Supplementary Information:

An integrated image analysis platform to quantify signal transduction in single cells

Serge Pelet, Reinhard Dechant, Sung Sik Lee, Frank van Drogen and Matthias Peter

Requirements

Operating system: Windows XP or later or MacOS 10.5 or later

Filemaker: ver. 11 or 10

Matlab: ver 2010a or later

Matlab Toolboxes: Image Processing Toolbox, Database Toolbox, CurveFitting Toolbox*, Parallel Computing Toolbox*

*These toolboxes are recommended, but the platform will work without them.

Installation

The YeastQuant platform is based on two widespread commercial softwares: Matlab (http://www.mathworks.com/) and Filemaker (http://www.filemaker.com/). Refer to the documentation of these two programs to install them.

The zip archive YeastQuantCode.zip contains the database and the Matlab code needed to perform the data analysis. The YeastQuant.fp7 file contains the Filemaker database. Open it in Filemaker to edit or create new entries. The YeastQuantCode folder contains the Matlab scripts.

An installation script is present in the Install folder to establish the link between Matlab and Filemaker (ver 11). Run the `Install` script in the Install folder in Matlab. This script performs automatically the following tasks:

- Copies the fmjdbc.jar drivers from a user specified location (Usually the Filemaker install CD (\FileMaker Pro 11 Advanced\xDBC\JDBC Client Driver Installer)) to the Install/FMconnect folder.

- Saves the database connection parameters specified by the user (File name, user name password and server name) in the DatabaseConnexionParameters.mat file in the CommonProcess folder.

- Tests the connection between Matlab and Filemaker via JDBC.

- Saves a network path where the images can be located under ImageStoragePath.mat in the CommonProcess folder.

- On a Mac, edits the Install/FMconnect/YQlink.txt file with the current locations of the YeastQuant matlab code folder and the Matlab root directory and saves it in the ~/Library/FMconnect folder under the name Link_*DATABASENAME*.command. This file can be directly called by Filemaker via AppleScripts to perform an analysis on a given set of records.

FileMaker database

The Filemaker database consists in three tables containing different types of information. The access to these tables are control by various credentials to prevent standard users from inadvertently deleting or modifying pre-existing records (Supplementary Table 3).

Experiment

The experiment table describes all the parameters of the experiment such as the date, the user, the sample and the type of experiment performed (Supplementary Figure 1). It details where the data are stored what type of microscope and software were used and what illumination setting and how many XY position where imaged. It also specifies which analysis will be performed on a given data set. A number of entries in this record with a gray shading are for documenting the experiment. They are not strictly needed for the analysis of the data. This table is connected to the Analysis table via the analysis number. Clicking on the "GoTo" button opens the linked Analysis record.

Analysis

The analysis table specifies the flow of the analysis: which images are loaded how they are segmented which secondary objects are generated and which measurements are performed (Supplementary Figure 2). The first step is to define which images will be loaded. The name of the illumination has to match the ones provided in the experiment record. Note that not all illuminations specified in the experiment record have to be loaded for the analysis. Depending on the segmentation selected a Tab describing the parameters required is active. This allows the definition of the primary objects based on the loaded images. Further routines for secondary object definitions, image correction, image measurement, display and export are defined in the lower tabs of the layout.

To facilitate the generation of the analysis record, the object and images entered in the different field are saved to create a value list. This allows the user to select directly from a pop-up list the name of the image or object to be entered in a field. In some instances these list can become over-populated. Clicking on the update list button should remove unnecessary items from those lists.

Segmentation parameters

The segmentation parameters table contains information about the size of objects to be segmented (Supplementary Figure 3). The types of settings depend on the type of image to be segmented (fluorescence or bright field) and also some image acquisition parameters such as objective magnification and binning. Each record in this database is determined by the name of the segmentation parameter specified in the analysis record and by the magnification and binning used in the experiment record. Therefore the same analysis flow can be used for two experiments with different resolutions. Only the relevant segmentation parameters will be modified according to the size of the cells in the images.

Filemaker to Matlab connection

JDBC

The information stored in the Filemaker database can be read by Matlab. To achieve this, the database toolbox from Matlab is required. This allows Matlab to connect to database via JDBC via SQL queries. To achieve this, the Filemaker database itself needs to be activated for JDBC sharing. The drivers to read the Filemaker database from Matlab (fmjdbc.jar Ver 11 or sljcinstaller.jar Ver 10) have to be added to the JavaClassPath of Matlab. This file can be found in the installation CD from Filemaker (/FileMaker Pro 11 Advanced/xDBC/JDBC Client Driver Installer).

The `querybuilder` GUI from Matlab can be useful to test if the connection to the database works. The settings for the database connection are specified in Supplementary Table 4

Note that the database can either be installed on a local machine or on a dedicated Filemaker server. In the latter case, the URL for the connection has to match the server name and port 2399 has to be open on the server.

PrepareAnalysis

Once connection between the database and Matlab is established, the `PrepareAnalysis` script can be performed. Given one or multiple experimental record number, the script will load all necessary information from the database. This information is stored in a Matlab structure where each database table (Experiment, Analysis and SegmentationParameters) are sub-structures. Each field in these tables will be a fieldname of these structures. Database fields that consist of a single entry such as `User`, `Filename` or `Microscope`, are save directly as string or numbers in the Matlab structure. However fields that can have multiple entries such as `Filter_1`, `Filter_2`, ... or `ExportMeasurement_1_1`, `ExportMeasurement_1_2`,... are converted to cell arrays (i.e. `Filter{1}`, `Filter{2}`, ...or `ExportMeasurement{1,1}`, `ExportMeasurement{1,2}` , ...).

Once the database information has been read out, the script will access the images using the provided file path to calculate how many time points were acquired for the given experiments based on the number of illumination settings and the number of XY stage positions entered in the record. This information will be saved in a `VarCell_`*`filename`*`.mat` file stored in the folder of the first experiment processed by the script

This `PrepareAnalysis` routine can be run from the command line in Matlab:

```
VarCell = PrepareAnalysis('ExpNum', [1], 'Mac');
```

Alternatively, on a Mac, the script can be run directly from the Filemaker database using Applescript called by FileMaker. If only a single experiment record needs to be analyzed, the Analyze button on that script can be clicked. Otherwise, if multiple records need to be pooled together, they can be searched by Filemaker and the script `Prepare Found Set` can be run from the Script menu. These two scripts will automatically start Matlab via the Terminal, run the `PrepareAnalysis` routine and save the `VarCell_filename.mat` file. The user can

then decide to run the analysis on it's own machine or use the `VarCell_filename.mat` file on a dedicated machine.

Matlab code

VarCell

The `VarCell` variable saved in the `VarCell_`*`filename`*`.mat` file is a cell array which contains as many entries as experimental records to be analyzed. Each one of these cells contains a structure that carry all the information to analyze one XY position independently from the others. This variable can used on any machine which has access to the image files. The limitation comes from the specific file paths required for different platforms (Mac/UNIX or Windows). The desired platform can be specified in the `PrepareAnalysis` routine. Therefore, the `VarCell_`*`filename`*`.mat` file can be prepared on a Mac and run on a windows machine or vice versa.

ParallelImgAnalysis

The `VarCell` variable is the argument passed to the `ParallelImgAnalysis` routine. This Matlab script essentially contains a loop that runs through every cell of `VarCell` to analyze all the XY positions of all the experiments loaded in this variable. Each position can be processed independently from any other. This loop can be run as a `parfor` loop, which runs every instance of the loop on a different Matlab worker. Currently, the local Matlab scheduler is used which allows a maximum of eight simultaneous workers to run in parallel. This number is sufficient for most analysis where each worker has to analyze two to three XY positions, but it could be expanded by setting up a dedicated scheduler using the Matlab distributed computing server.

PositionAnalysis

In each loop of the `ParallelImgAnalysis` routine, the function `PositionAnalysis` is run with a different cell from the `VarCell` variable which is renamed `Var`. This function performs all the task of the image analysis: image loading, segmentation, tracking, secondary object definition, image background correction and object measurements (refer to Supplementary Tables 1 and 2 for a list of the routines). Which ones of these sub-function are run and how many times they are run is determined by the Analysis data stored in the `Var` structure. For each time frame all the desired sub-functions are run and a for loop iterates over all time-points of the analysis.

Sub-functions

All these sub-functions are stored in the CommonProcess folder. They take two input: `Var` and `CallNum` and deliver as output the updated `Var`.

```
Var  = SubFunction(Var, CallNum);
```

`Var` is the structure containing all the info about the analysis and measurements and `CallNum` is an integer allowing to differentiate which instances of a variable should be taken for a different iteration of a given sub-function. As an example, the `LoadImage` sub-function is called multiple times for a given time-point to load for instance a CFP, YFP and RFP

images. The variable `CallNum` allows to differentiate the different instances when `LoadImage` is run to access each time a different set of files corresponding to specific illuminations.

Var structure

The Var structures is presented in Supplementary Table 5. This structure stores all the information relative to the analysis of the images. At each time point, the new images are loaded in this structure replacing the previous ones. At the end of the `PositionAnalysis` routine, this structure is saved to the disk under the name: `Filename_PosXX_Data.mat`. The images stored in the structure are removed from it before saving to reduce file size.

Objects definition

Objects are defined by the segmentation processes outlined in Supplementary Table 2. This information is stored in the `Var` structure as an image. In this image, pixels which don't belong to objects are set to zero. All pixels belonging to the same object share the same intensity value. This value is defined using using the `bwlabel` function from Matlab, which assigns numbers to distinct regions in a binary image starting from the upper left corner of the image.

Along to the object image, additional information about the objects is stored in the `Var.Measurement` structure. The center of mass of the object is saved in `CenterX` and `CenterY` cell arrays. The indices of the pixels belonging to an object are saved in a vector in the `PixelList` cell array. Finally, the tracking label is saved in a cell array named `Label`. In the first frame of the movie, the tracking number matches the number assigned by the `bwlabel` function. In later frames the `TrackObjects` routine will match objects from the previous frame to the frame currently analyzed based on the shortest distance between the center of mass of each object. Due to divisions, settling down, washout or improper segmentation, the number of objects will vary from frame to frame and the tracking label will thus be different than the number assigned by `bwlabel`.

As an example, in the first frame of a time lapse experiment 10 cells are identified in an image. In the second frame, a new cell appears in the top left corner of the image. It's bwlabel will be 1 and the cell previously labeled 1 will have a bwlabel of 2. However the tracking label of this cell will remain 1 and the new cell will have a get the tracking label 11. For each timepoints, these tracking labels are stored in a vector where the index corresponds to the bwlabel (Supplementary Figure 9).

Find the label of an object given its coordinates:

```
Index = Var.Img.Object(CoordX, CoordY);
Label = Var.Measurements.Label{FrameNumber}(Index);
```

Find the pixels belonging to an object based on its tracking label:

```
Index = find(Var.Measurements.Label{FrameNumber} == Label);
Pixels = Var.Measurements.PixelList{FrameNumber, Index};
```

Object Measurement

Once the object are segmented and tracked, the next step is to measure their features. Before that however, the fluorescence intensity image often has to be corrected. The "flatness", or uniformity of the illumination of the image can be corrected using a reference uniform image and the background intensity of the sample has to be subtracted to measure only the fluorescent signal arising from the sample. In the examples presented in the main paper, only the latter correction was performed. The background intensity of the image was measured by fitting a Gaussian on the histogram of the image. The mean of the Gaussian was subtracted from the raw image to obtained a corrected image which could then be quantified.

The measurements of the objects are performed in the `MeasureObject` sub-function. For a given object and intensity image, tens of measurements are calculated. Most of them are directly obtained from the `regionprops` Matlab functions but other features can easily be measured since all the intensity and locations of each pixel of the object are known. For instance, in the case of the Hog1 relocation in the microfluidic device (Figure 4), we used the `HiPix` measurement which is the average intensity of the 20 brightest pixels of an object. This usually provides a less noisy measurement than simply taking the `MaxIntensity` measurement which is the intensity of the single brightest pixel in the object.

Export structure

Once all cells in the VarCell structure have been processed by the PositionAnalysis routine and the Var files are saved to the disk. The `ParallelImgAnalysis` function combines all the data from a given experimental record in a single structure. This `Export` structure is detailed in Supplementary Table 6. Only a few number of measurements (selected in the Analysis Table of the database) are transferred from the `Var` to the `Export` variable. All the XY-positions that belong to the same experimental condition are grouped together in the same matrix enabling a faster analysis of the dataset.

Two measurements are automatically stored in the Export structure. The CheckLabel is the tracking label of each cell. If a cell was not present in a frame, the CheckLabel is set to zero. The other measurement is the `Position` which allows to verify for each cell in which XY frame the cell was found.

Access the measurement of a cell based on its XY-position and tracking label:

```
P = find(max(Export.Cond.Illum.Object.Position.Cells) == XYPos);
L = find(max(Export.Cond.Illum.Object.CheckLabel.Cells) == Label);
CellIndex = intersect(P, L);
Trace = Export.Cond.Illum.Object.Measurement.Cells(1:T,CellIndex);
```

GUIs

To facilitate the visualization and analysis of the data, two small programs with a GUI have been written in Matlab.

Image2Meas

The first one called Img2Meas allows to visualize directly the measurement performed on individual cells by selecting them in an image. The `Var` structure for a given position is

loaded by selecting the `Filename_PosXX_Data.mat` file. The user can then select to display either intensity images or objects in an RGB image. This allows the combination of three images in a single window. The individual cells can be selected in this window. The measurement to be displayed is easily selected by drop-down lists. For each cell selected the individual trace is plotted in the graph. The tracking number is displayed on the image and as a legend on the graph. It is possible to either display a single measurement or the difference or ratio of two measurements.

In Supplementary Figure 10, the data from the NaCl induced expression from Figure 3 are displayed. The RGB image combines the segmented object SplitCell in red, the YFP image in green and the CFP image in blue. Cells selected in this image have their label displayed on them and their single cell trace appears in the plot.

Export2Meas

The second GUI is called Export2Meas. It allows to load the export file and display the mean and standard error of the mean of a selected measurement. In Supplementary Figure 11, the data from the Msn2 Hog1 experiment from Figure 4 was loaded. The different conditions present in the export file can be loaded or multiple Export files from the same experiment can be combined. The conditions appear in a list and the desired ones can be selected. Using drop-down menu it is straightforward to plot the desired measurement. In the present case, the normalized Standard deviation of the SegCell object in the CorrGFPImg is plotted as well as the HiPix of the HiIntPix object in the CorrRFP image allowing to reveal the behavior of the Msn2 and Hog1 relocation events. In this experiment, two strains were combined one of them bearing a nuclear CFP marker. In this GUI using the Set Limit button, it is possible to filter cells based on various properties. In Supplementary Figure 12, Limits were set to exclude the non-tagged CFP cells. These are the 88 cells which are plotted in Supplementary Figure 10.

Supplementary Table 1. List of Matlab routines for all task except image segmentation functions listed in Supplementary Table 2.

Tab	function name	Input	output	description
	`LoadImage`	Illumination name	Image	Loads images for files
Secondary objects	`SecondaryObject`	1 Object, 1 Image	Object	Generate secondary object based on a primary object and an intensity image
	`ExpandObject`	1 -2 Objects	Object	Generate a secondary object based on a primary object using geometrical cues
Image Correction	`CorrectFlatness`	1 Image and a reference image	Image	Calculates an intensity corrected image by correcting the intensity based on a reference image for the field flatness
	`SubtractBackground`	1 Object, 1 Image	Image	Calculates an intensity corrected image by subtracting a constant value to the whole image
Measurements	`MeasureObject`	1 Object, 1 Image	Measurement	Measures properties of the object in the given intensity image
	`CalculatedObject`	2 Measurements	Object + Measurements	Calculates the ratio or difference between two measurements and generates a new object
Display	`Img2RGB`	1-3 Images, 1 Object	RGB Image	Creates an RGB image from up to three intensity images and an object
	`Img2MovieRGB`	1 Image or 1 RGB Image, 1 Object	RGB Images or movie saved to disk	Saves to disc an RGB Image with an object overlaid as movie or images
	`DisplayMeasurements`	1 Object, 1 Image, 1-5 Measurements	Plot saved to disk	Generates a plot with all the single cell traces for a measurements
Export	`CutThumbnails`	1 Object, 1 Image	Thumbnails	Generates thumbnails from all the segmented objects
	`GroupForExport`	1 Object, 1 Image, 1-6 Measurements	Export file	Combines the specified measurements from all XY positions from a single experiment record in one file
	`Export2Excel`	Export File	Excel sheet	Converts the Export .mat file in an excel spreadsheet

Supplementary Table 2. Segmentation processes and the various Matlab routines associated with them.

Name	Function name	Input	output	description
Fluo	`SegmentYeastFluo`	Fluorescence Image	Cells	Segments the fluorescence image based on intensity threshold to identify cells in the image
	`TrackObjects`	Cells		Tracks the cells
MediaFluo	`SegmentYeastFluoMedia`	Negative Fluorescence Image of the objects	Cells	Segments the negative fluorescence image based on intensity threshold to identify cells in the image
	`TrackObjects`	Cells		Tracks the cells
Nucl-Cell	`SegmentYeastFluo`	Fluorescence Image of nuclei	Nuclei	Segments the fluorescence image based on intensity threshold to identify nuclei in the image
	`TrackObjects`	Nuclei		Tracks the Nuclei
	`CellsAroundNuclHough`	Nuclei, fluorescence images of nuclei and of cells	Nuclei and Cells combined	Finds the cell around each nuclei based on a fluorescence image of the cell
Brightfield	`SegmentYeastPhase`	1-2 brightfield images	Group of cells	Segment the brightfield images based on edge detection to identify clusters of cells
	`TrackObjects`	Group of cells		Tracks the Group of cells
	`SplitGroupHough`	Group of cells, 1-2 brightfield images	Individual cells	Find single cells in groups using Hough transform
	`TrackObjects`	Individual cells		Tracks cells

Supplementary Table 3. Privilege set to access the Filemaker database

Username	Password	Permission
admin	yq_admin	full access
user	yq_user	add records, cannot modify experiments or analysis
user_edit	yq_user_edit	edit records, can modify old experiments or analysis
matlab	yq_matlab	read only, used by Matlab to connect via JDBC

Supplementary Table 4. Parameters to access the Filemaker database via JDBC

Name	YeastQuant
Driver	com.filemaker.jdbc.Driver
Username	matlab
Password	yq_matlab
URL (local)	jdbc:filemaker://localhost/YeastQuant
URL (FM server)	jdbc:filemaker://*servername.domain.ch*/YeastQuant

Supplementary Table 5. Description of the `Var` structure

`Var`	`Database`	Data to connect and perform search in the Filemaker database via JDBC. There is a single instance of the fields in these database and the `Table`, `SearchField` and `SearchStr` are replaced for each access to the different tables of the database.
	`Experiment`	Contains all the fields from the Experiment Table from the corresponding record in the database
	`Analysis`	Contains all the fields from the Analysis Table from the corresponding record in the database. Also serves as a container for all the variables created during the analysis of the images.
	`Segmentation Parameters`	Contains all the fields from the Experiment SegmentationParameters from the corresponding record in the database
	`Figure`	Contains a list of all sub-functions that require an image display. This list is created for each analysis by the `FigureFlow` routine.
	`Image`	Intensity and object images loaded or created during the analysis are stored in this structure for the current time-point analyzed. This field is removed when saving the Var structure at the end of the analysis routine.
	`Measurements`	Stores the properties of an object and the measurements performed for this objects on an intensity image
	`Error`	If the PositionAnalysis routine fails the error will be caught and added to the structure before it is saved

Supplementary Table 6. Description of the `Export` structure

<table>
<tr><td rowspan="11">Export</td><td rowspan="9">Cond1</td><td>Experiment</td><td></td><td colspan="2"></td><td>Experiment record</td></tr>
<tr><td>Time</td><td></td><td colspan="2"></td><td>Time axis</td></tr>
<tr><td rowspan="5">Illum1</td><td rowspan="4">ObjectA</td><td>MaxIntensity</td><td>Cells</td><td rowspan="4">Measurements for all cells saved in a NxM matrix where each one of the M column represents the data from a single cell containing N time points</td></tr>
<tr><td>HiPix</td><td>Cells</td></tr>
<tr><td>Area</td><td>Cells</td></tr>
<tr><td colspan="2">...</td></tr>
<tr><td rowspan="2">ObjectB</td><td colspan="2"></td><td rowspan="2">Each object measured for a given illumination</td></tr>
<tr><td>...</td><td colspan="2"></td></tr>
<tr><td>Illum2</td><td></td><td colspan="2"></td><td rowspan="2">All illumination measured</td></tr>
<tr><td>...</td><td></td><td colspan="2"></td></tr>
<tr><td>Cond2</td><td></td><td></td><td colspan="2"></td><td rowspan="2">All different conditions combined in the same export file from on experimental record</td></tr>
<tr><td>...</td><td></td><td></td><td colspan="2"></td></tr>
</table>

Supplementary Table 7. List of yeast strains

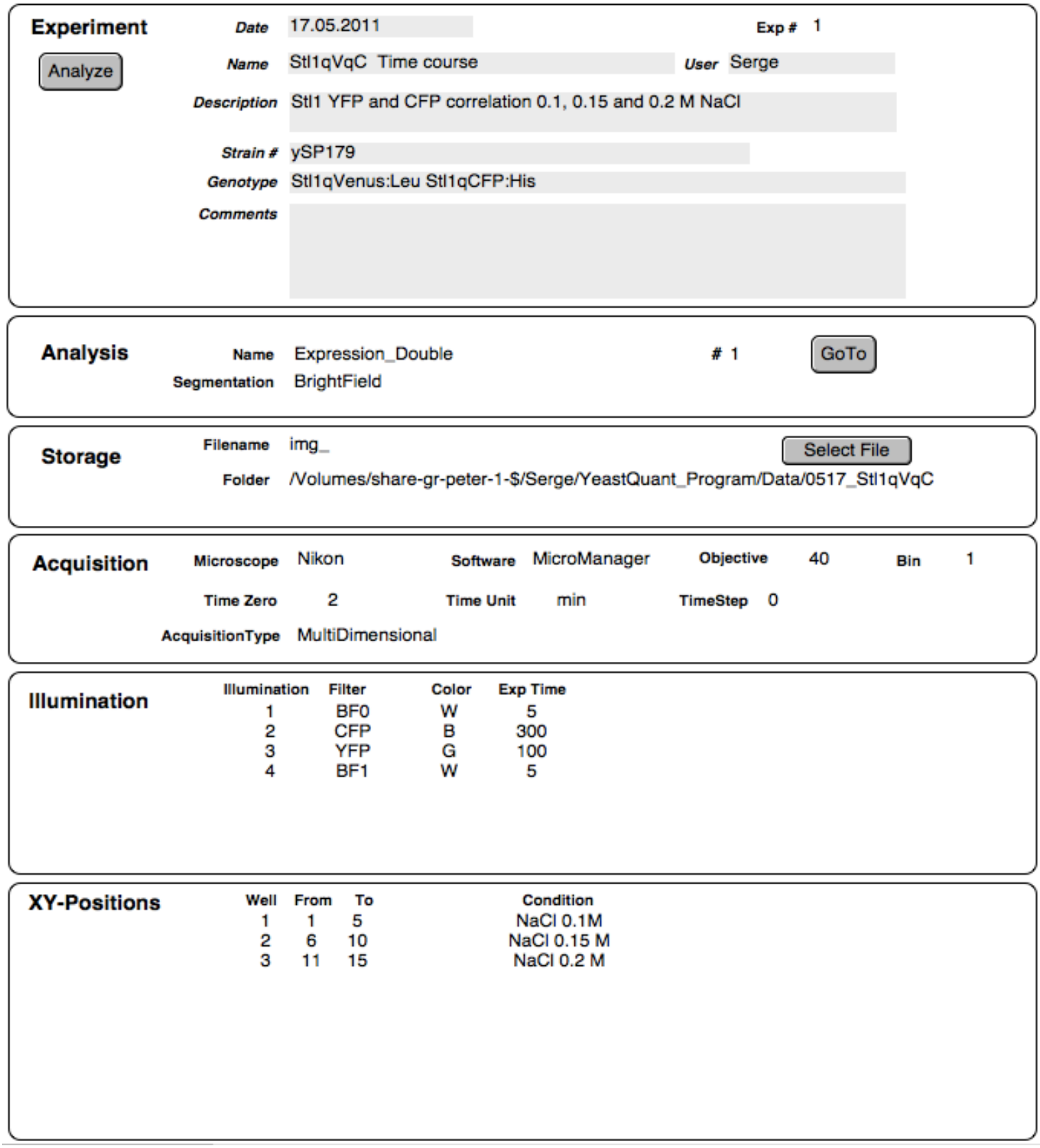
Strain	Genotype	Source /Ref
ySP2	Mata leu2-3,112 trp1-1 can1-100 ura3-1 ade2-1 his3-11,15	QUASI consortium
ySP179	Mata leu2::LEU2-p*STL1*-quadrupleVenus his3::HIS3-p*STL1*-quadrupleCFP	(1)
ySP110	Mata hta2::hta2-CFP hog1::hog1-mCherry-HIS3	this study
ySP329	Mata hta2::hta2- mCherry-URA3 hog1::hog1-GFP-HIS3	this study
ySP310	Mata hta2::hta2- mCherry-URA3 fus3::fus3-GFP-HIS3	this study
ySP331	Mata hta2::hta2- mCherry-URA3 mpk1::mpk1-GFP-HIS3	this study

1. Pelet et al., *Science* 2011, **332**:732–735.

Supplementary Table 8. List of plasmids

Plasmid	Insert	Backbone	Source /Ref
pSP83	Msn2-GFP	pRS315	(1)
pSP193	pPKI-NES- Msn2p(567-704)-GFP	pRS315	(2)

1. Görner et al., *Genes & Development* 1998, **12**:586–597.
2. Görner el al., *EMBO J* 2002, **21**:135–144.



Experiment

Analyze

Date	17.05.2011	Exp #	1
Name	Stl1qVqC Time course	*User*	Serge
Description	Stl1 YFP and CFP correlation 0.1, 0.15 and 0.2 M NaCl		
Strain #	ySP179		
Genotype	Stl1qVenus:Leu Stl1qCFP:His		
Comments			

Analysis

Name	Expression_Double	# 1	GoTo
Segmentation	BrightField		

Storage

Filename	img_	Select File
Folder	/Volumes/share-gr-peter-1-$/Serge/YeastQuant_Program/Data/0517_Stl1qVqC	

Acquisition

Microscope	Nikon	Software	MicroManager	Objective	40	Bin	1
Time Zero	2	Time Unit	min	TimeStep	0		
AcquisitionType	MultiDimensional						

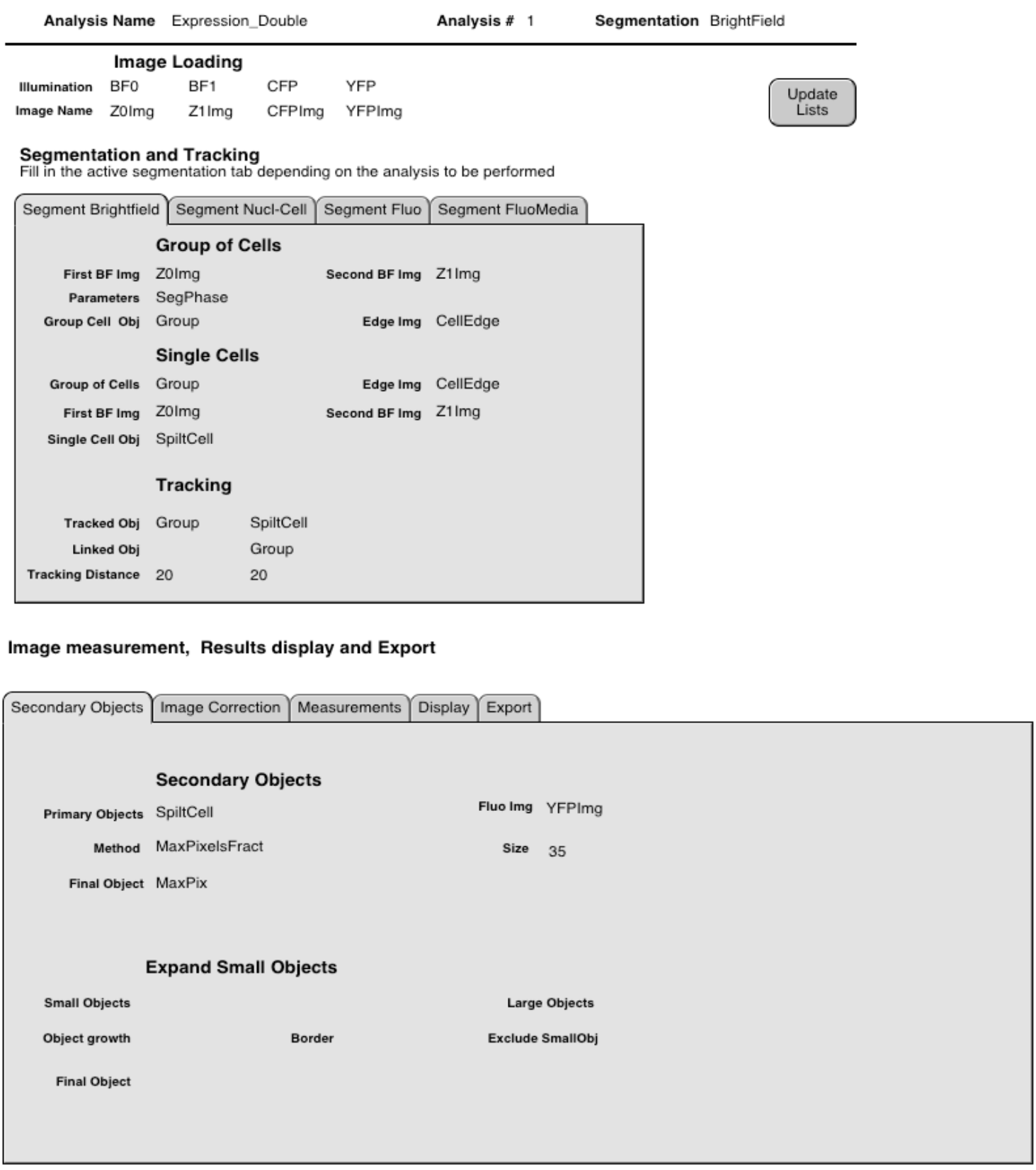
Illumination

Illumination	Filter	Color	Exp Time
1	BF0	W	5
2	CFP	B	300
3	YFP	G	100
4	BF1	W	5

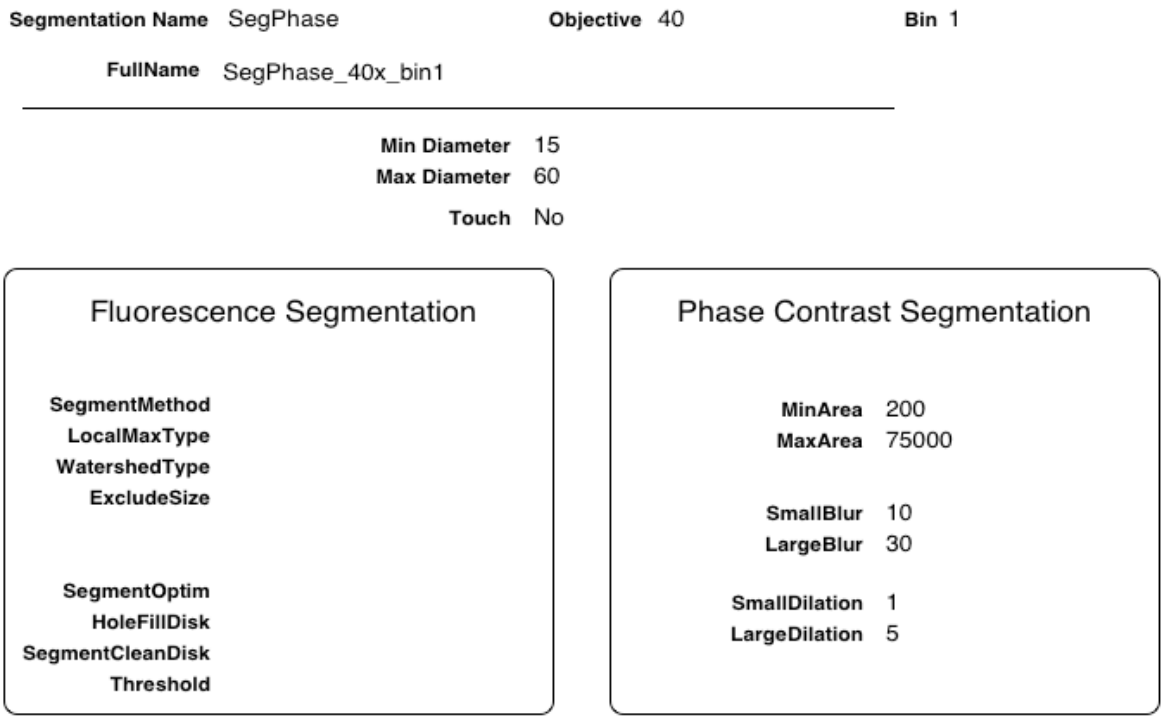
XY-Positions

Well	From	To	Condition
1	1	5	NaCl 0.1M
2	6	10	NaCl 0.15 M
3	11	15	NaCl 0.2 M

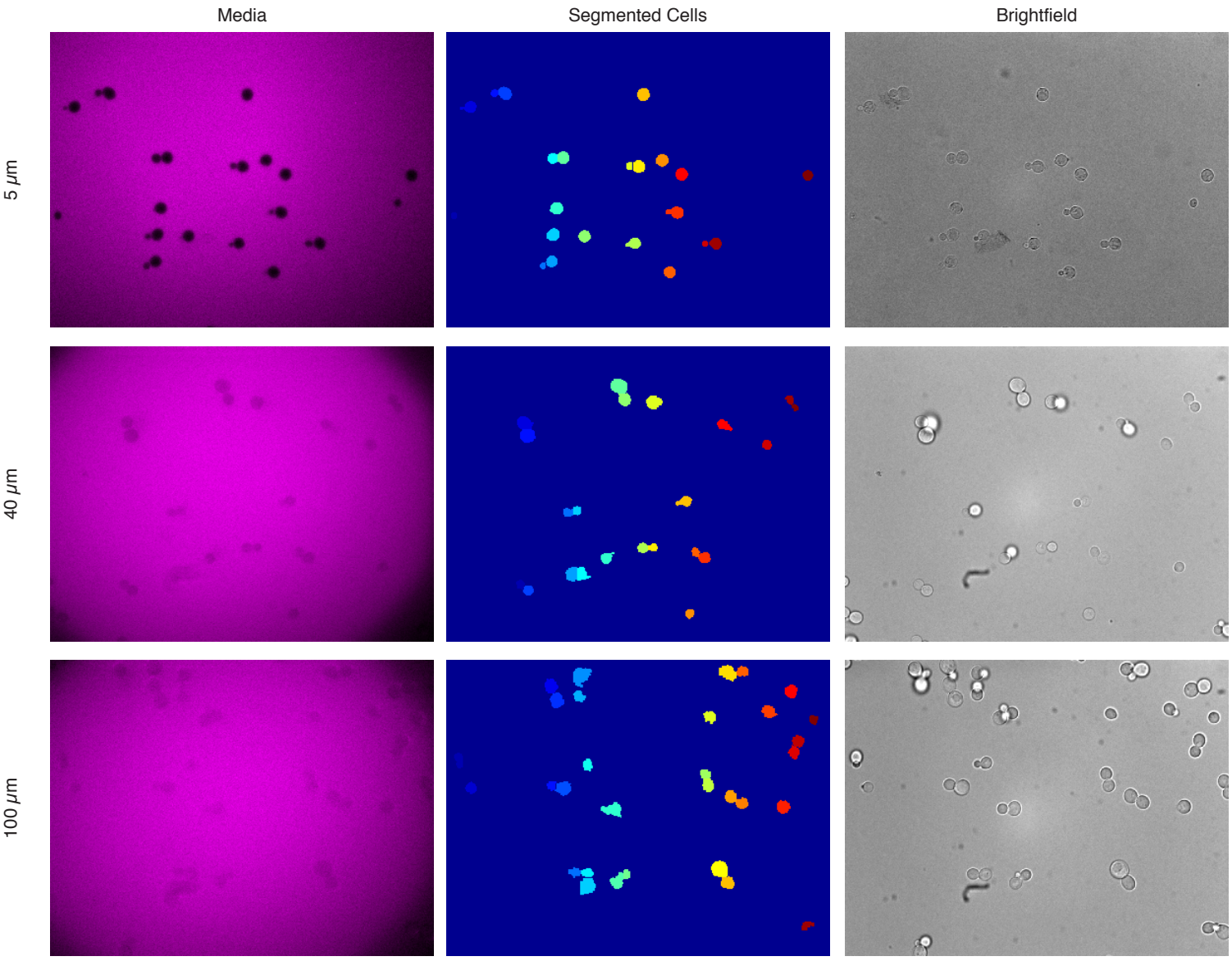
Supplementary Figure 1. Example of a Filemaker experiment table entry.



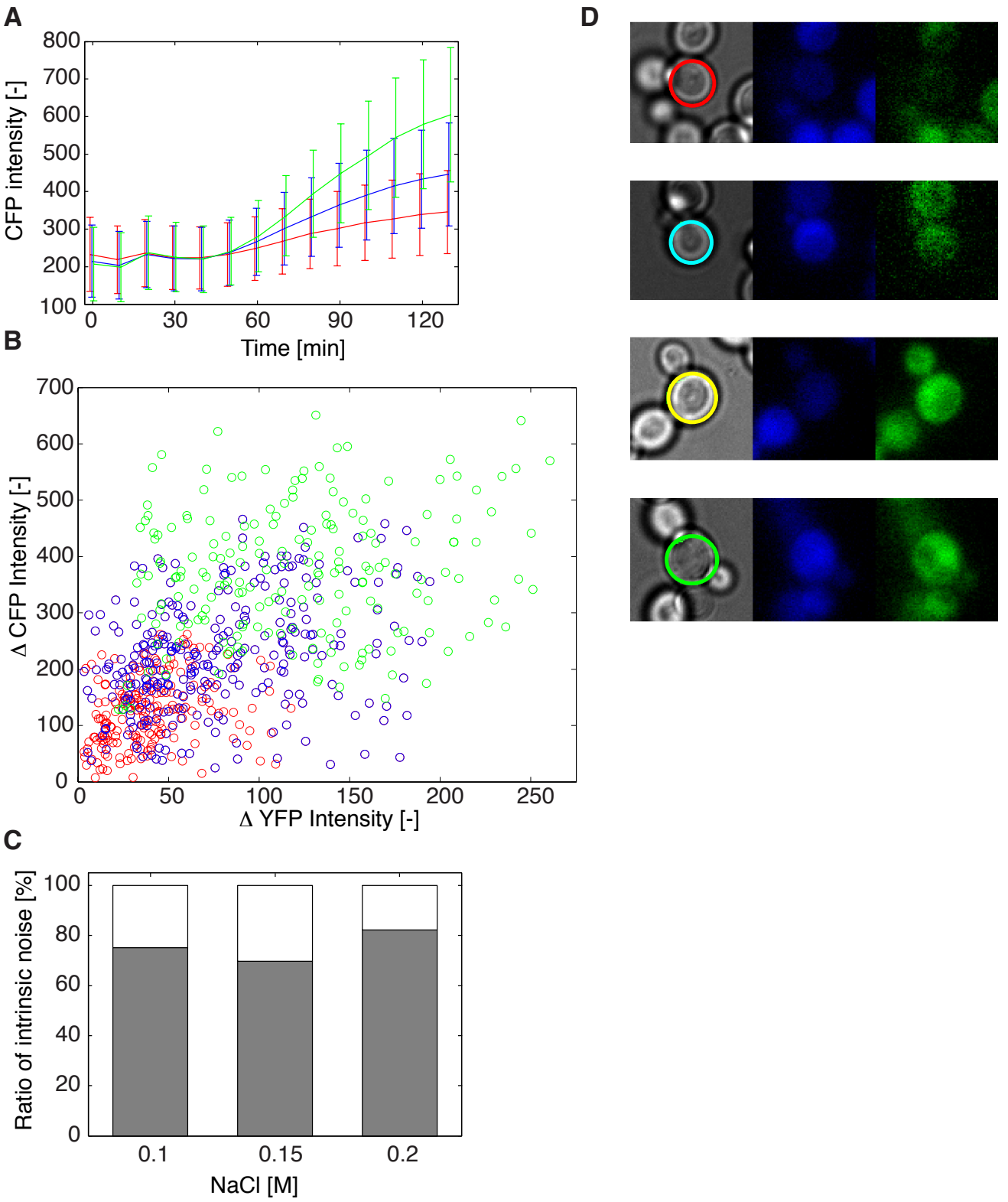
Supplementary Figure 2. Example of a Filemaker analysis table entry.



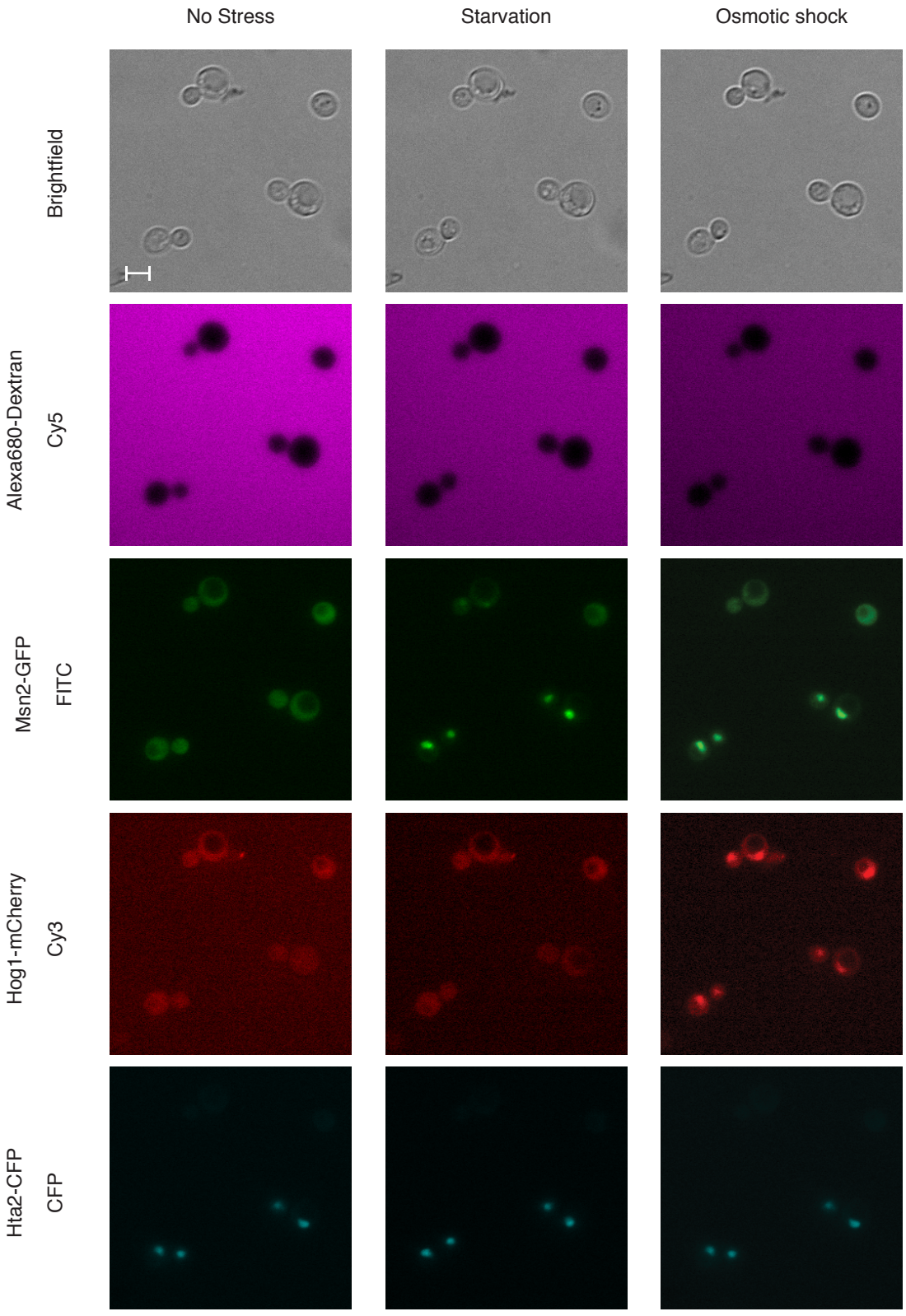
Supplementary Figure 3. Example of a Filemaker Segmentation Parameters table entry.



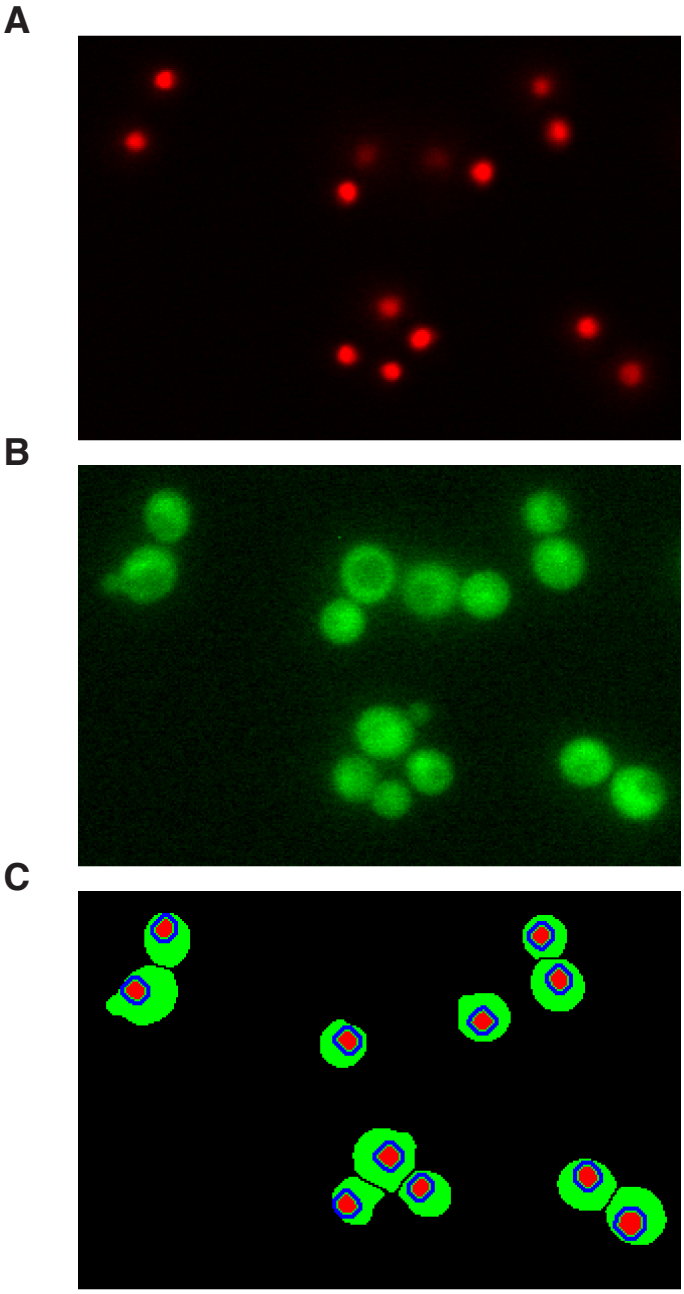
Supplementary Figure 4. Media segmentation in microfluidic channels of different thickness. The 5 μm chip are the commercial Cellasic chip. The 40μm and 100μm are built in-house using standard protocols. It is clearly apparent that the smaller thickness offers the best contrast for segmentation. The 100μm chip allow only a rough determination of the cell edges.



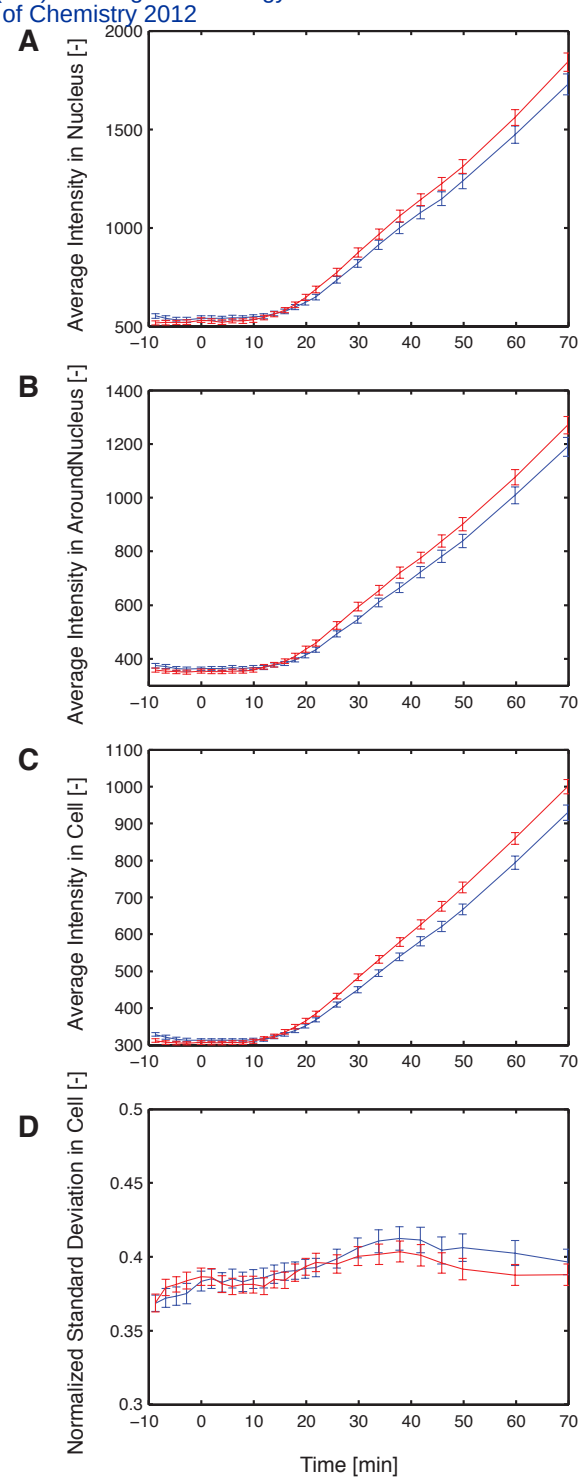
Supplementary Figure 5. Cells bearing the p*STL1*-qCFP and p*STL1*-qV expression reporters were subjected to 0.1M (red), 0.15M (blue) and 0.2M (green) NaCl. **(A)** The mean and the standard deviation in CFP intensity is plotted. **(B)** Correlation between the difference in final (2hrs) and initial fluorescence in the YFP and CFP channels at all three concentrations tested. **(C)** Ratio of intrinsic versus extrinsic noise in those three experiments. **(D)** Thumbnail images of the selected cells highlighted in Figure 3B two hours after induction. The colored circle in the bright field image corresponds to the perimeter of the cell detected by segmentation.



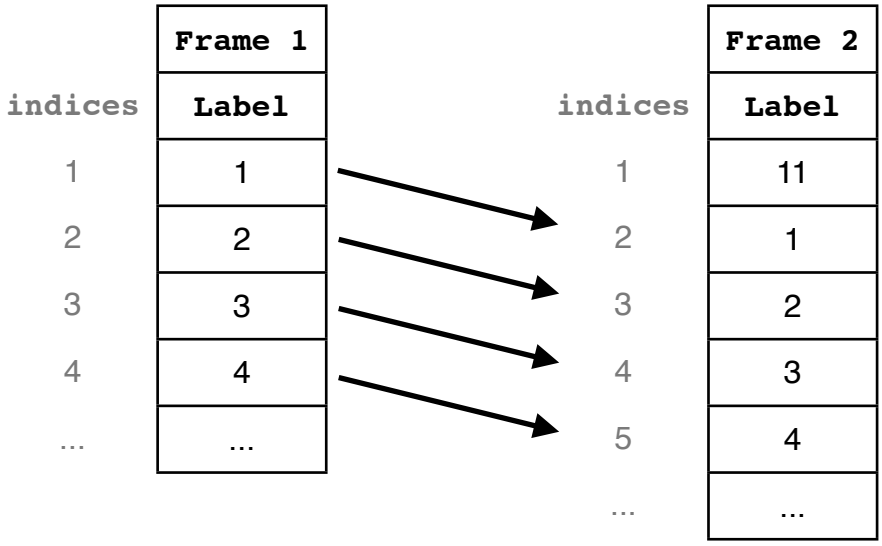
Supplementary Figure 6. Images of cells during exposed to various conditions with all the illumination channels used. Cy5: Alexa680-dextran, the average brightness of the image changes depending on the media in the flow channel; FITC: Msn2-GFP plasmids either full-length or only the NLS part responding to PKA; Cy3: Hog1-mCherry; CFP: Hta2-CFP. Scale bar 5µm.



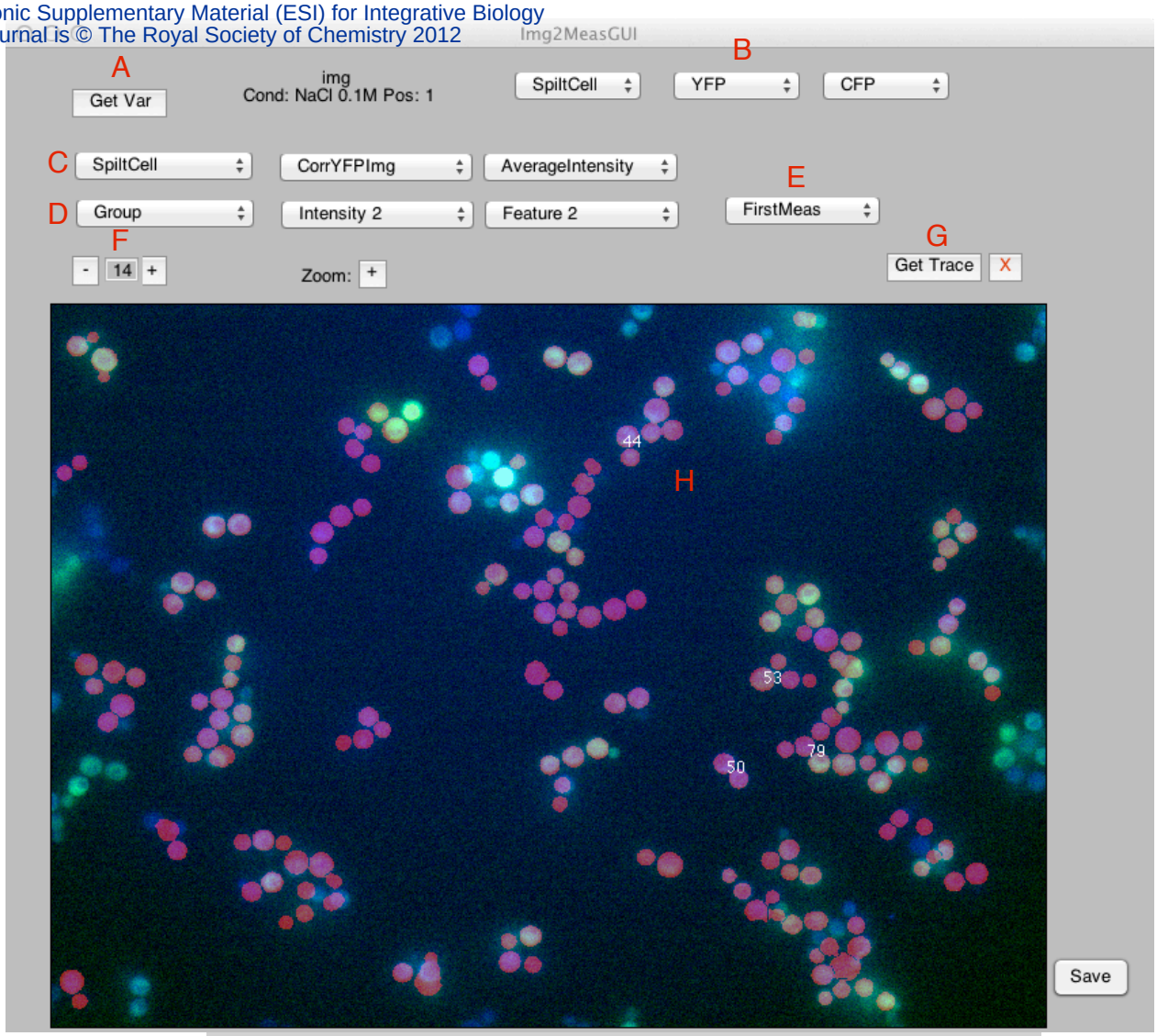
Supplementary Figure 7. A Image of nuclear stain. **B**. Image of cytoplasmic stain. **C**. Result of object segmentation and secondary object definition. In red the nucleus, in green the cytoplasm and in blue the AroundNucleus object

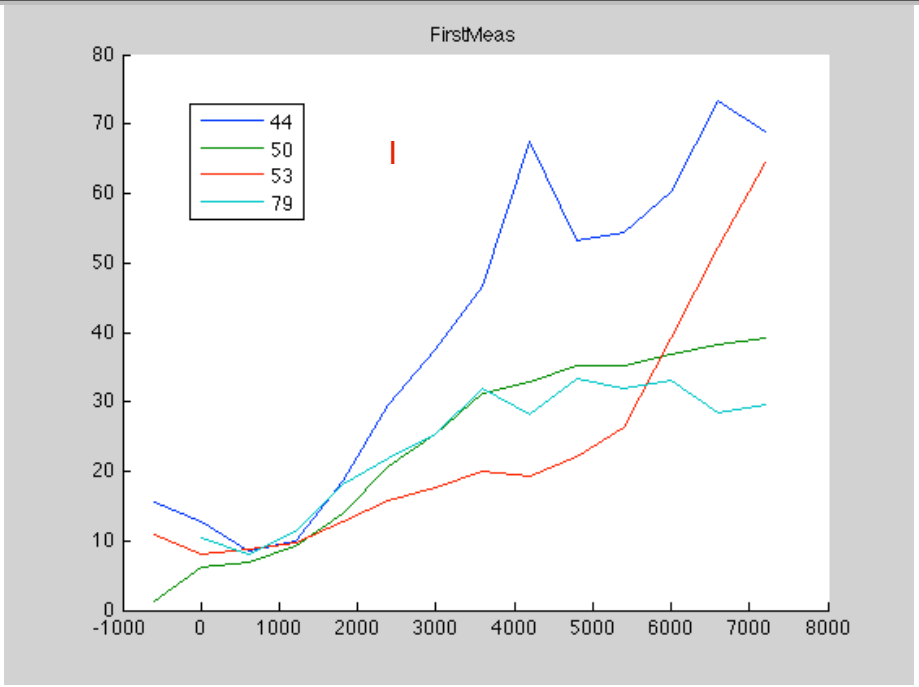


Supplementary Figure 8. Fus3-GFP characterization upon mating pheromone treatment. (A,B,C) Intensity measurements performed on the Nucleus (A), AroundNucleus (B) and Cell (C) objects. **(D)** Normalized standard deviation in the Cell object. This measurement shows greater variability than the nucleus over cytoplasmic measurement plotted in Figure 5.

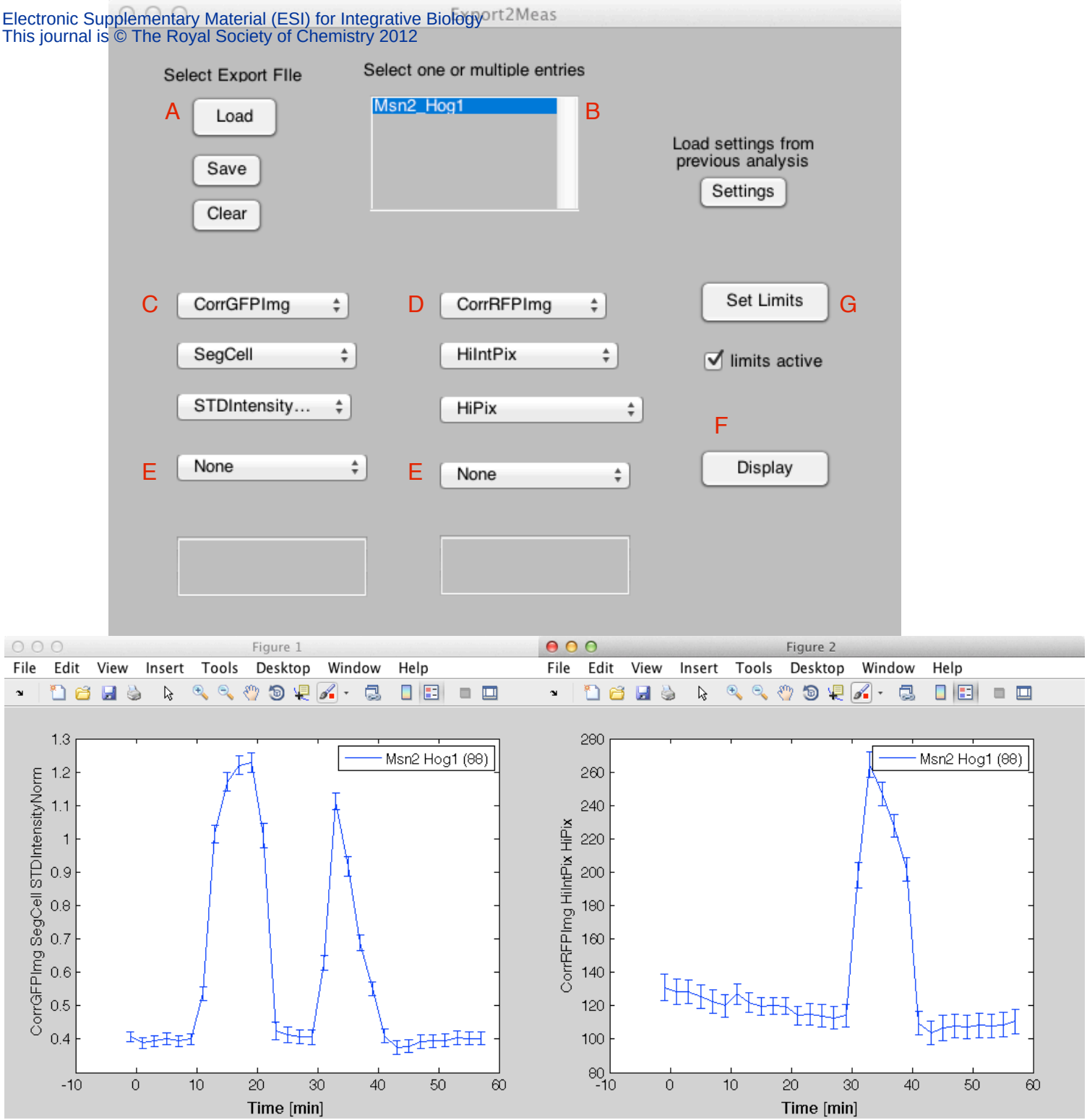


Supplementary Figure 9. Object Tracking. The Objects tracking Labels in the first frame of the movie correspond to the bwlabel assigned in the image. In the later frames, the indices of the Label vector correspond to the bwlabel

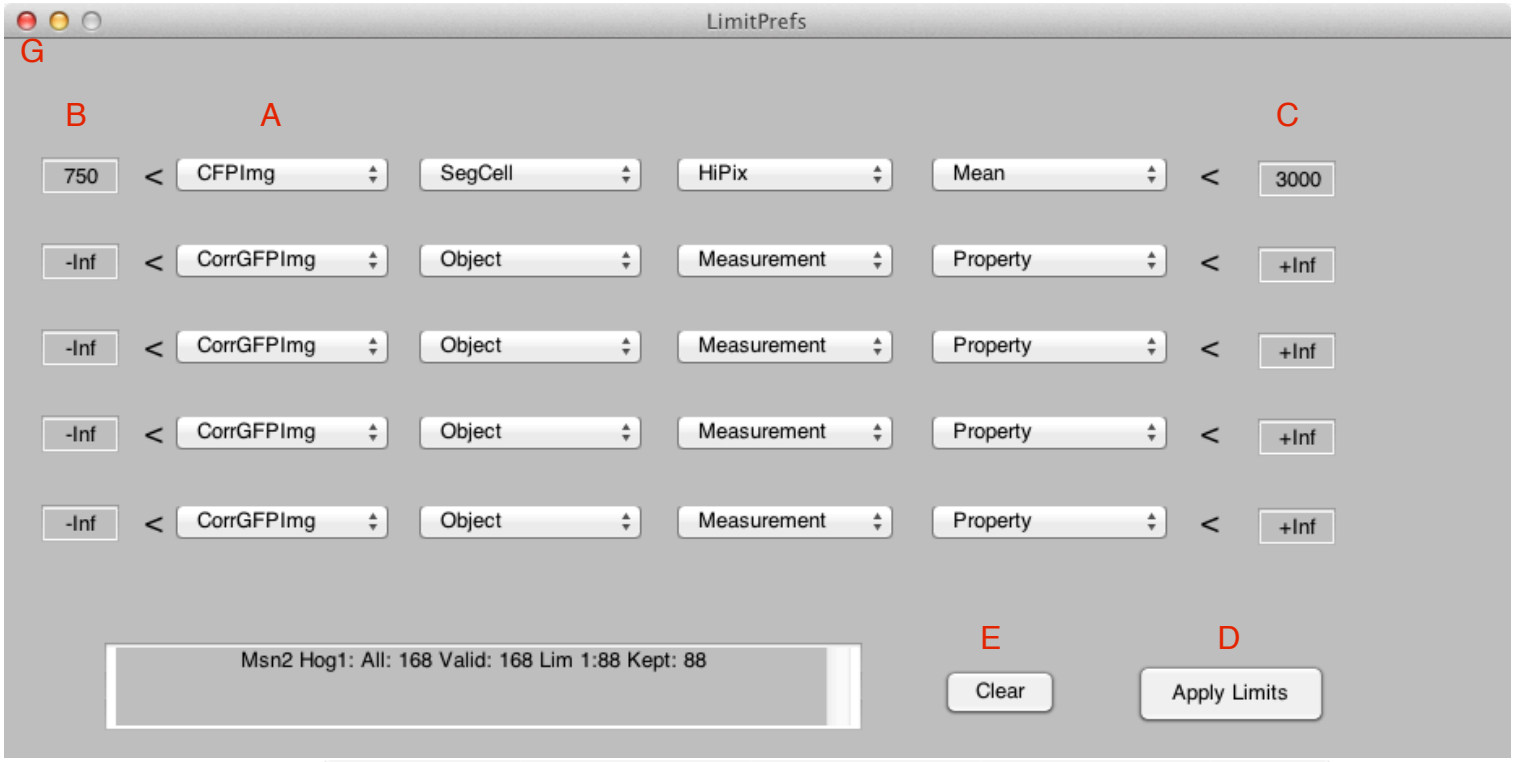


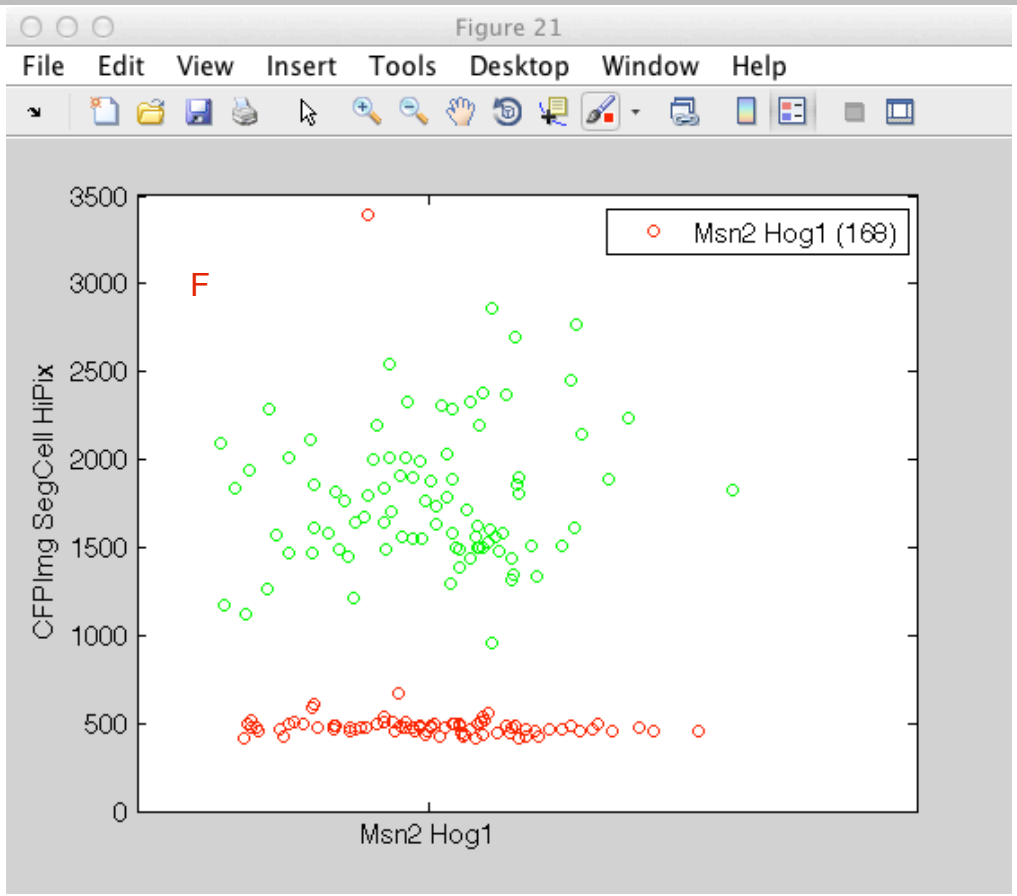


Supplementary Figure 10. Image2Meas GUI. Graphical user interface to access directly single cell traces. **A** Loading of the Var structure saved in the `Filename_PosXX_Data.mat` file. **B** Drop-down lists to select respectively the Red (SplitCell Object), Green (YFP) and Blue (CFP) images composing the RGB image displayed in H. **C** Drop-down menus to select the first measurement plotted. **D** Drop-down menus to select the second measurement plotted. **E** Drop-down menu to select if only the first measurement is plotted or if the ratio or difference is calculated between the first and second measurements. **F** Selection of time frame displayed in the image **G** Export displayed traces to the Matlab workspace. **H** RGB image. Clicking on a cell in this image displays its number and a measurement in plot I. **I** plot of single cell traces



Supplementary Figure 11. Export2Meas GUI. Graphical user interface to plot the mean and standard error of the mean of the single cell measurements. **A** Loading of one or multiple Export structure saved in the `Export_Filename.mat` file. **B** List of all conditions contained in the export file. Multiple ones can be selected for display. **C** Drop-down menu to select the First measurement plotted. **D** Drop-down menu to select the Second measurement plotted. **E** Drop-down menu to select a measurement performed on the single cell traces. This will display a scatter plot of all measurements and if two measurements are selected a graph will display their correlation as in Figure 3B. **F** Displays the desired graphs **G** Allows to set the limits on some measurements (see Supplementary Figure 12).





Supplementary Figure 12. Export2Meas GUI Limits. Graphical user interface to select which cell are plotted in the main GUI. **A** Select measurement to define limit **B and C** Set lower and upper bounds for cell selection **D** Apply the limits and display the scatter plot **E** Clear all limits settings. **F** Scatter plot of the individual cells measurements. Cells in green are kept and cells red are discarded from the plots. **G**. close the window to make the limits active on the Export2Meas GUI window.