|  |
| --- |
| King’s College London |
| Cell Tracking Profiler |
| Manual for Installation and Use |

Contents

1 Introduction to Cell Tracking Profiler 12

2 Installation (Windows) 13

2.1.1 Installing ICY 13

2.1.2 To install Cell Tracking Profiler on your computer: 13

3 Installation (Mac OS) 14

3.1.1 Installing Icy 14

3.1.2 To install cell tracking profiler on your computer 14

4 Running Cell Tracking Profiler: 15

4.1 Determining segmentation parameters 15

4.2 Running CTP 15

4.3 Track Manager 16

4.4 Data Milkshake 17

4.5 (Optional) Reformatting data 18

4.6 (Optional) Plotting in R 20

5 Appendix 20

5.1 CTP Troubleshooting 20

5.2 Tips for Track Manager 21

5.3 Sample R code 22

# Introduction to Cell Tracking Profiler

CTP is designed to analyse multi-channel time-lapsed images and is based around the freely available Icy package and the MATLAB program *Phagosight* (Henry et al., 2013). Segmentation utilises a modified Icy plugin *HK-Means*. For each cell detected during the segmentation process a Region Of Interest (ROI) is created, and information about position, shape and intensity of ROIs is saved in an excel file *using plugin ROI with Labels in Excel*. An Icy plugin launches a MATLAB executable file in Matlab Runtime to track segmented cells by *Keyhole Tracker*, based on the keyhole algorithm implemented in *PhagoSight*. Editing of tracks is accomplished by CTP sending tracks to the Icy plugin Track Manager. Combining of edited tracks and cell shape is achieved using the protocol ‘DataMilkshake’ which combines measurement of shape, intensity, displacement and speed for each track into an Excel file and folder of .csv files (Mac OS X). Post-processing of data is achieved using CTP2R, an app run in MATLAB Runtime. CTP2R allows the user to select a region of interest and write out measures of cell shape and movement as well as calculating mean squared displacement and directional autocorrelation.

# Installation (Windows)

### Installing ICY

1. Download Icy from http://icy.bioimageanalysis.org/download
2. Extract the **icy.zip** file to a **Documents\icy** folder.

### To install Cell Tracking Profiler on your computer:

1. Download the Windows 2016a MATLAB Compiler Runtime from <http://ssd.mathworks.com/supportfiles/downloads/R2016a/deployment_files/R2016a/installers/win64/MCR_R2016a_win64_installer.exe>
2. Open the **MCR\_R2016a\_win64\_installer**. Accept all conditions and install to the default location.
3. Copy and paste the folder **PhagoSightExe** in the **Users** folder.
4. Copy and paste the folder **caroff** in the folder **plugins** in the **icy** folder generated during ICY installation.
5. Copy and paste the folder **protocols** in the folder **icy.**

# Installation (Mac OS)

### Installing Icy

1. Download Icy from <http://icy.bioimageanalysis.org/download>
2. Install Icy to the general (not user-specific) **Applications** folder.

### To install cell tracking profiler on your computer

1. Download the Mac OS 2014b MATLAB Complier Runtime from <http://ssd.mathworks.com/supportfiles/downloads/R2014b/deployment_files/R2014b/installers/maci64/MCR_R2014b_maci64_installer.zip>
2. Open the **MCR\_R2014b\_maci64\_installer** folder and double-click on **InstallForMacOSX.app**. Accept all conditions and choose the **Users** folder as the installation location.
3. Copy and paste the folder **PhagoSightExe** in the **Users** folder.
4. Copy and paste the folder **caroff** in the folder **plugins** in the **icy** folder generated during ICY installation **(/Applications/icy/plugins)**.
5. Copy and paste the folder **protocols** into the folder **icy (/Applications/icy).**
6. In the *icy* folder (Applications/icy) find the app *osx-appfix* and run this to update security preferences (for Mac OS X 10.11 and above).
7. Check that Java is updated: open **System Preferences** and select the icon for **Java**. In tab ‘Update’ select option to update Java.
8. Ensure you have the latest version of the Java Development Kit (JDK) installed and if not download and install from Oracle:

https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

1. Open the Terminal program (under Applications/ Utilities) and enter the following commands:

cd /Applications/icy/

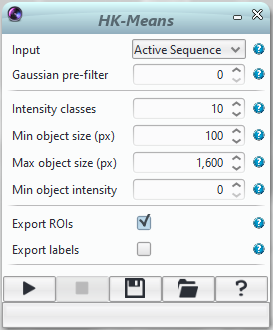
xattr -dr icy.app/

java -jar updater.jar

This will update the Java environment and launch Icy. Due to permissions restrictions in Mac OS X it will need to be performed every time the machine is restarted in order to run CTP.

# Running Cell Tracking Profiler:

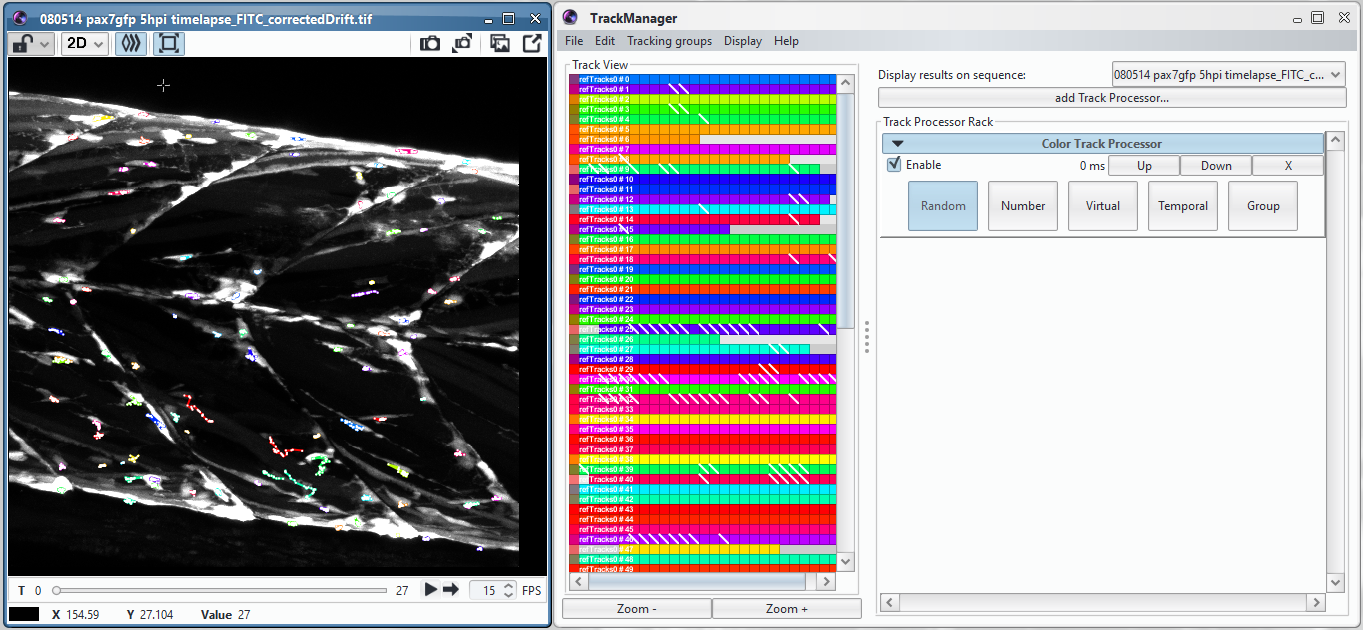
## Determining segmentation parameters

1. Launch **ICY**.
2. Open the dataset to analyse (drag the dataset from the file explorer onto the ICY panel).
3. Use the ICY search bar to open the **HK-Means** plugin.
4. Choose initial parameters for **Intensity classes**, **Min object size** and **Max object size**.
5. Launch the process by clicking on the **Run** (play symbol) button, and adjust the parameters until the segmentation fits the data optimally.
6. If there is more than one channel in the image to analyse, segmentation parameters should be determined for each channel separately.

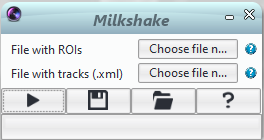
## Running CTP

1. Use the **ICY** search bar to open the **Protocols** plugin.
2. In the **Protocols** window, select **Load**, and choose the *CellTrackingProfiler.protocol* file saved in the icy/protocols folder.
3. In the first block (**Folder**), click on **Choose folder name…** and select the folder where the dataset to analyse, is saved (each dataset should be in its own folder).
4. In the next block (**File**), click on **Choose file name…** and select the dataset for analysis.
5. Check that the value of **Frame** in each **My hierarchical K Means** is **-1**. This tells HKMeans to process all timepoints.
6. Input optimal values for **Intensity classes**, **Min object size** and **Max object size** for each channel as determined in 4.1.
7. If your image has more than one channel, the segmentation parameters for each **My hierarchical K Means** block (up to 3 channels in total) must be entered.
8. Click on **Run** button. The process is running!
9. If an error message appears “*Error in block “Extract Channel”: Parameter “channel”: Channel index must be between 0 and 0 / 1*” **ignore it.** This appears for single channel images and provided the user does not acknowledge the error message the *Keyhole* *Tracker* program will start in MATLAB Runtime. The program is finished when the MATLAB icon is no longer displayed on the dock and message “*phagosight: File .xml generated*” appears in the output terminal of Icy.
10. The protocol will produce several outputs. Here X denotes a number 0-2 depending on how many channels were segmented and tracked e.g. If three channels are processed then three files will be produced at each point, Channel0\_..., Channel1\_.. and Channel2\_..
    * A file called *ChannelX\_ROI+Label.xls* which contains an identifier, positional information and measures of cell shape. One for each channel if more than one channel is present.
    * A folder termed *ChannelX* (and Channel1, Channel2, depending on how many channels were segmented in the previous step). This contains individual .tiff files representing segmented objects and an .xml file containing parameters of the image files for each time-point.
    * A track .xml file called *ChannelX\_tracks.xml.*  One for each channel if more than one channel is present.

## Track Manager

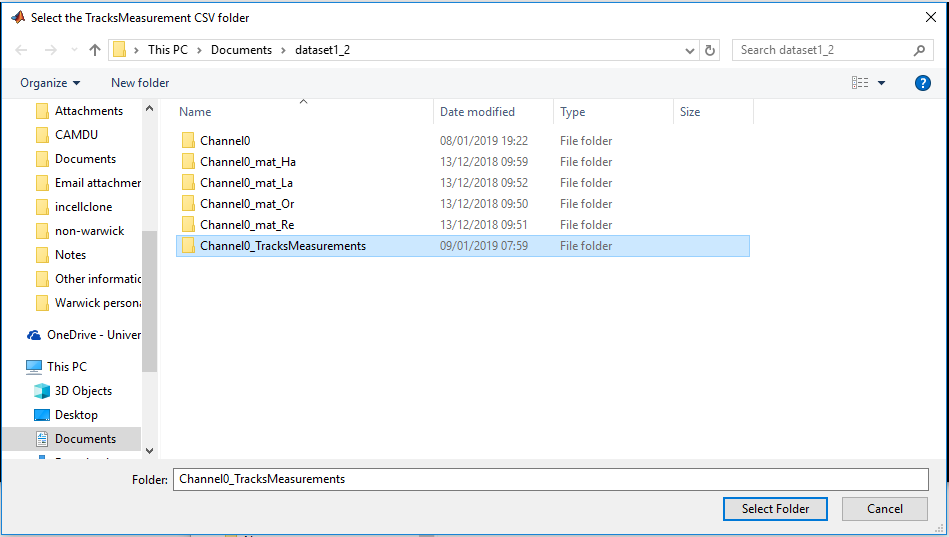
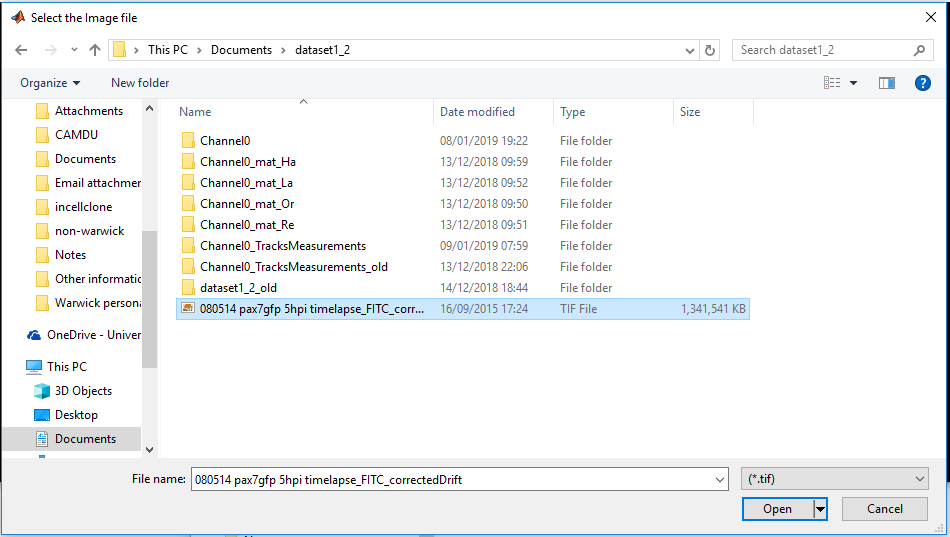
1. **Track Manager** is an ICY plugin that can be used to view and correct the tracks generated by Keyhole Tracker.
2. Open the **Track Manager** plugin and the image if not already open.
3. Use **File -> Load** on Track Manager to select the *ChannelX\_tracks.xml* file generated by Keyhole Tracker.
4. Choose the image as the sequence with **display results on sequence**.
5. Adjust tracks as required (see Appendix for hints and tips on using Track Manager).
6. Save the corrected track file (.xml).

## Data Milkshake

1. Now the corrected tracks and segmented cell data can be combined.
2. Use the ICY search bar to find and open the **Milkshake** plugin.
3. Select the *ChannelX\_ROI+Labels.xls* file in the dialogue box labelled **File with ROIs**.
4. Select the corrected track .xml file in the dialogue box labelled **File with tracks**.
5. Run Milkshake by pressing the ‘play’ button.
6. A file called *ChannelX\_TracksMeasurements.xls* is generated. This contains an initial summary sheet and then the measured parameters for each tracked cell are saved in a separate worksheet.
7. Additionally, a folder of .csv files is also generated, with each .csv file being derived from a worksheet of file *ChannelX\_TracksMeasurements.xls.* This is useful for further formatting, especially in non-Windows OS.

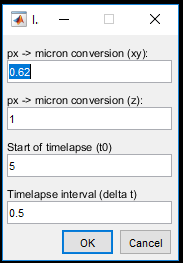
## (Optional) Reformatting data

1. CTP2R is a MATLAB executable that can be used to select a region of interest (ROI) and provides output files allowing processing of CTP outputs for plotting in R using ggplot2.
2. Run the executable by clicking on **CTP2R.exe** (Windows) or **CTP2R.app** (Mac) in the relevant folder. Screenshots shown below are taken from a Windows computer but are relevant for Mac OS too.
3. After a short load time, the program will ask first to navigate to the relevant image file (the original dataset CTP was run on) and then to the TracksMeasurements folder of csv files. CTP2R must be run separately for each channel.

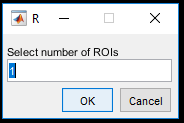


1. Input the relevant imaging parameters when prompted. These are:

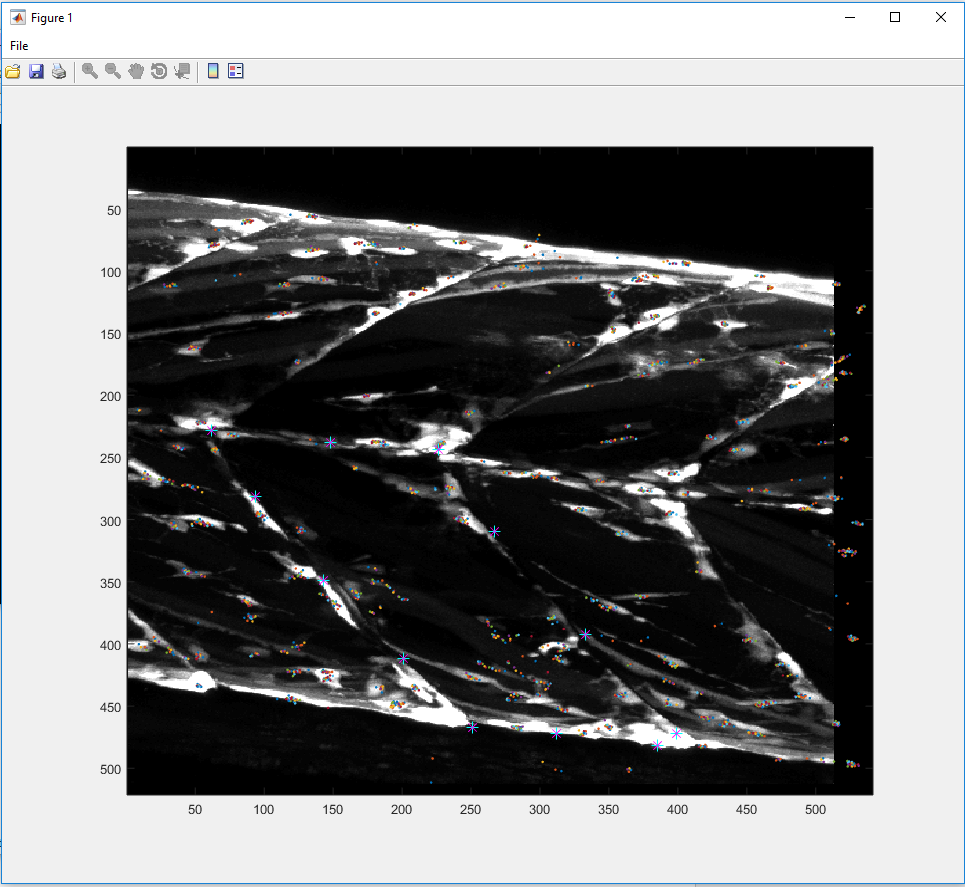
* *px -> micron conversion (xy)* (defines the pixel size for xy in µm)
* *px -> micron conversion (xy)* (defines the pixel size for z in µm)
* *t0* (defines the initial timepoint if not 0 e.g. if a timelapse starts several hours post-manipulation and this varies for each timelapse you can adjust t0 so that the manipulation occurs at t0 each time)
* *delta t* (defines the time interval between frames)



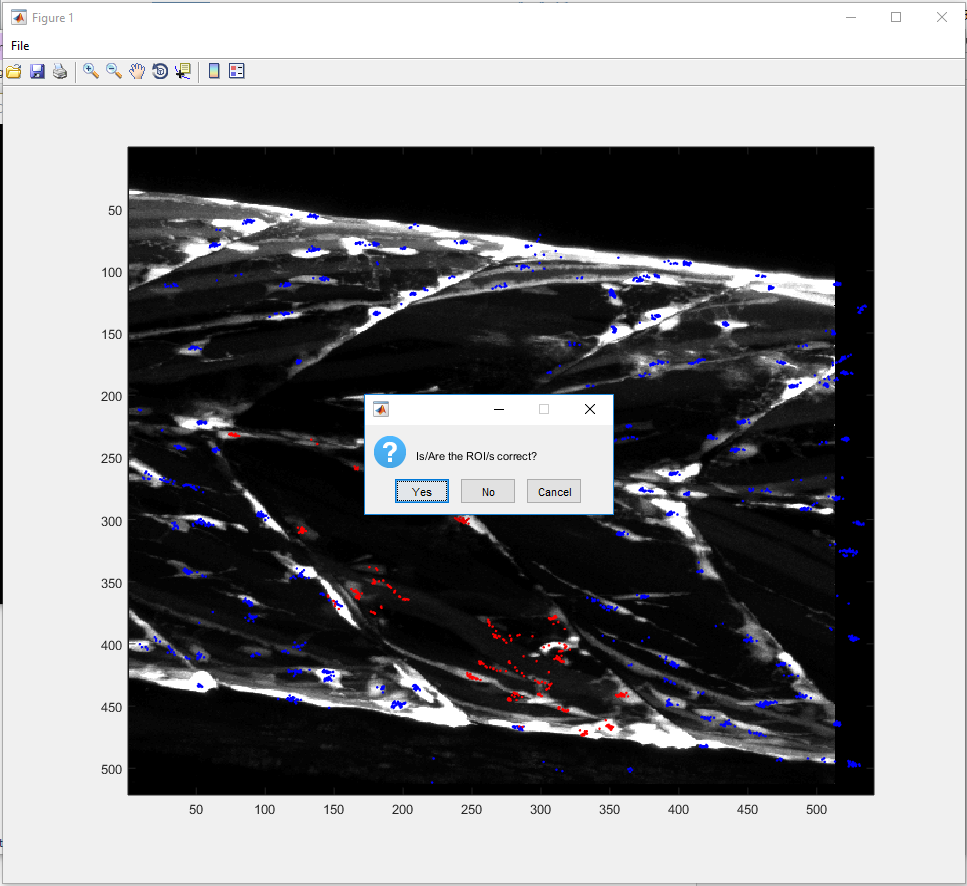
1. Input the number of desired ROIs when prompted (from 0-2 allowable).



1. If an ROI number or 1 or 2 has been entered, the program will then display a maximum intensity projection of the image stack, overlaid with track data for freehand selection of an ROI.



1. Click around the perimeter of the desired ROI, once an ROI is drawn, a double-click or return key will either continue the program or allow selection of another ROI if 2 ROIs were requested.
2. The program will then display cells within the ROIs in red and/or green and the cells outside the ROIs in blue. The blue cells will still be included in the output files as ROI label 0. The user is prompted to confirm if the ROIs are correct, selecting “*Yes*” will continue the program, selecting “*No*” will open up the ROI selection image again.



1. The program will then create two files called *<filename>\_ggplot.csv* and *<filename>\_ggplot\_mean.csv* which can be used in Section 4.6 for generating comparative plots of your CTP data.

## (Optional) Plotting in R

1. Install and open **RStudio**.
2. Import the data using **File -> Import Dataset -> From CSV** or using the readr library (see section 5.3 for an example).
3. See the appendix section 5.3 for sample R code.

# Appendix

## CTP Troubleshooting

1. If you are running the CTP protocol on less than three channels then you will get an error message “*While running block “Extract Channel”: Parameter “channel”: Channel index must be between 0 and 0 / 1*”. Wait until the Icy output window shows “Lecture ok => \_mat\_Or” and then click continue. Clicking continue too early may abort the tracking process.
2. While running the CTP protocol, you will see the warning “Label not found for xxxx (ROI line)” printed in the Icy output, this occurs when CTP cannot match a label to an ROI and is expected for a few ROIs within a dataset.
3. If you are running the CTP protocol on a Mac, you may see the error “*No Info.plist file in application bundle or no NSPrincipalClass”* in the Icy output. In older Mac OS (10.11 and below) you can fix this by running the osx-appfix file in the Icy folder, this will launch Icy and the launched version should not show this error. If you are running newer Mac OS (10.13 and above) then you will need to run the following command lines in the Terminal:

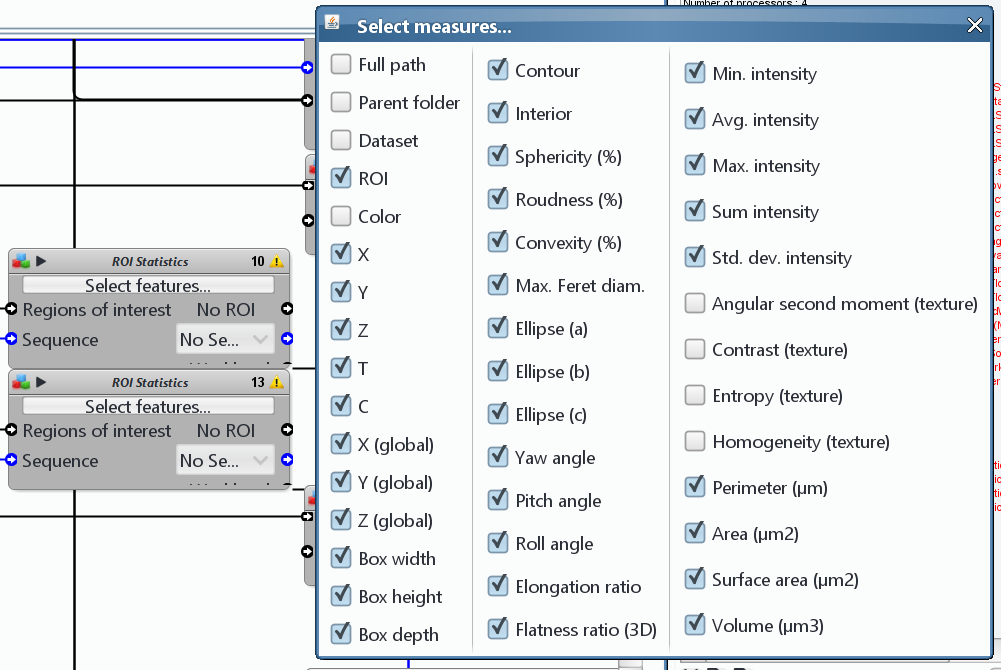
cd /Applications/icy/

xattr -dr icy.app/

java -jar updater.jar

These commands will need to be rerun after each Java update. In our experience this is needed every time we need to restart Icy and wish to use CTP.

1. Occasionally, the CTP protocol will not output every desired parameter, this can affect downstream processing. This problem can be confirmed by opening up the *ChannelX\_ROI+Labels.xls* file. If the following parameters shown as ticked in the figure below are not present then you may not be able to run Milkshake or CTP2R correctly. This can be rectified by clicking and unclicking the missing parameters, resaving the protocol file and running it again.



1. While running Milkshake, you may see a message “*ERROR: format of tracks is unadapted*”, this is a warning rather than an error and should not affect the operation of Milkshake.

## Tips for Track Manager

* To connect two tracks, simply drag from one block on a track to another. If you are happy with the connection, select Fuse All track-segments to fuse the connection into one track.
* To split incorrect connections, connect a track at the point of the incorrect connection. You can then use “delete links on selected tracks” to then delete the undesired connection.
* **Save frequently**, it’s very easy to accidentally delete a track and there is no undo.
* It is usually sufficient to plot the tracks on a maximum intensity projection (use the plugin Intensity Projection on the image) for correction, this makes things easier to view. It is only when two cells are on top of each other that the full 4D view is required.
* The Track Processors **Time Clip** and **Z Clip** are indispensable when tracks are dense and overlapping. You can use these to display only certain parts of tracks, selecting by time and Z position.
* Colour the tracks by number (effectively time) using **Color Track Processor**, this allow you to see more readily tracks that might need connecting as earlier tracks are coloured green, through to pink and red for later tracks.
* One of the downsides to Track Manager is that you can only connect tracks that are both visible on the Track View. Zooming out on Track View allows you to see more tracks on your screen.

## Sample R code

#This will plot the final directionality value using the “\_mean.csv” spreadsheet generated by CTP2R as a separate violin plot for each ROI with mean and standard deviation error bars and significance values

#Load required libraries

library(ggplot2) #plotting library

library(readr) #excel reader

library(ggpubr) #significance values

#Read in Excel spreadsheet

dat <- read\_csv(“insert path to spreadsheet here”)

#defines a function to plot the mean and standard deviation

data\_summary <- function(x) {

m <- mean(x)

ymin <- m-sd(x)

ymax <- m+sd(x)

return(c(y=m,ymin=ymin,ymax=ymax))

}

#defines the ROIs labels and colours for plotting

lab1 = c("control", "ROCKout", "y-compound")

val1 = c("black", "red", "blue")

lab2 = c("control", "control", "blebbistatin", "blebbistatin")

my\_comparisons <- list(c("0","1"), c("0,2"), c("0,3"), c("1","2"), c("1","3"))

ggplot(data = dat, aes(x = roi, y = directionality)) + #set the data and x and y aesthetics

geom\_violin(aes(color = roi)) + #plot a violin plot coloured by ROI

theme\_bw() + #simple plotting theme

ylab('Directionality curve') + xlab('') + #set the x and y axis labels

stat\_summary(fun.data=data\_summary, aes(color = condition)) + #add the mean and standard deviation error bars

scale\_x\_discrete(limits = lab1, labels = lab2) + #label the x axis

scale\_colour\_manual(values = val1) + #colour the plots

stat\_compare\_means(comparisons = my\_comparisons) #add significance values (if desired)