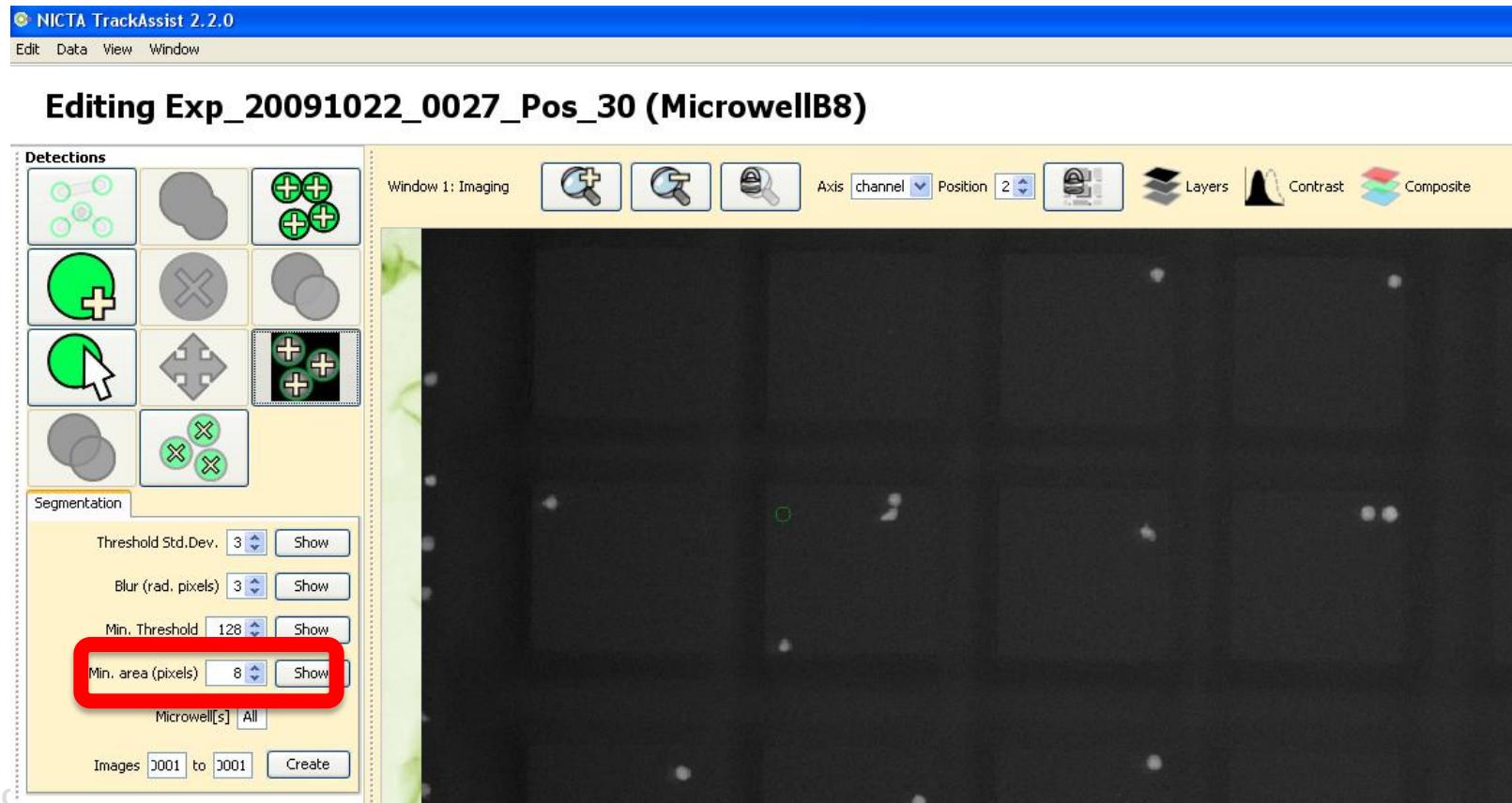
- Min threshold : The image intensity is scaled to between 0 and 255. Pixels less than this threshold will be ignored. Pixels higher than the threshold can become part of detections.
- Use the “Show” button to verify an appropriate threshold has been given. No background should be visible in the thresholded image.



Constant Fluorescence Parameters

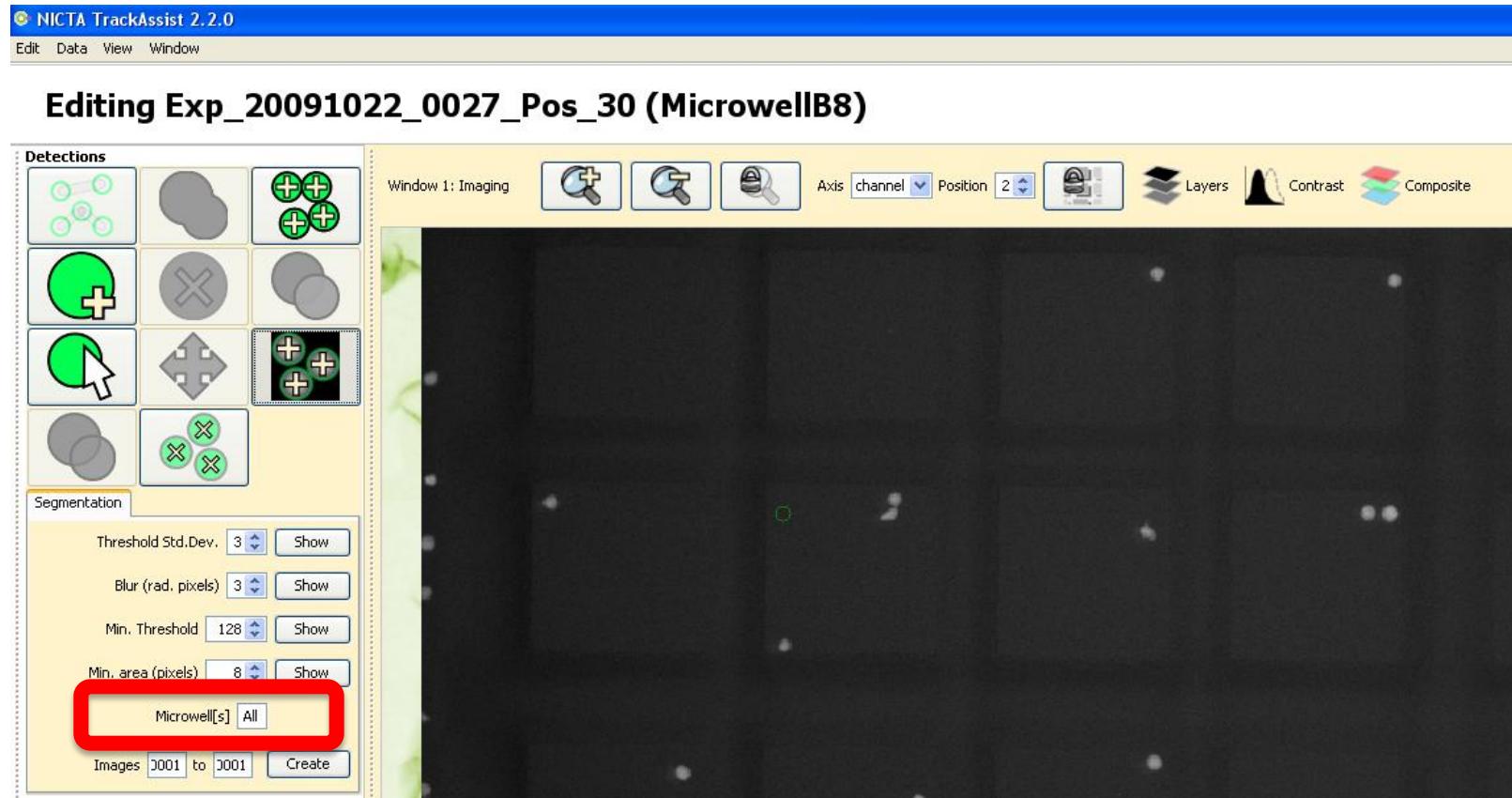


- Min area : Cells or other foreground components below this size (in pixels) will be filtered and will not become detections.



Constant Fluorescence Parameters

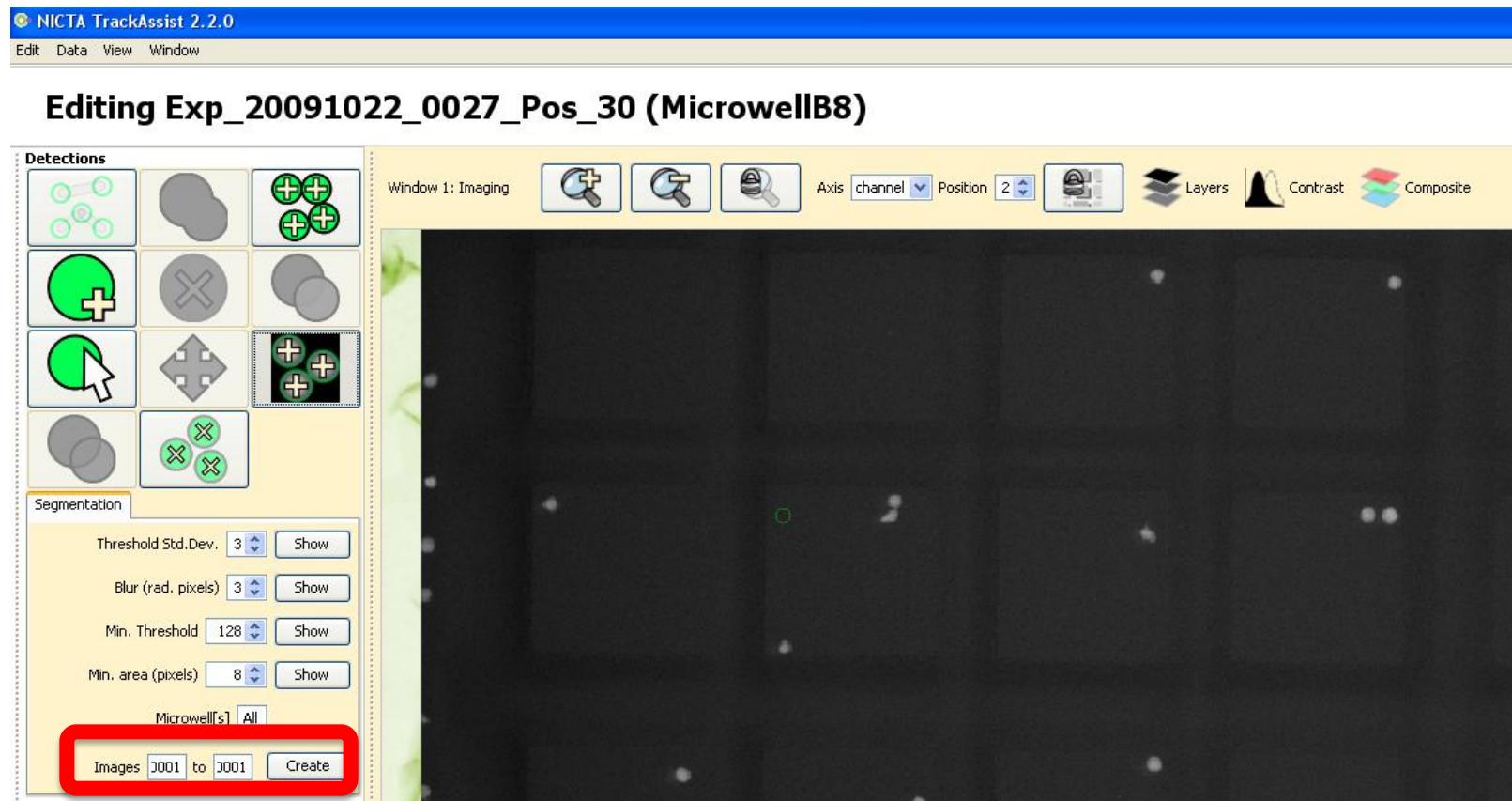
- Microwells: Default is “All”, i.e. whole image. To select individual microwell[s], give them by name in a comma-separated list e.g. B3,B4. We recommend working with a single well at a time. Foreground components in other microwells will be ignored and will not become detections.



Constant Fluorescence Parameters



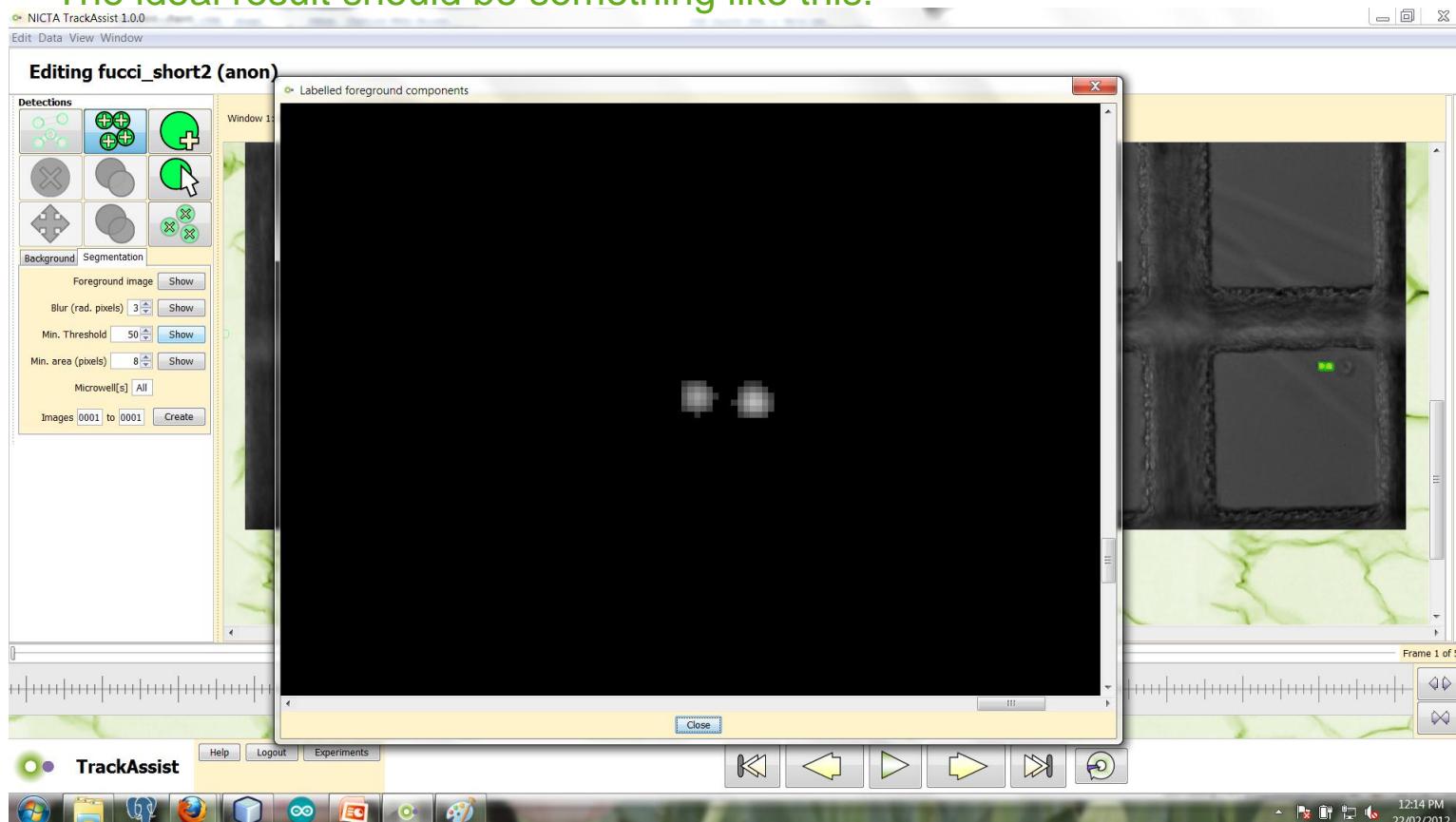
- All parameters may need to vary throughout an experiment. The image-range defaults to the current image. We recommend testing parameters on a single image, then applying in ranges of 300-500 frames and checking periodically that the results are still acceptable.



Program Walkthrough – Detections



- The objective is to retain the cells and exclude all the background.
 - We recommend you filter detections to a single specific microwell. Process each microwell in a separate solution. Test the results with a few images first.
 - The ideal result should be something like this:



Program Walkthrough – Detections



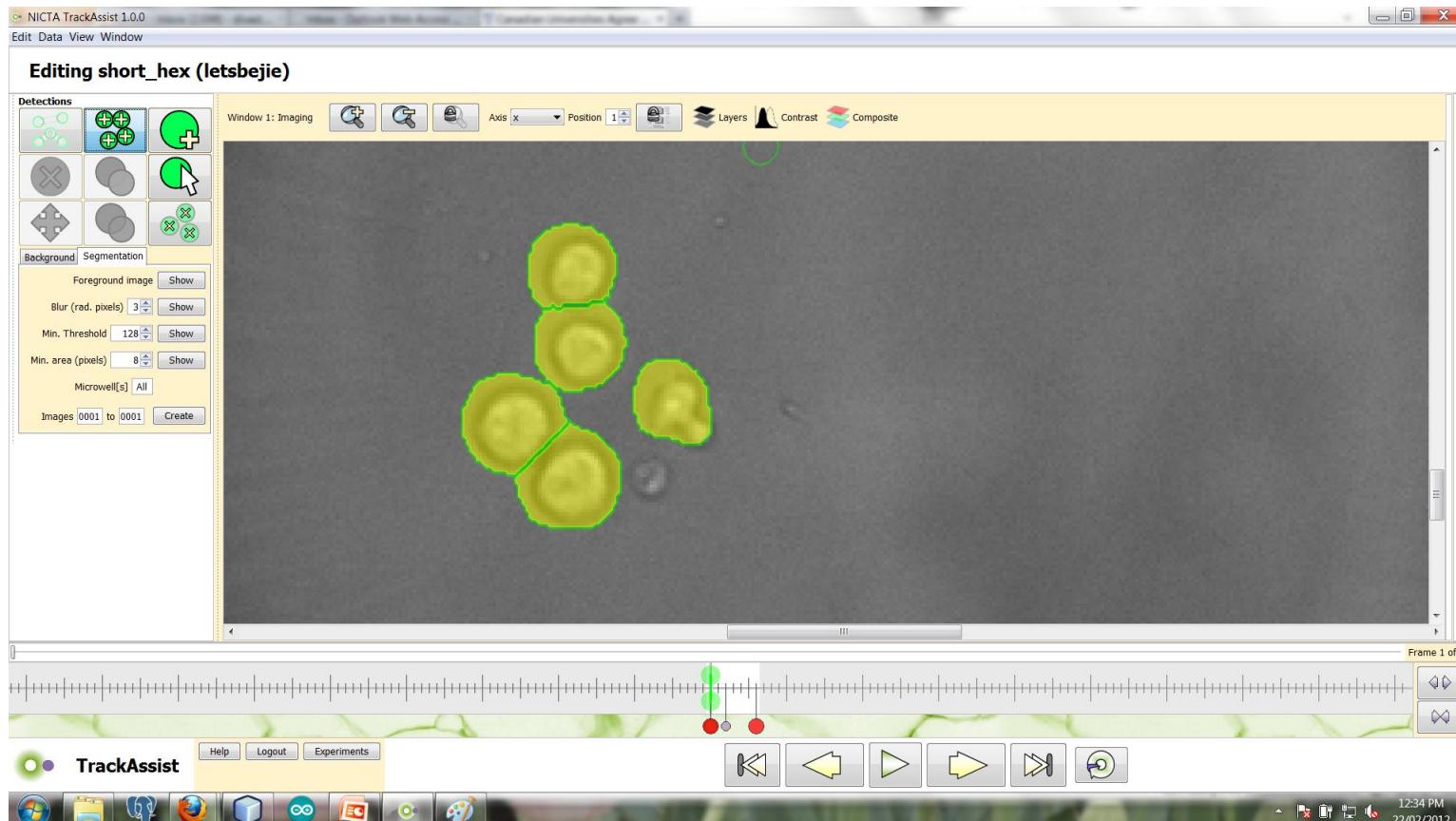
- After clicking “Create” the detections will be created.
 - Detections appear as yellow-green blobs.
 - Let’s zoom and pan to get a clear view of some detections:



Program Walkthrough – Detections

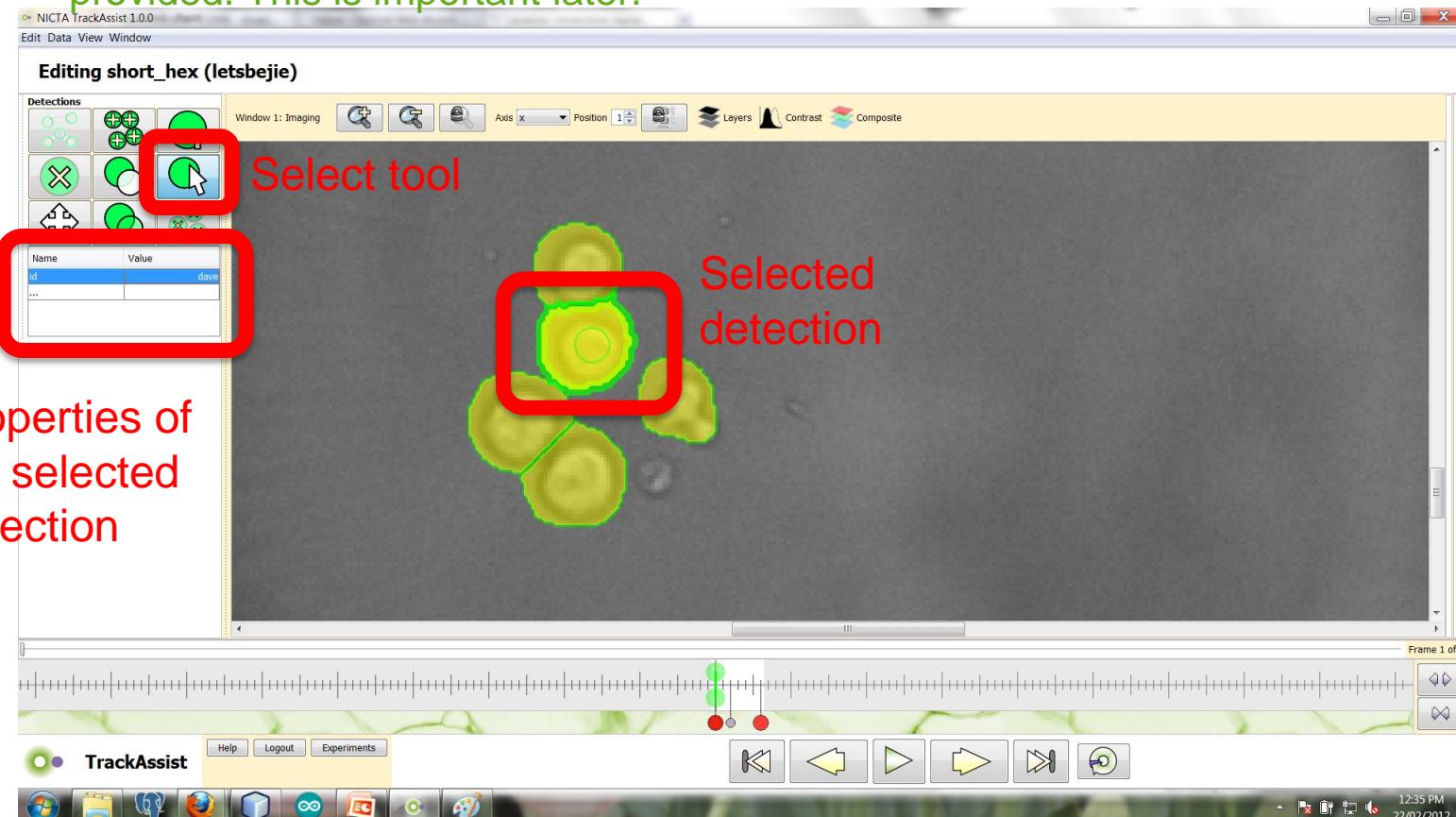


- Here are some detections I made earlier...
 - There are several “painting” tools you can use to edit detections
 - Typically, these tools affect only selected detections.



Program Walkthrough – Detections

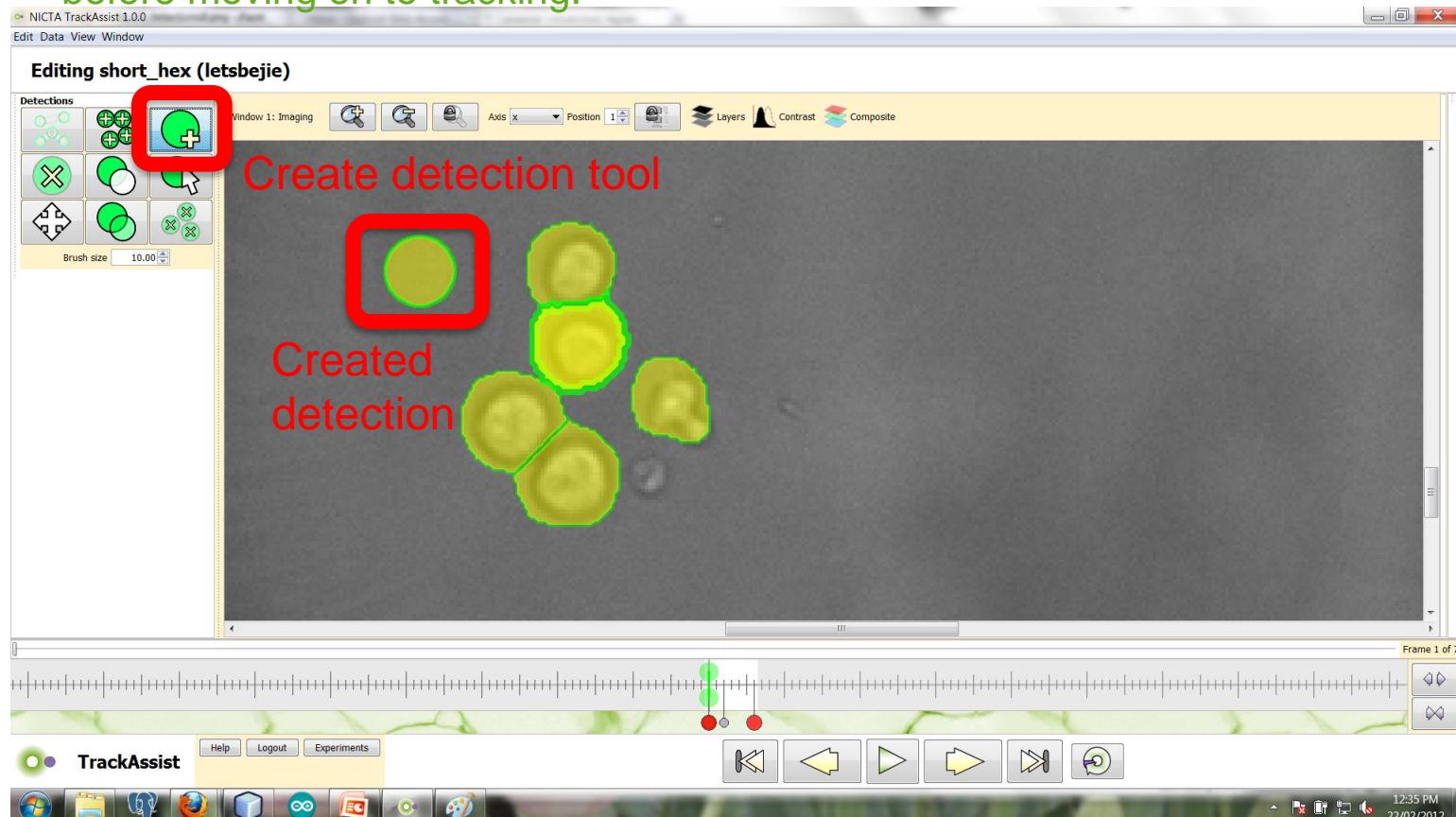
- Select some detections to edit.
 - Use the tool with the mouse pointer icon. Here, one detection has been selected.
 - You can also modify “properties” of a single selected detection using the table provided. This is important later.



Program Walkthrough – Detections



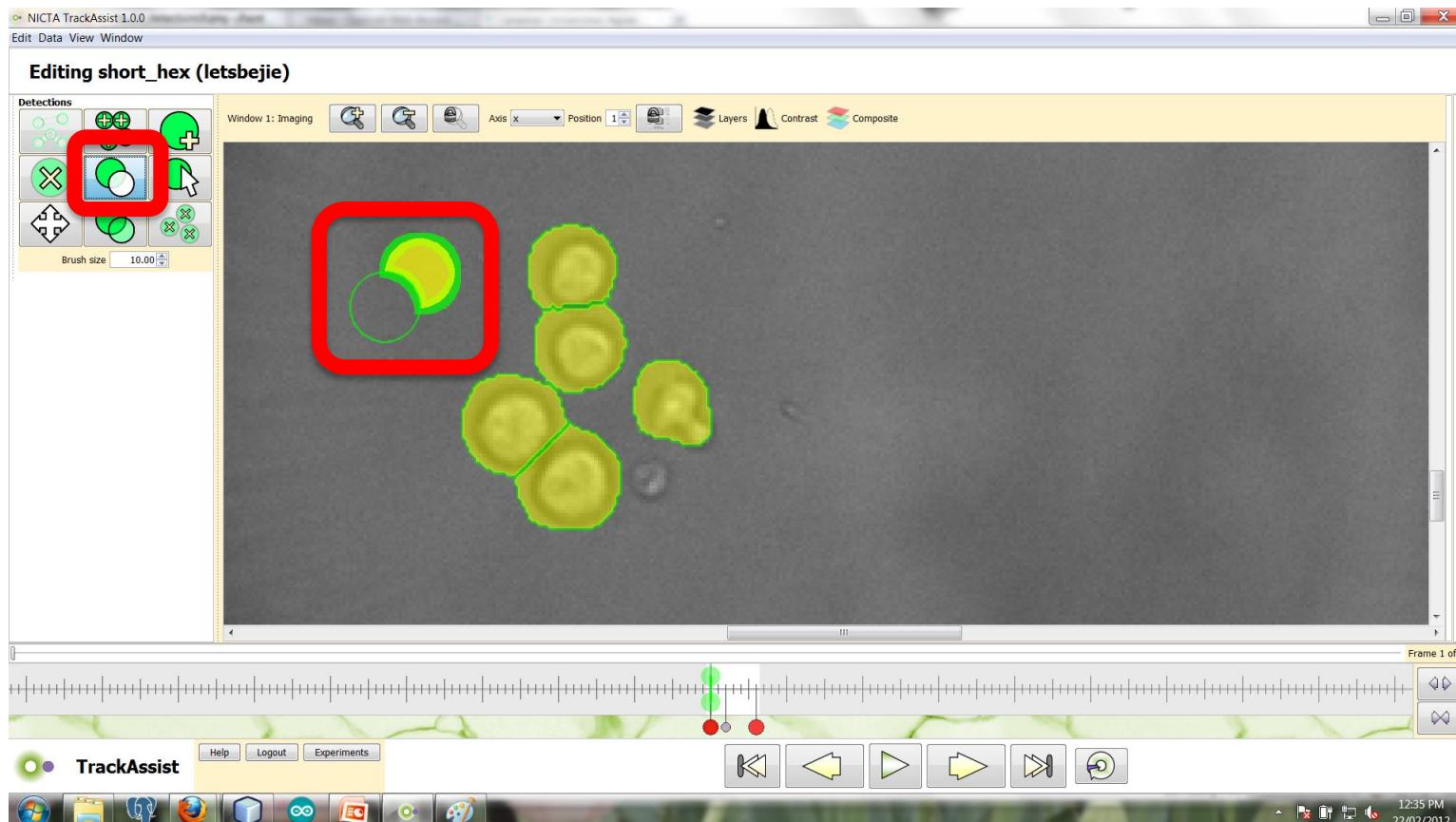
- Select “Create detection” tool (+) and click image to create detections
 - You can create and delete as many detections as you like
 - We recommend you delete **all** erroneous detections and add **all** missed detections before moving on to tracking.



Program Walkthrough – Detections



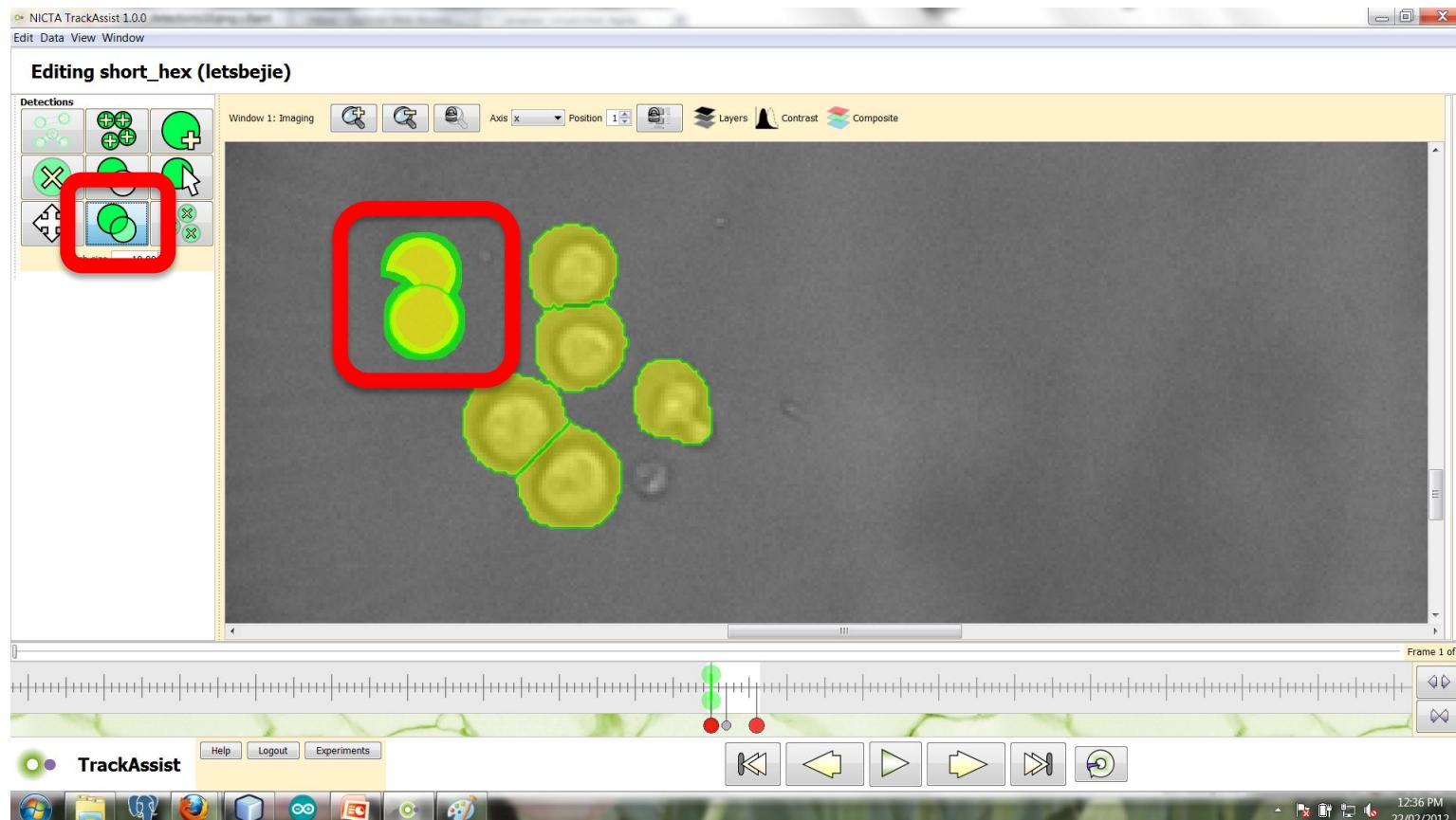
- You can use the “add” and “remove” painting brushes to change the contours of detections
 - Here we are removing part of a detection



Program Walkthrough – Detections



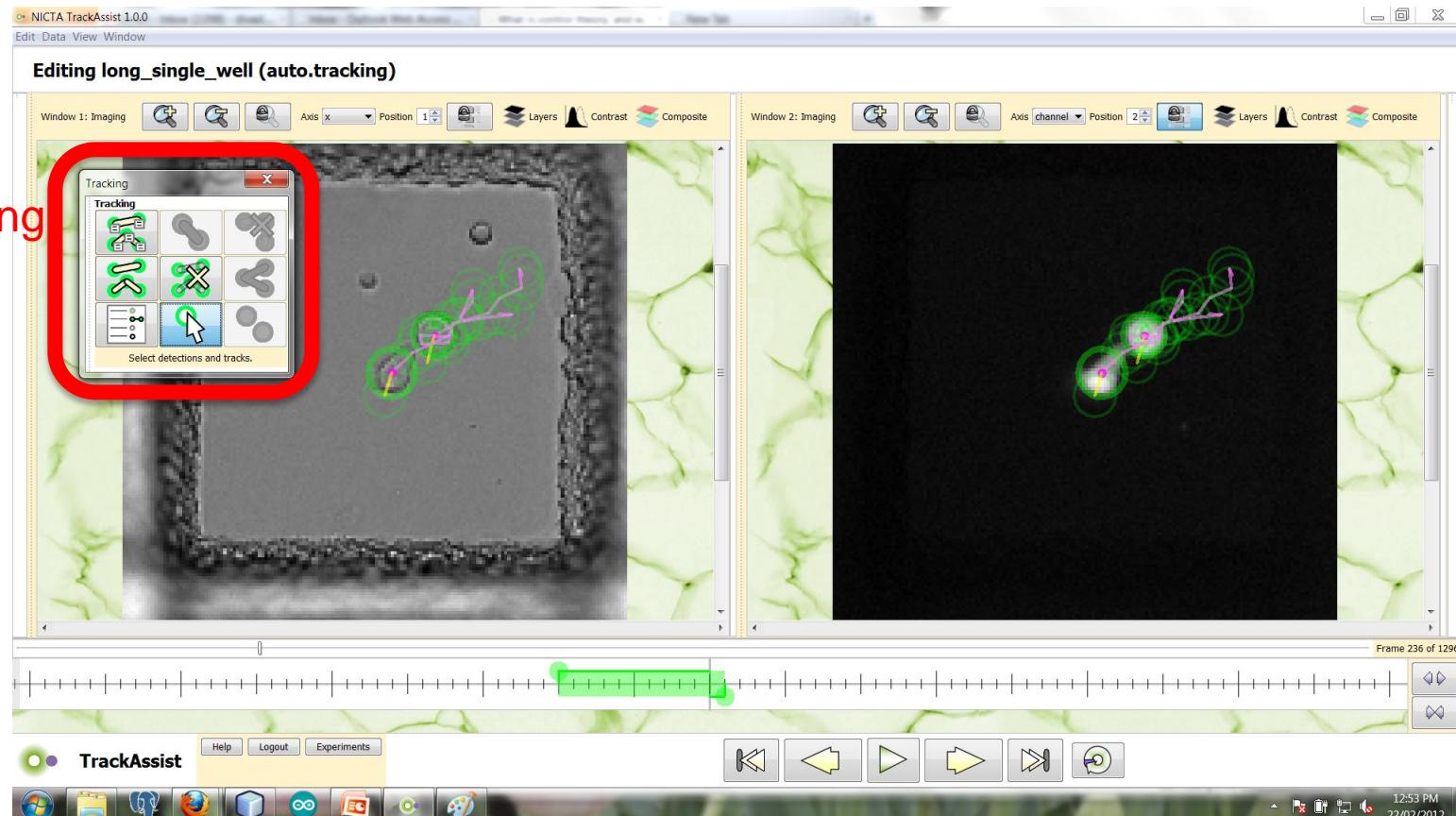
- You can use the “add” and “remove” painting brushes to change the contours of detections
 - We have just added some area to the detection we created.



Program Walkthrough – Tracks



- We will first explain how tracks are displayed.
 - To view tracks, select from the Menu: View→Tracking
 - The tracking tools will appear. You can drag tools to undock them:

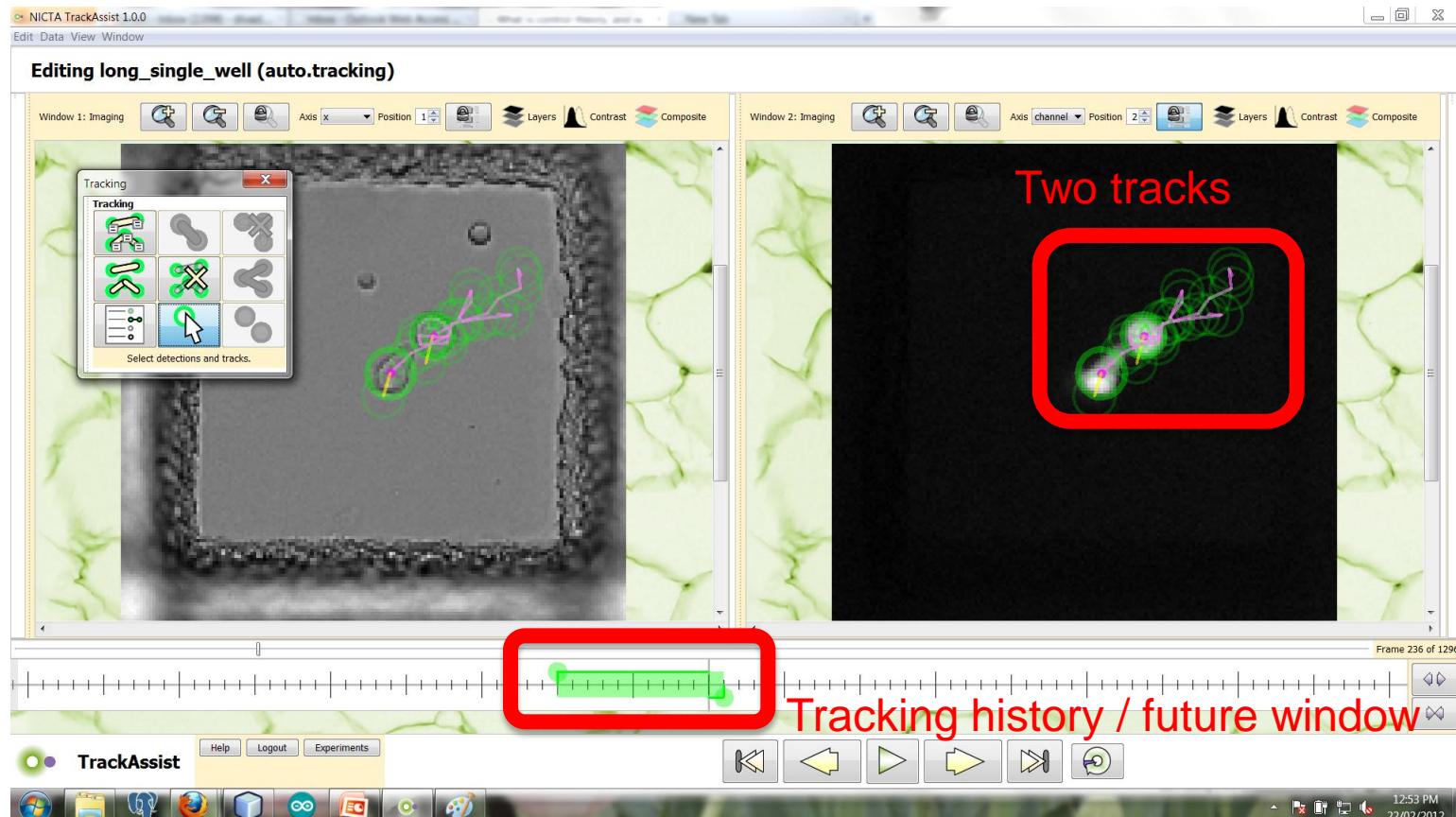


Tracking
tools

Program Walkthrough – Tracks



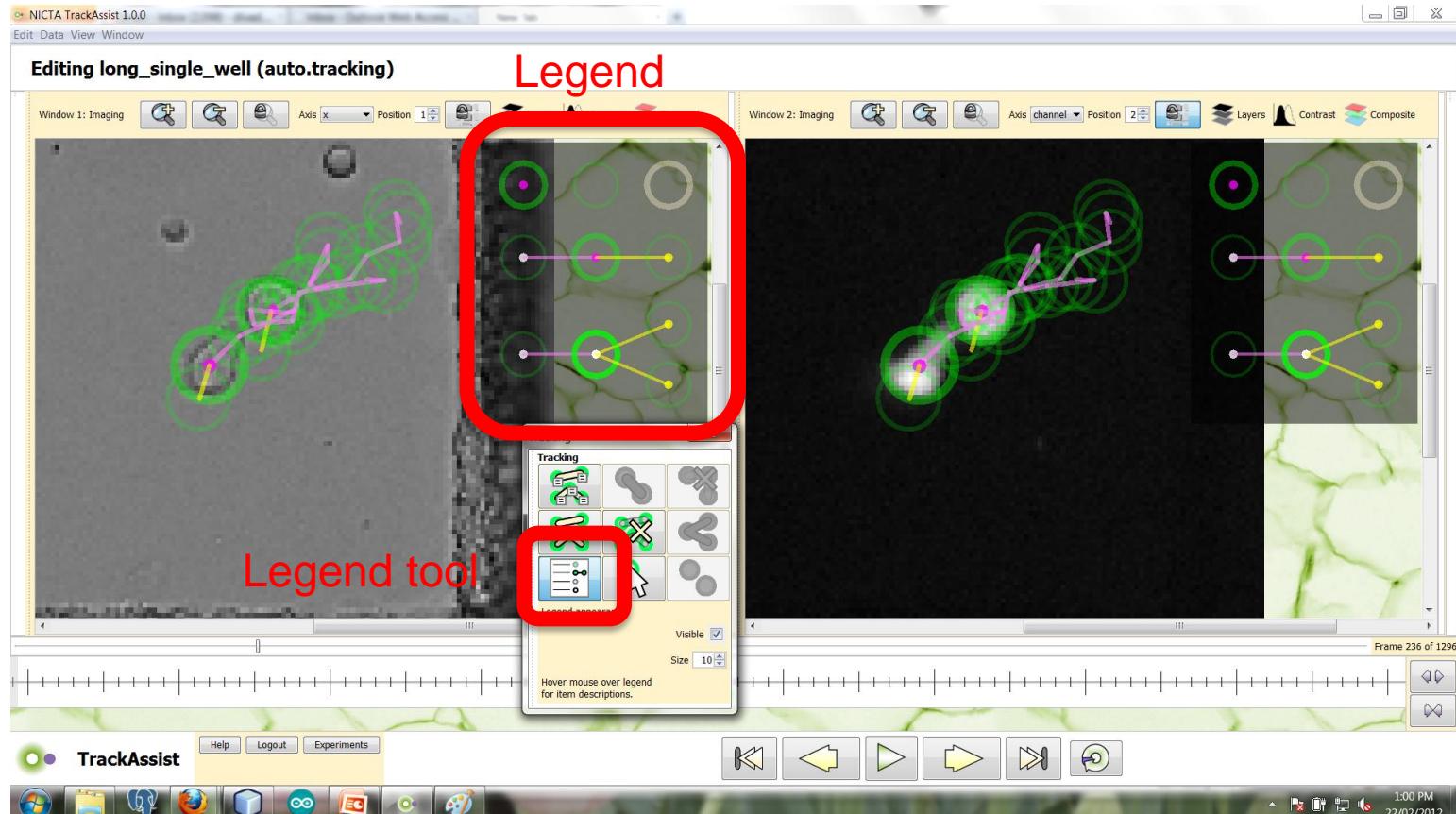
- Tracks are shown as a “history” and “future” of their detections.
 - To adjust the history and future windows, drag the green dots on the timeline.
 - Here, we are showing 1 future step and 10 history steps. You can see 2 tracks:



Program Walkthrough – Tracks



- Use the Legend to interpret the tracking view.
 - Click on the Legend tool then tick “Visible” in its tool card.
 - Hover the mouse over each item in the legend for a description.



Program Walkthrough – Tracks



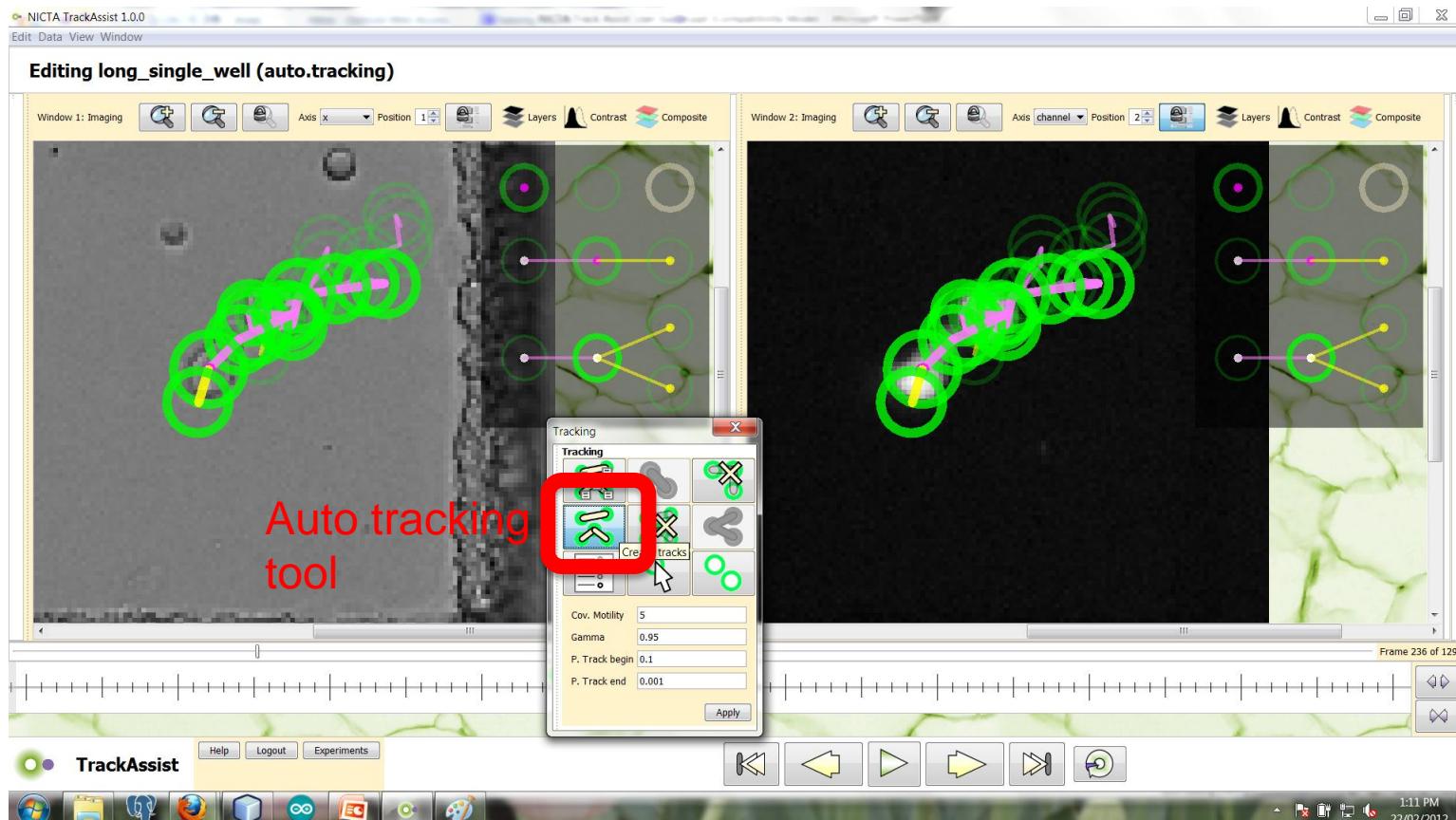
- Select tracks with the mouse pointer tool
 - Here we have selected one of the two visible tracks.
 - Selection is an important for editing tracks. It tells the program which track to modify



Program Walkthrough – Auto-Tracking



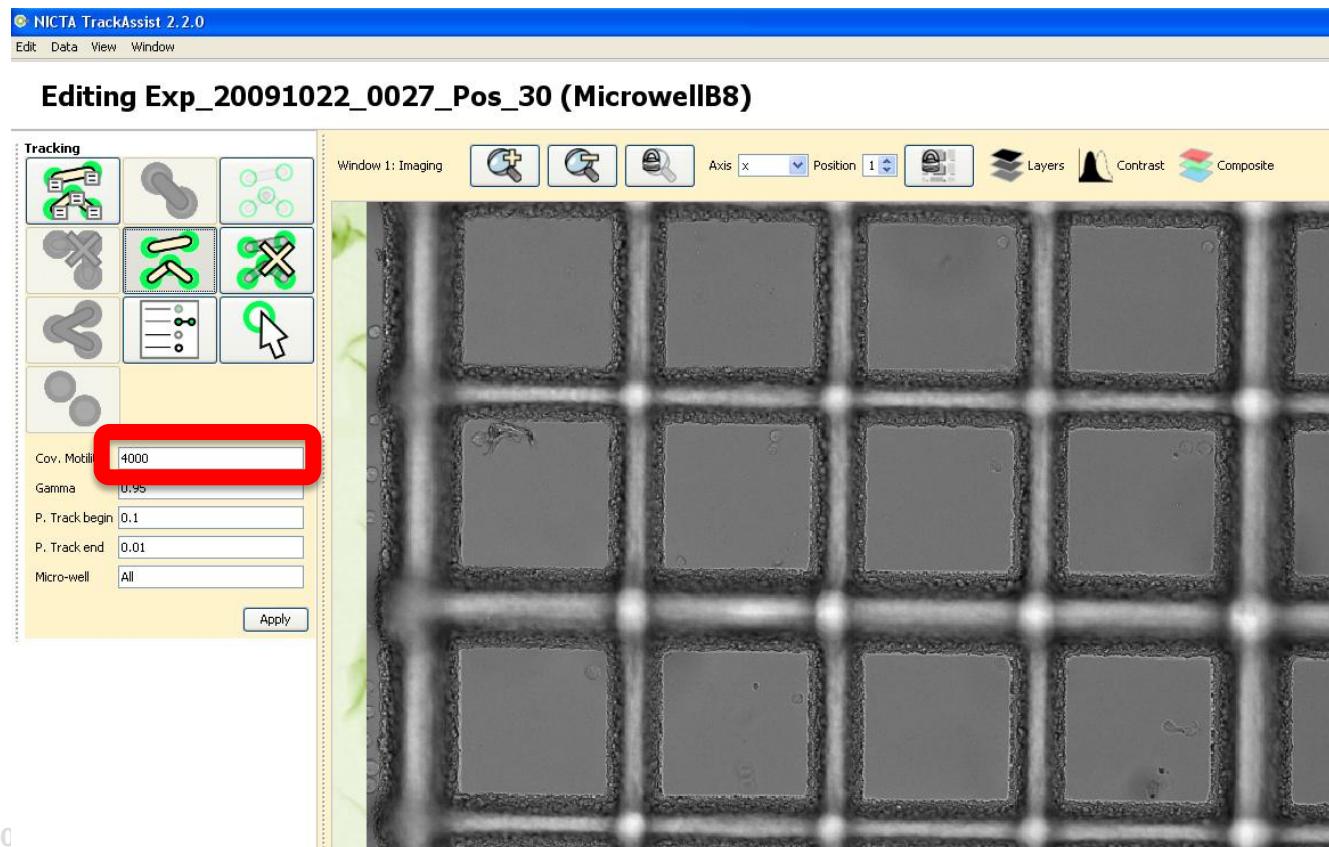
- The automatic tracking tool uses cell positions to track
 - Select the tool and optionally modify any parameters (the defaults should be fine)
 - Click “Apply” to re-generate all tracks in the solution. This may take some time.



Auto-Tracking Parameters

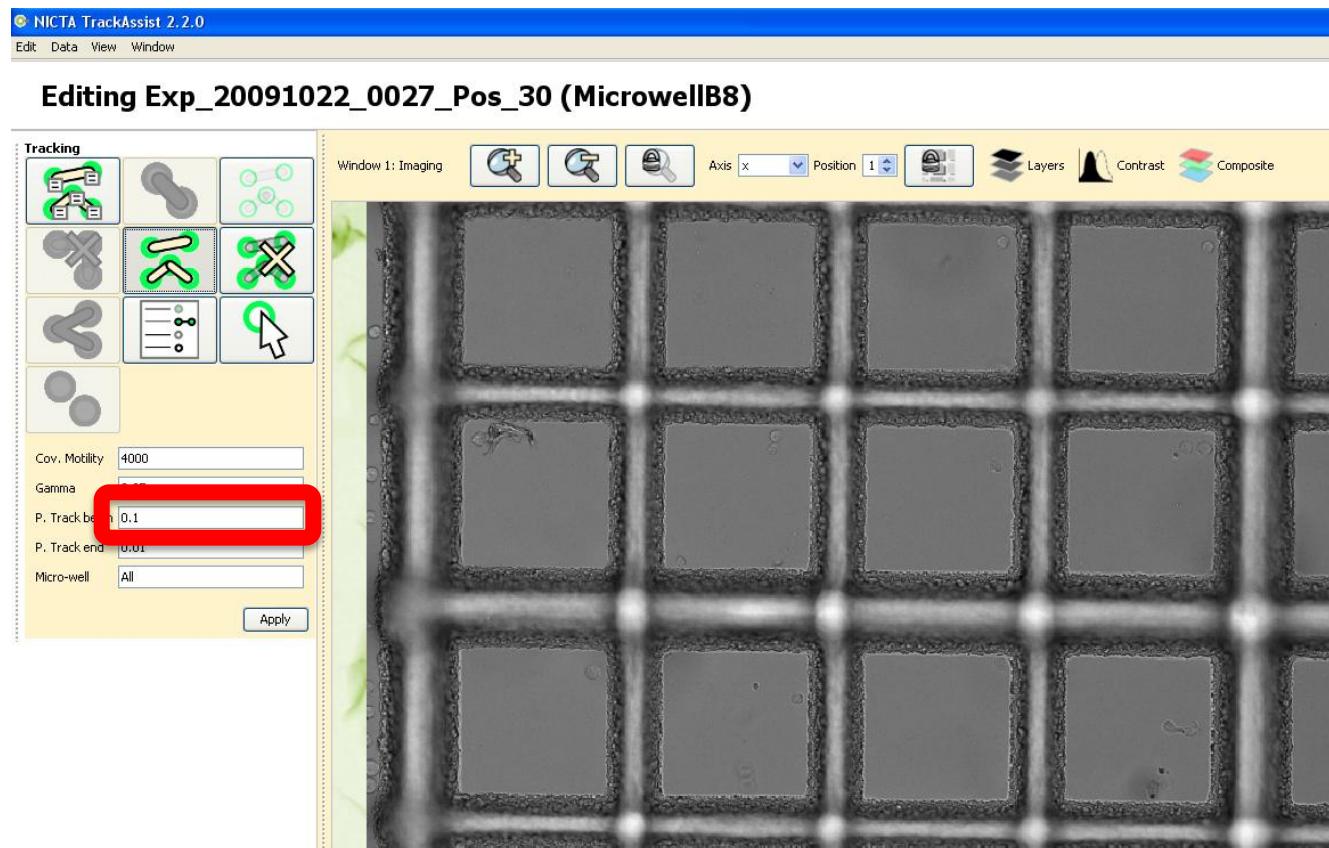
- Parameter : “Cov. Motility” [Q]

- The area of the “search window” (in pixels) when trying to find the continuation of a track in the next frame. $[Q = D^2 \text{ where } D = \text{the length of the search window}]$
- Default value is 4000 [roughly 65 pixels in each dimension]. For more motile cells it values go up to 8000. Try to limit the value to prevent



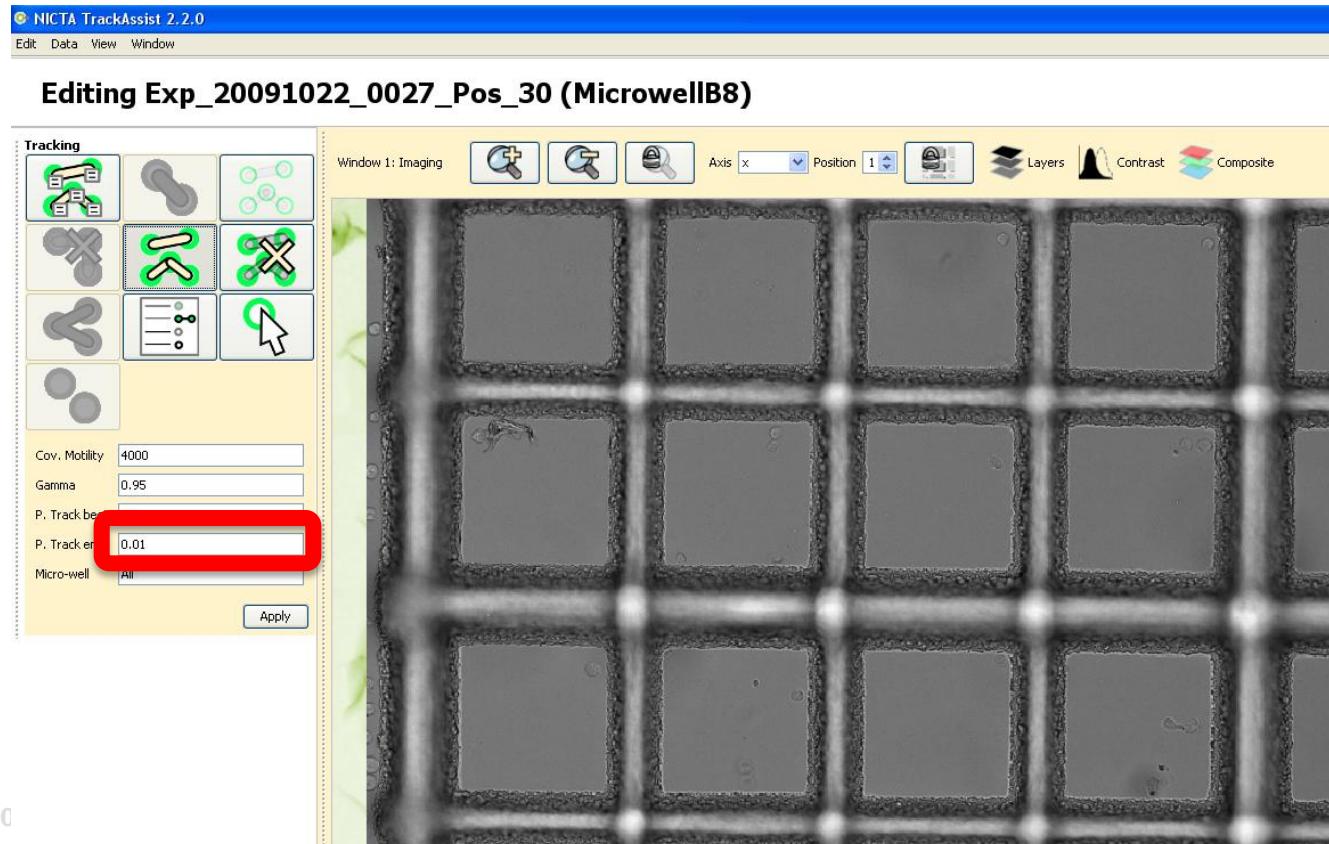
Auto-Tracking Parameters

- Parameter : “P. Track begin” [$p(E_0 = 1)$]
 - When initiated, each track is given this probability of existence [between 0 and 1] that is recursively updated in successive frames by measurements].
 - Default value is 0.1. This parameter rarely needs to be altered.



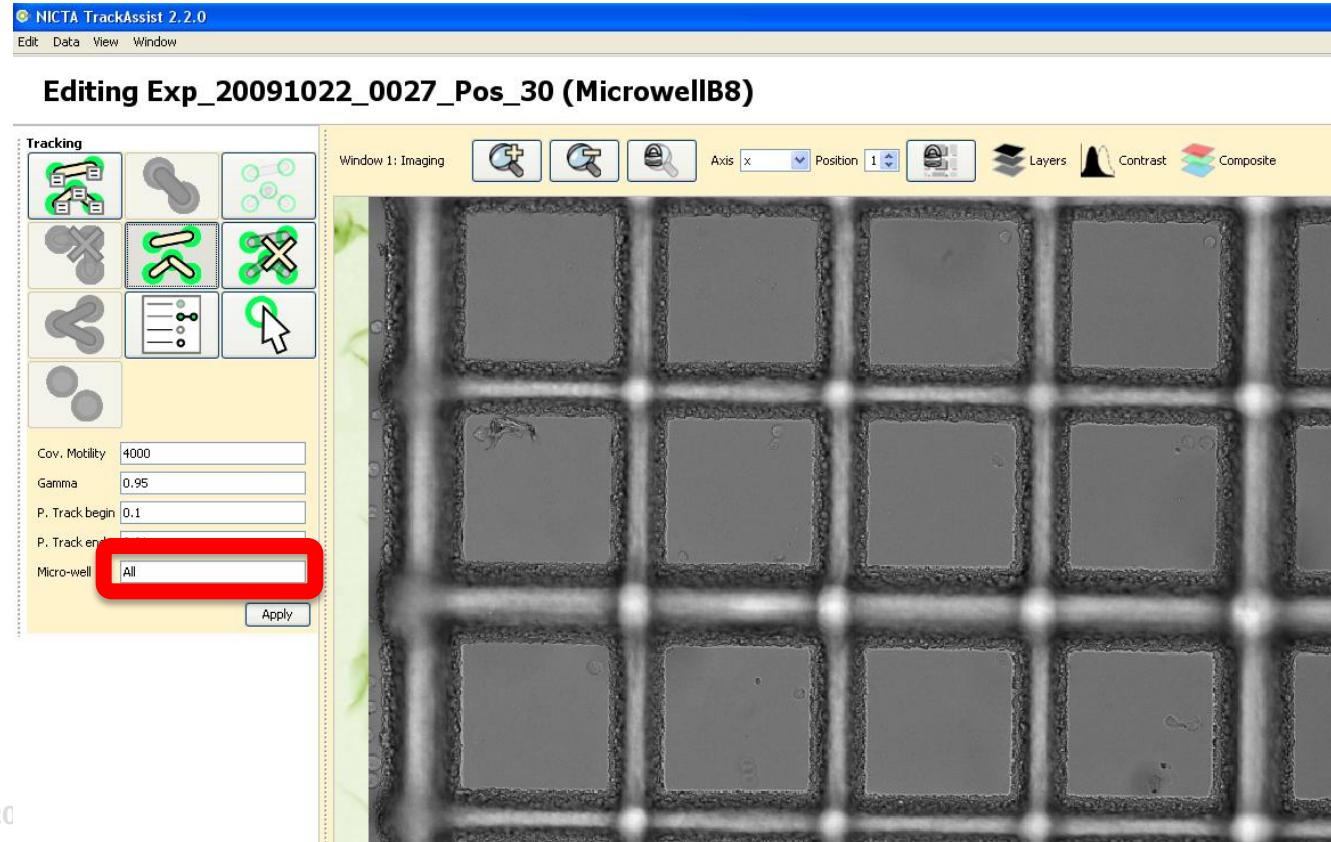
Auto-Tracking Parameters

- Parameter : “P. Track end” [$p(E_t = 1)$]
 - Tracks are terminated if the associated probability of existence is below this value [between 0 and 1]. Detection reliability has to be considered while setting this value. (the more accurate the detection is, the lower this value can be)
 - Default value is 0.01. For more accurate detection, this can be lowered to ~0.001.



Auto-Tracking Parameters

- Parameter : Microwell
 - Choice to run the tracking on individual microwells, or on all microwells.
 - Default is “All”. For individual microwells, give the names of the wells.

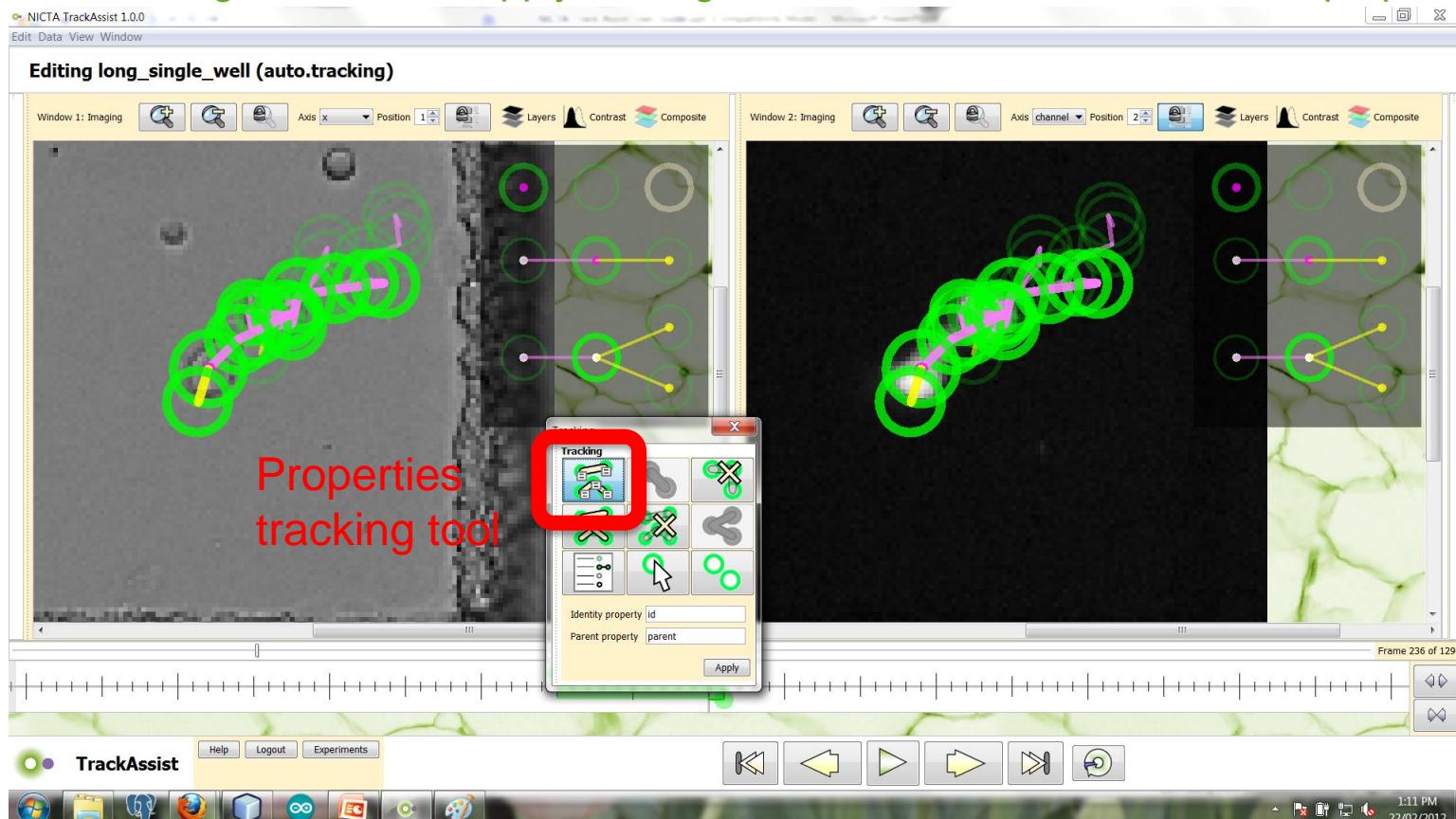


Program Walkthrough – Manual Tracking



NICTA

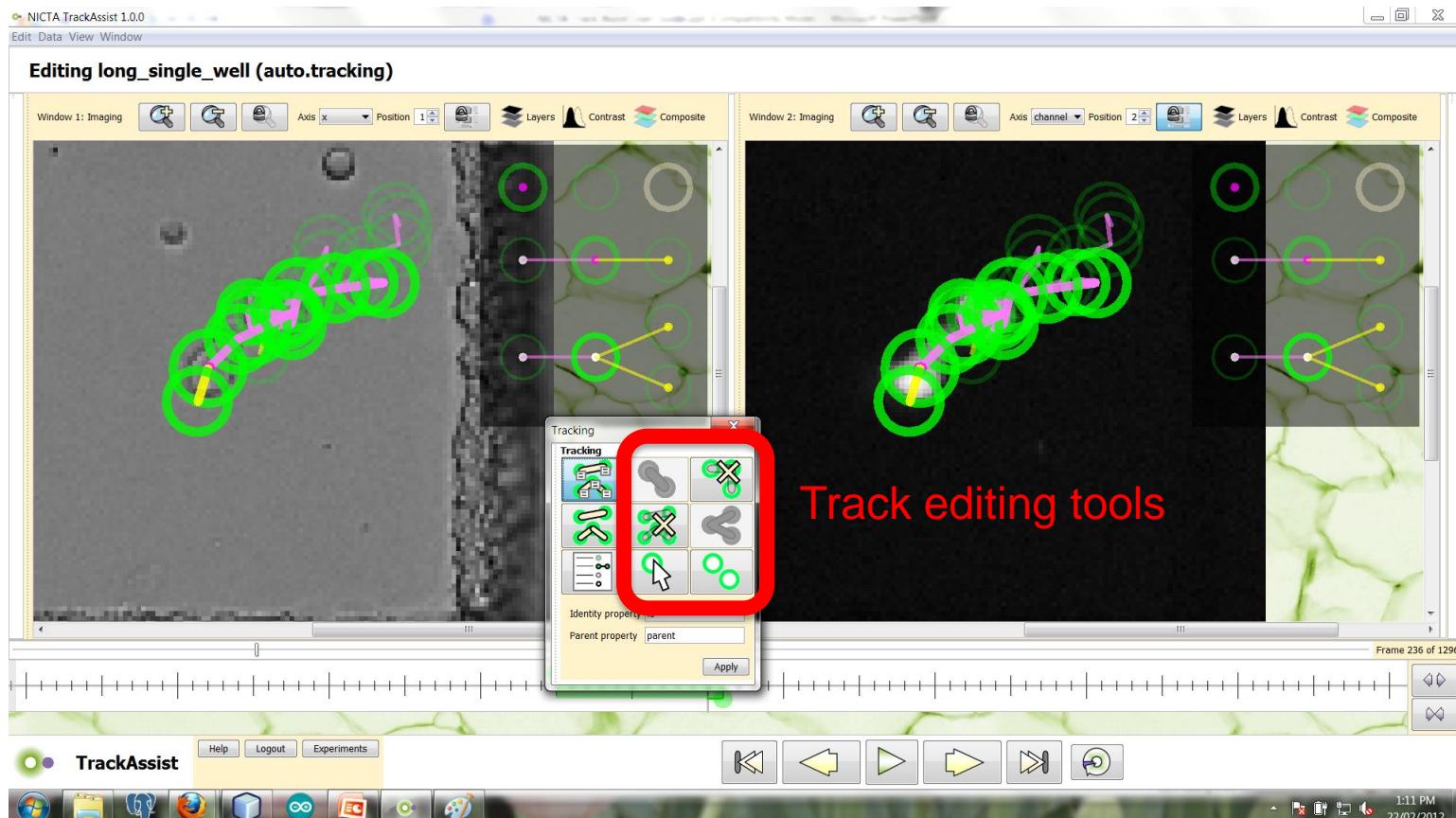
- An alternative is manually-defined tracking using detection properties.
 - This relies on you to set the **ID** and **PARENT** properties for any detections you want to use in tracking. You can ignore any detections that are **not** at branching points in the lineage tree. Click “Apply” to re-generate all tracks based on these properties.



Program Walkthrough – Editing Tracks



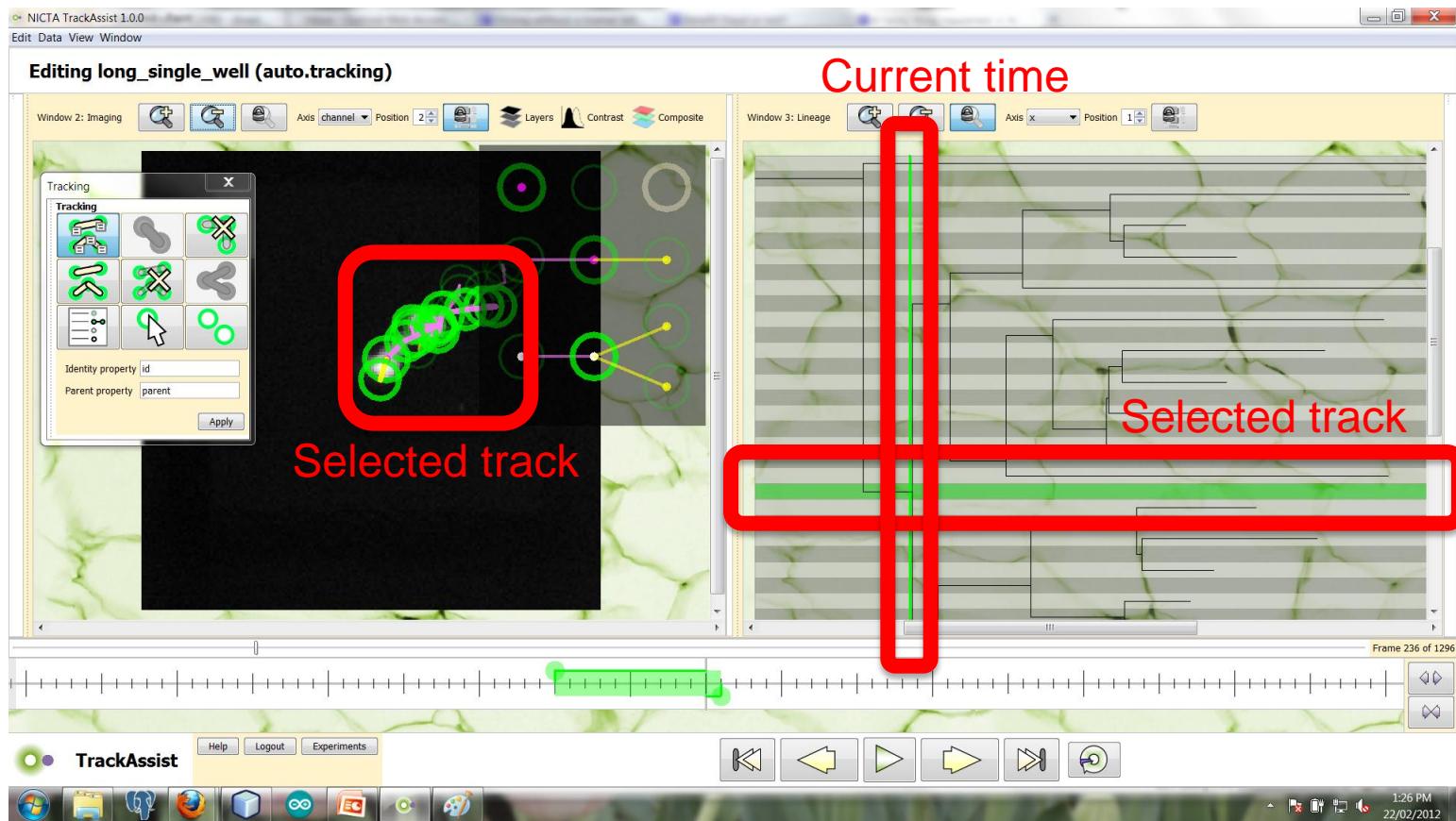
- Whichever method of tracking you have used, you can edit tracks
 - You can add and remove detections from select tracks and combine split tracks. You can also delete entire selected tracks or all the tracks in the solution.



Program Walkthrough – Lineage View



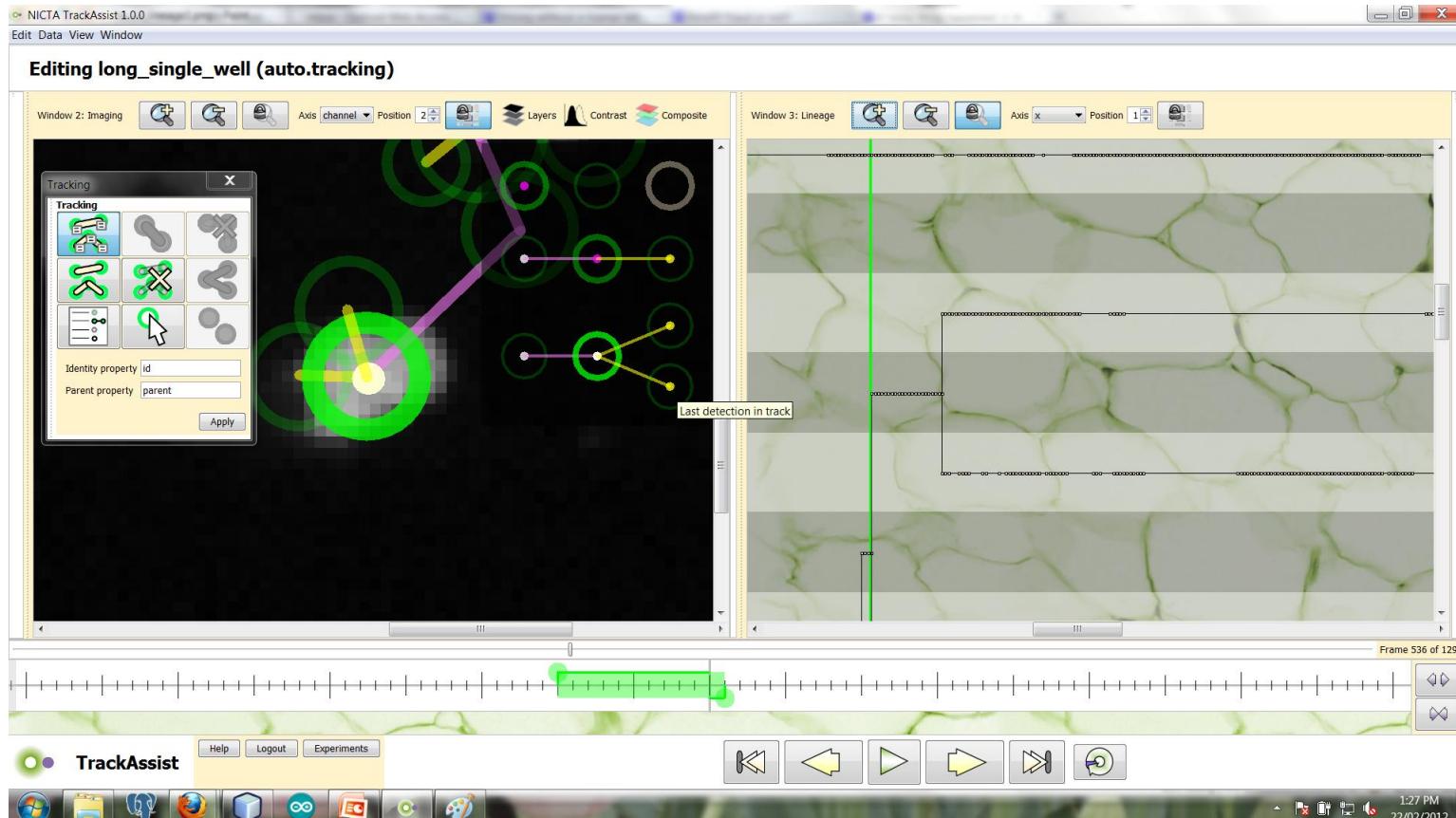
- The lineage is implicitly defined by the tracks.
 - From the menu select Window→Add→Lineage Window.
 - Lineage and imaging windows are linked; selecting tracks in either is visible in both:



Program Walkthrough – Lineage View



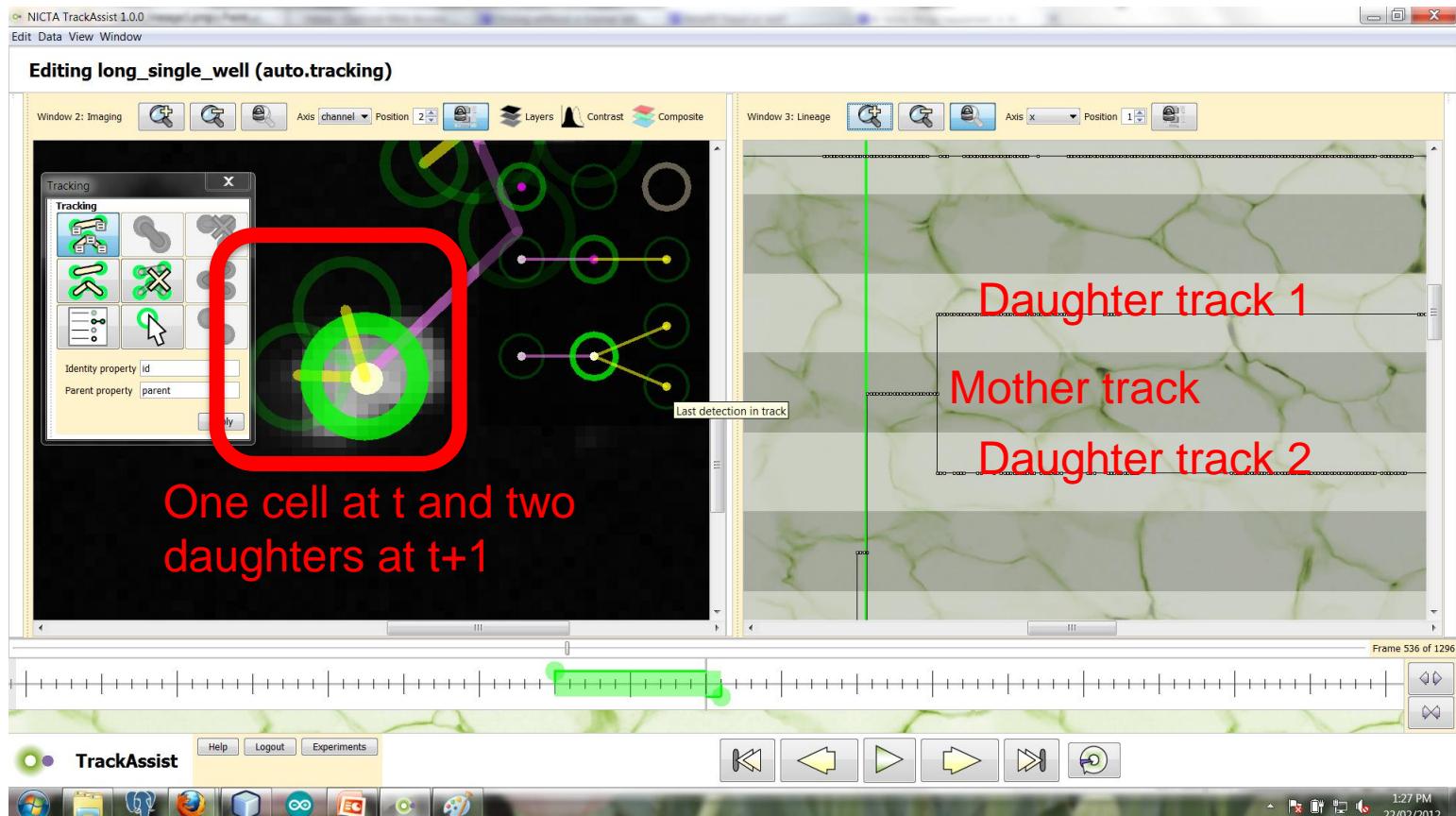
- If you zoom in you can see every individual detection in every track.
 - Click detections in the Lineage view to have them shown in the imaging view[s].
 - Double-click rows in the Lineage view to select whole tracks.



Program Walkthrough – Lineage View

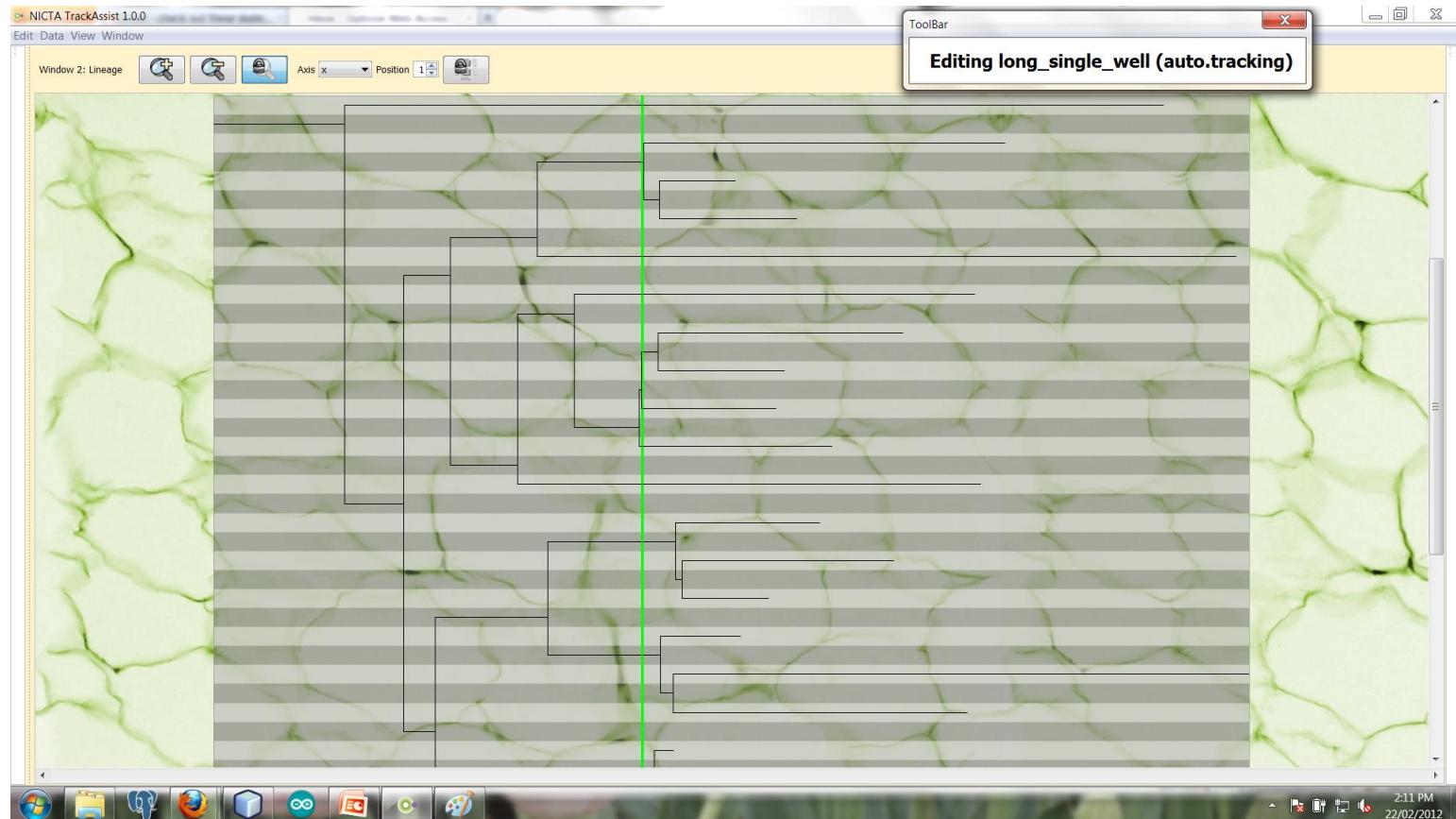


- Representing tracks and forks (cell divisions)
 - A cell division terminates the mother cell track and creates two daughter tracks.
 - These three tracks share an overlapping detection. Here you can see a division:



Typical Results

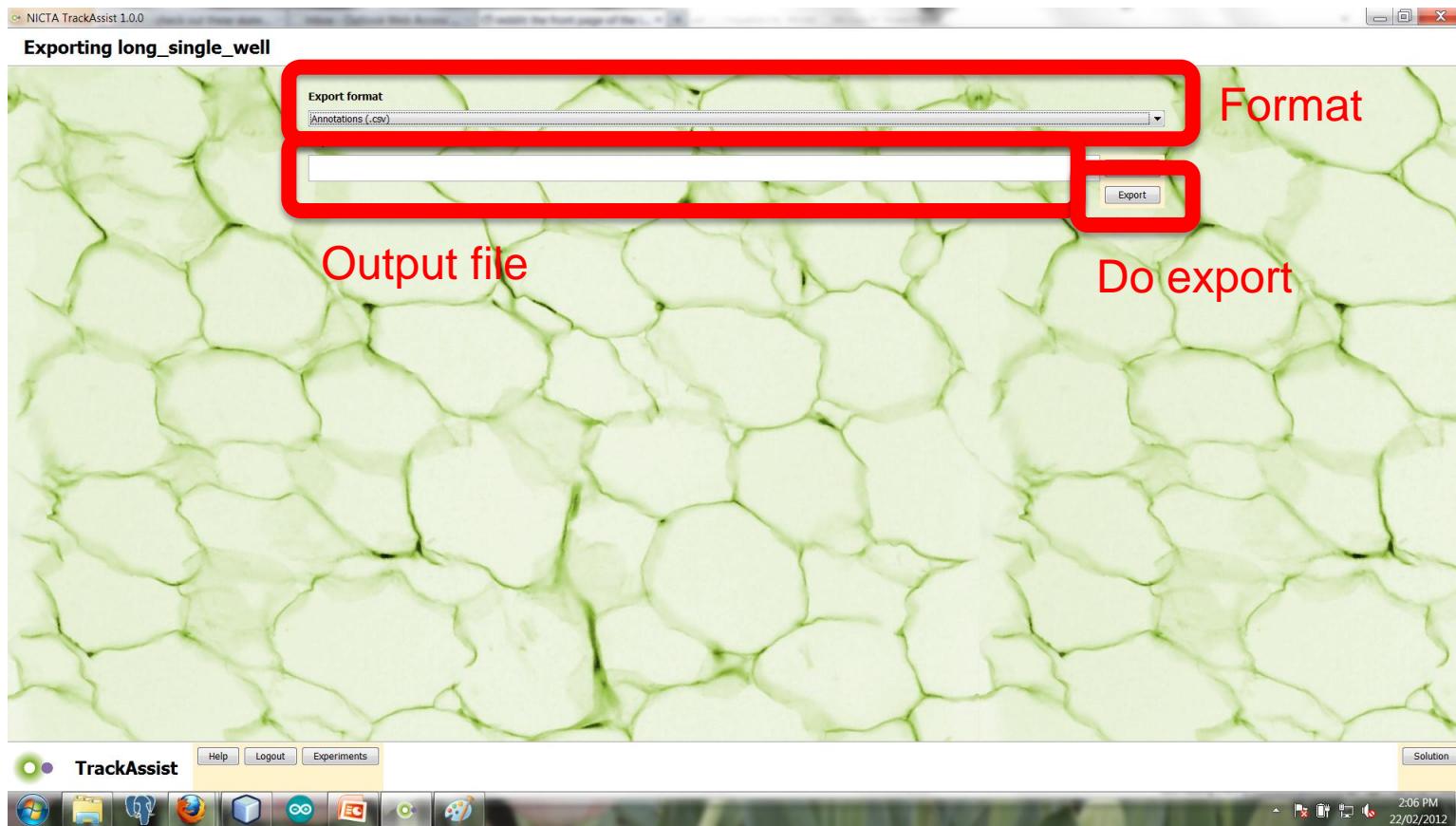
- The lineage of a microwell initially containing a single B-Lymphocyte should look something like this:



Program Walkthrough – Exporting Data



- Data can be exported through the Menus
 - Select Data→Export
 - Use the “Format” combo-box to select the type of data and set the export filename.
 - Click “Export” to actually produce the data.



Program Walkthrough – Exporting Data



- Annotations
 - Exports manually added comments / marks
 - Output file format is CSV (Comma-Separated Values), 1 line per annotation.

A screenshot of Microsoft Excel showing a CSV file named "pk_annotation". The table has columns A through I. Row 1 contains the column headers: "pk_annotation", "type", "fk_solution", "pk_image", "image_uri", "value", "x", and "y". Row 2 contains the data for the first annotation: "1", "type", "fk_solution", "pk_image", "image_uri", "value", "x", and "y". Red arrows point from the labels below the table to the corresponding columns:

A	B	C	D	E	F	G	H	I
1	pk_annotation	type	fk_solution	pk_image	image_uri	value	x	y
2								
3								

Annotations:

- ID of annotation in the database
- Type of annotation e.g. division, death, comment
- ID of the solution in Database
- ID of the image containing the annotation in Database
- Location of the image (on disk drive)
- The comment
- The x, y co-ordinate of the annotation on the image

Program Walkthrough – Exporting Data



- Detection properties
 - Properties of detection as CSV
 - NB: These files are shown as they appear in Microsoft Excel

A screenshot of Microsoft Excel showing a table of data. The table has columns labeled A through G. Row 1 contains the column headers: 'pk_entity_property', 'pk_solution', 'solution_name', 'entity_name', 'entity_pk', 'name', and 'value'. Row 2 contains the data: '1', 'pk_entity_property', 'solution_name', 'entity_name', 'entity_pk', 'name', and 'value'. Row 3 is empty. Red arrows point from the labels below the table to the corresponding columns in row 2.

A	B	C	D	E	F	G
1	pk_entity_property	pk_solution	solution_name	entity_name	entity_pk	name
2						value
3						

ID of property in the database

ID of the solution in Database

Name of the solution

Dave/Alan

Name of the property

Value of the property

Program Walkthrough – Exporting Data



- Detection statistics
 - Several statistical measures of the detections
 - Output file format is CSV

p8.A1.detection_statistics.csv - Microsoft Excel											
Home Insert Page Layout Formulas Data Review View Developer											
Clipboard		Font		Alignment		Number		Conditional Formatting		Styles	
A1	pk_detection	B	C	D	E	F	G	H	I	J	K
1	pk_detection	area	centroid_x	centroid_y	second_moment_about_x	second_moment_about_y	second_moment_lxy	bounding_box_x	bounding_box_y	bounding_box_width	bounding_box_height

ID of detection in the database

Centre of the detection

Area of detection

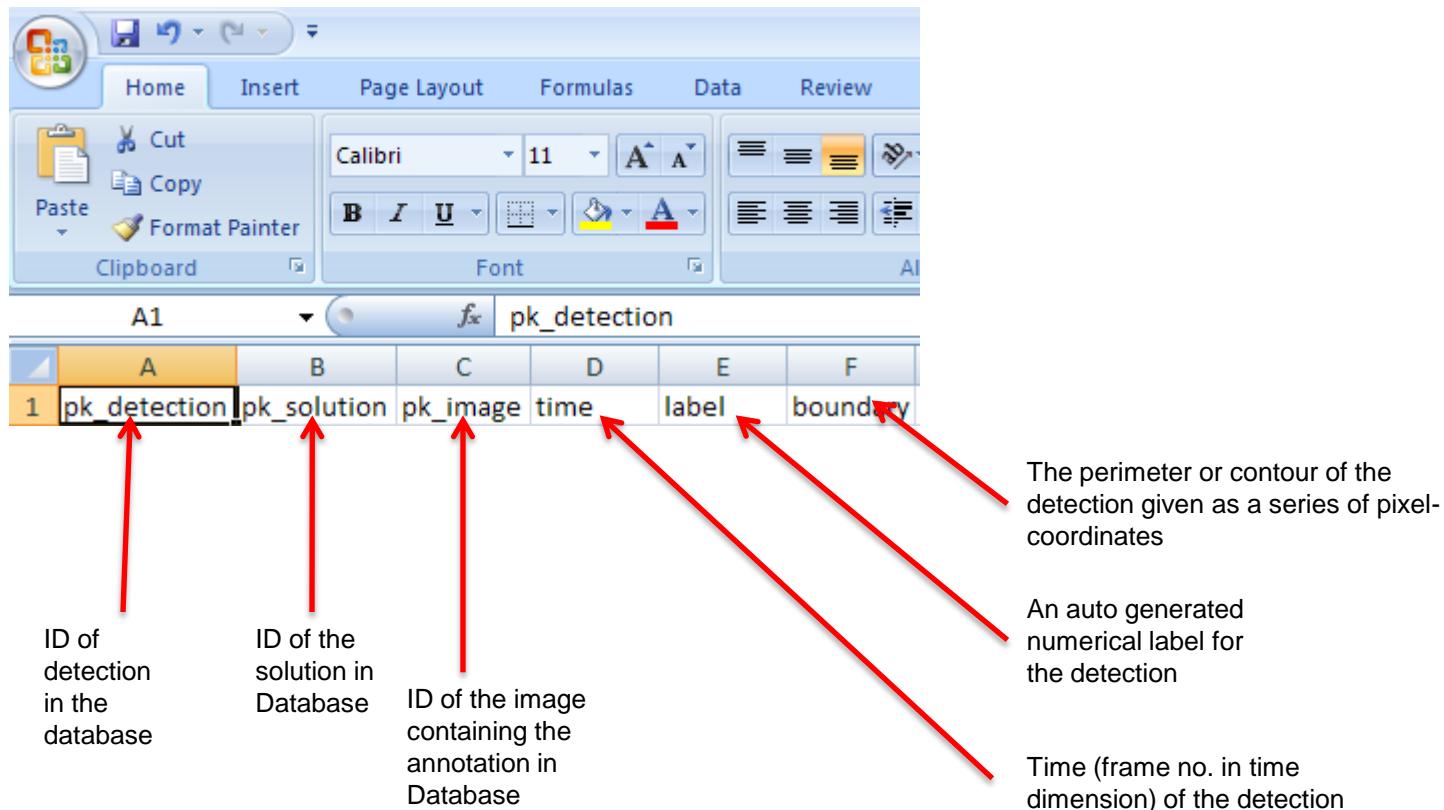
Spread of the detection in x,y

The top-left coordinate of the bounding box

Width and height of the bounding box

Program Walkthrough – Exporting Data

- Detections
 - The identified cells in the images
 - Output file format is CSV



A1	pk_detection	pk_solution	pk_image	time	label	boundary
1	pk_detection	pk_solution	pk_image	time	label	boundary

ID of detection in the database

ID of the solution in Database

ID of the image containing the annotation in Database

The perimeter or contour of the detection given as a series of pixel-coordinates

An auto generated numerical label for the detection

Time (frame no. in time dimension) of the detection

Program Walkthrough – Exporting Data



- Tracks

- The tracks of cells in the images. We provide a row in the file for each detection in each track. All the detections in each track will have the same value for pk_track
- Output file format is CSV

A screenshot of Microsoft Excel showing a table with four columns labeled A, B, C, and D. The first row contains the column headers: 'pk_track_detection' (in A), 'pk_track' (in B), 'fk_detection' (in C), and an empty cell (in D). The cell 'pk_track_detection' is highlighted with a yellow background. Red arrows point from the descriptive text below each column header to the corresponding column in the table.

D1	A	B	C	D
1	pk_track_detection	pk_track	fk_detection	

ID of the track-detection association in the database
ID of the track in database
ID of the detection associated to this track

Program Walkthrough – Exporting Data



- Track properties
 - Properties of tracks
 - Output file format is CSV

A screenshot of Microsoft Excel showing a CSV export of track properties. The spreadsheet has a single row of headers: A1 (empty), B (pk_entity_property), C (pk_solution), D (solution_name), E (entity_name), F (entity_pk), and G (name). The cell containing 'pk_entity_property' is selected, highlighted in orange. Red arrows point from descriptive text below each column header to the corresponding column in the table.

A1	B	C	D	E	F	G
1	pk_entity_property	pk_solution	solution_name	entity_name	entity_pk	name
2						
3						

ID of property in the database
ID of the solution in Database
Name of the solution
The entity name is the table being described; the entity value is a ID from that table.
Name of the property
Value of the property

Program Walkthrough – Exporting Data



- Track Events
 - Events (division, death, state-switching etc.) associated with each track
 - Output file format is CSV ordered by Time

A screenshot of Microsoft Excel showing a table of event data. The table has columns A through E. Row 1 contains the column headers: A1 is labeled 'Time', B1 is labeled 'Microwell', C1 is labeled 'Identity', D1 is labeled 'Event', and E1 is empty. Row 2 contains the data: A2 is labeled '1', B2 is labeled 'Microwell', C2 is labeled 'Identity', D2 is labeled 'Event', and E2 is empty. The 'Time' column is highlighted with a yellow background. Red arrows point from the labels below the table to the corresponding columns:

	A	B	C	D	E
1	Time	Microwell	Identity	Event	
2	1	Microwell	Identity	Event	

Time (frame no) of the event
Microwell where the event took place
The lineage of the cell. It encodes the cell generation
A text descriptor of the event (e.g. Track Split, Track End etc.)

Cells have an *Identity* computed as follows. A spontaneous creation of a cell is given a unique integer number starting with zero (e.g. 56). If this cell divides, the two children will be named 56.0 and 56.1. The division of 56.0 gives 56.0.0 and 56.0.1. (Note that Excel may not display numbers with many 0s correctly, and this file should be viewed in a text editor).

Program Walkthrough – Exporting Data



- Image shift for registration properties

- When background modelling was used to detect cells, we record the translation of images required to align the sequence. Alignment is necessary due to imprecision in positioning of the microscope stage. These registration parameters are necessary if you wish to do further work with the same images in another program.

The screenshot shows a Microsoft Excel spreadsheet titled "imageShifts.csv - Mi". The table has the following structure:

	A	B	C	D	E	F
1	SolutionPK	Solution Name	Time	Registration Value (x)	y)	

Annotations with red arrows point to specific columns:

- ID of solution in the database (points to column A)
- Name of the solution (points to column B)
- Time index (frame no) (points to column C)
- x and y translation of the image required to reproduce detection contours around the correct pixels (points to columns D and E)

x and y translation of the image required to reproduce detection contours around the correct pixels

Program Walkthrough – Exporting Data



- Labelled images

- We provide the option to export the current set of detection **contours** as a set of labelled images. Note that the way the detections were produced is irrelevant; the exported labelled contours are from the final, manually-edited set of detections.
- Detections not in tracks are also exported.
- These are produced in 16-bit TIFF format.
- An image is generated for every time point in the sequence.
- The labelled images can be very large (as large as an original sequence of images, for a single channel)
- In each image, pixel values indicate the **identity** of the segmented cells using a label value. For example, all pixels of a single cell will have the same intensity value. These values are 16 bit integers, so up to 65,000 cells can be identified.
- The purpose of this export is to allow researchers to compute custom functions of segmented cell appearance in 3rd party applications such as Matlab.

Suggested Workflow – to Detections



- Microwells
 - We need microwells for background texture (to register the images and detect motion) but they also break up the cell populations and make tracking easier.
 - We suggest you work on one microwell per solution.
- Annotations
 - We suggest the next step is that you should manually annotate the video and add comments regarding all items of interest. Annotations also help you keep track of your work (e.g. “All detections checked up to here”).
- Detections
 - We strongly recommend you perfect the detections before moving on to tracking.
 - We recommend you use the microwell feature and detect only in one microwell per solution
 - The optimal workflow is to get 90-99% accuracy using automated detection, and then fix the errors manually. Minimize time spent.
 - It’s up to you whether the cell contours are correct, the program doesn’t care

Suggested Workflow – to Detections



- Detections by Motion Detection (background modelling)
 - If you have problems getting a good background model consider these points...
 - Is the registration of the images accurate? Try increasing the registration search window.
 - Are the cells permanently occluding the background in some areas? You can use a different (shorter) time window to generate the background image, if this is the case.
 - Depending on the appearance of the cells, you may want to increase the blurring of the foreground result to merge parts of the cell together
 - You may want to vary the intensity threshold through the image sequence, e.g. every 100 frames.
 - Use the single-frame previews of each processing step to correct failures in automated detection.

Suggested Workflow - Tracking



- Using detections' properties
 - If you are using the manually defined detection properties for tracking, simply give each detection an **id** property at the start and end of each track. If the track has a parent, give the first detection in the track a **parent** property.
 - It's quick and easy to rebuild tracks repeatedly using this tracking tool. Keep checking your results as you proceed with defining detections' properties
- Using the automated tracker
 - When using this tracker it's really important that detections aren't missed and that erroneous (false) detections are not present. If the detection accuracy is very good, the tracking will also be very good
 - The tracker uses cell positions to track between images. If the cells move considerably between frames, you will need to visually verify the tracks. Occasionally there may be swaps. These can be corrected manually.
- Using the lineage view
 - The lineage view is a good sanity check for tracking accuracy. Cells don't (normally) divide twice in quick succession. Cells don't spontaneously appear. Browse both lineage and imaging views simultaneously to verify the tracks.

Planned Features

- Segmentation from constant fluorescence
 - An optional feature (currently disabled) is segmentation of cells using thresholded constant fluorescence
 - Many researchers alter cells causing them to constantly fluoresce under certain illumination. This however has the disadvantage that fewer cell state signals can be reported
- Segmentation by appearance modelling
 - We also plan to add the option to segment cells given a model of their appearance.
- Cell state detection
 - We plan to add an automated method of defining cell state using fluorescence under varying illuminations
 - Currently cell state information can be captured by annotations and by specifying detections' properties.
- Tracking improvements
 - We plan to allow the user to choose additional metrics to incorporate in automated tracking, e.g. cell size.

More information?



Business Contact:

Jia-Yee Lee

Director

Health Business Area,
National ICT Australia

jia-yee.lee@nicta.com.au

Technical Contact:

Rajib Chakravorty

Senior Research Engineer

Health Business Area,
National ICT Australia

rajib.chakravorty@nicta.com.au