User-guide for analysis of the cell wall regeneration post-protoplasting to obtain single plant cells

Q-Warg

Procedure overview and quick guide

Here we describe step by step how to operate the image analysis workflow to quantify cell wall regeneration after protoplast extraction to help the user to screen quickly and efficiently CW regeneration media. This workflow uses different components under the form of imageJ (Fiji) macro and R scripts.

1- Organize your files with one condition in one folder containing the label image and the original image only.
2- Run the macro CWRegenerationQuantification.imj to get quantification information from the images.
3- Run the R script Q-Warg-analysis.R to get one table with 1 row = 1 cell and plots to analyze the CW regeneration. The table is also usable to plot the data differently.
4- Run the R script Q-Warg-interactive-folder.R to reorganize the folders and be ready to use the shiny app.
5- Run the R script Q-Warg-shinyapp.R to check the data: observe the cells corresponding to the quantitative data.

See below for the detailed guide.

Prerequisites:

Software & plugins required:

- Fiji (https://fiji.sc/)
    - MorphoLibJ (“IJBP-Plugins” update site)
- RStudio (https://posit.co/download/rstudio-desktop/)
    - Libraries:
        - data.table
        - rlist
        - hrbrthemes
        - viridis
        - tidyverse
        - shiny
        - plotly
        - magick

Workflow scripts:

- CWRegenerationQuantification.ijm
- Q-Warg-analysis.R
- Q-Warg-interactive-folder.R
- Q-Warg-shinyapp.R

Data:

- Segmented images from the brightfield image of the cell suspension. Here we used CellPose (http://www.cellpose.org/) to segment our images.
- The original image in TIFF with the 3 channels:
    - Brightfield
    - Viability staining
    - Cell wall staining

Installation

1- To use the imaging macro

To install Fiji/ImageJ, see https://fiji.sc/

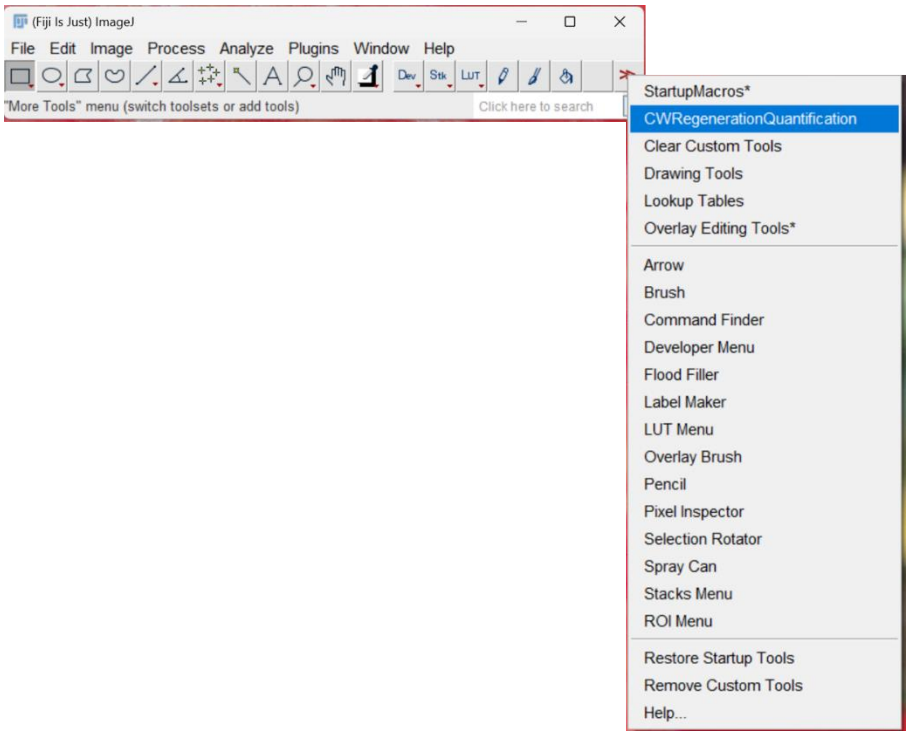
To install the plugin MorphoLibJ, turn on the IJBP-Plugins:

1- Go in the "Help" menu of Fiji and click on "Update...". This will open the "ImageJ updater".
2- In "ImageJ updater", click on "Manage update sites".
3- In the list, find e.g. "IJPB-Plugins" and click in the square next to it to add it.
4- Then close the window and apply changes on "ImageJ updater".
5- Once this is done, Fiji needs to be restarted.

If more explanation needed, see https://imagej.net/update-sites/following

Click on around the top right corner of the page, and then "download zip". Then, unzip the file in the folder of your choice. Finally, copy the CWRegenerationQuantification.ijm file and past it in the macros/toolsets folder of your Fiji install folder (on a Mac, access this by right clicking on the Fiji app in a Finder window and selecting "Show Package Contents").

To check if the toolset was loaded properly, open Fiji, and click on at the right end of the Fiji window. You should see CWRegenerationQuantification in the drop-down menu. Select it and the toolset should appear in your Fiji toolbar.



2- RStudio

Find the installer fitting your system here: https://posit.co/download/rstudio-desktop/

To install the different packages needed to run the different scripts, you can run the following in the console of RStudio:

```
# Define the packages you want to install
packages <- c("data.table", "rlist", "hrbrthemes", "viridis", "tidyverse",
"shiny", "plotly", "magick")
# Find out which packages are already installed
installed_packages <- rownames(installed.packages())
# Determine which packages are not installed
packages_to_install <- setdiff(packages, installed_packages)
# Install the packages that are not yet installed
if (length(packages_to_install) > 0) {
  install.packages(packages_to_install)
} else {
  cat("All packages are already installed.\n")
}
```

The code is also available in GitHub “Install_R_packages”. The code checks if the required packages are installed and add the ones that are not. For more information on how to install packages, see https://cran.r-project.org/doc/manuals/r-release/R-admin.html#Default-packages .

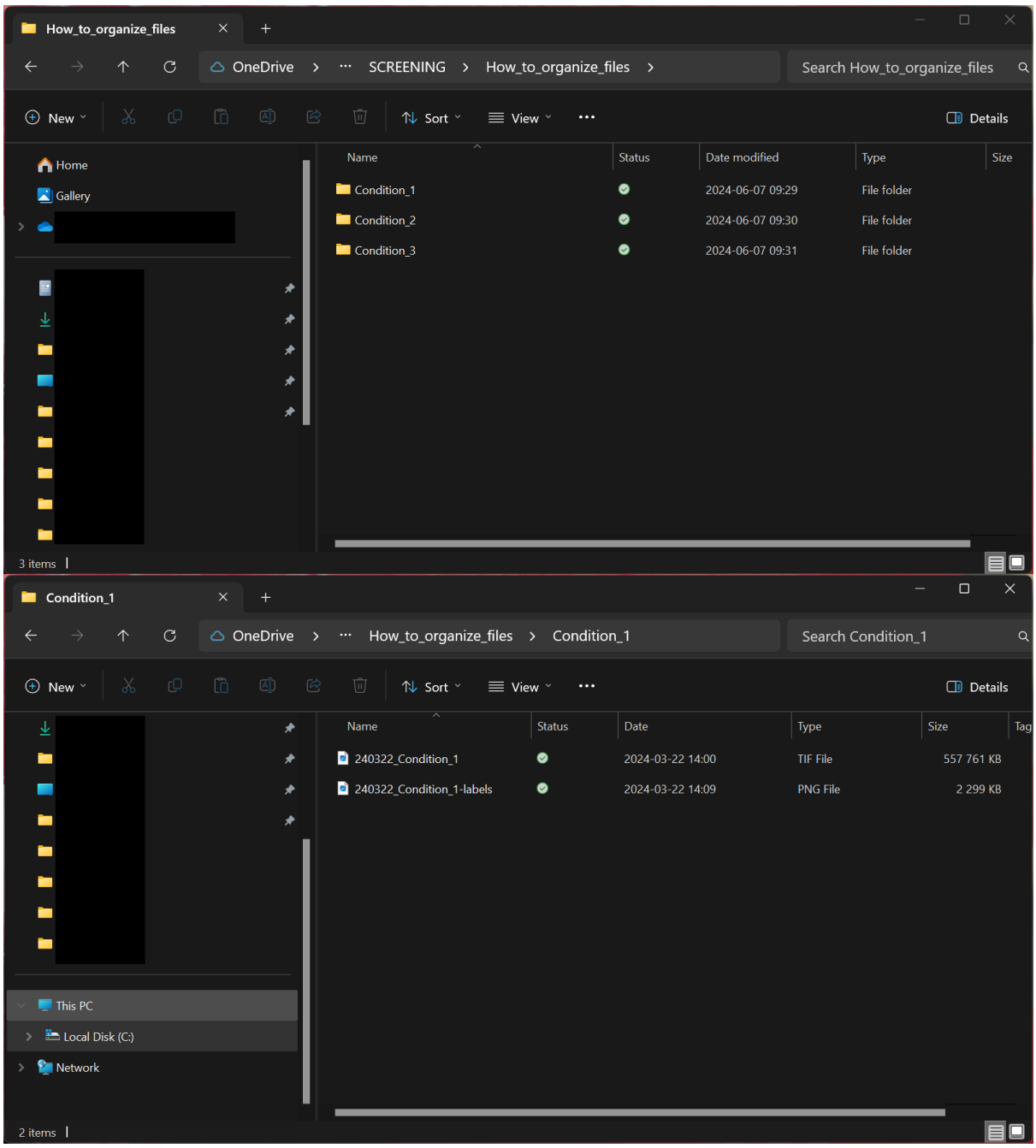
The R scripts are available on GitHub to download. Once RStudio and the scripts are downloaded, double-click on the script to open it with RStudio. It is then ready to use.

Q-Warg: the pipeline

After image acquisition, we advise keeping a separate folder with all the raw untouched data.

1- Data organisation

The data with which to use Q-Warg should be organized as follows:

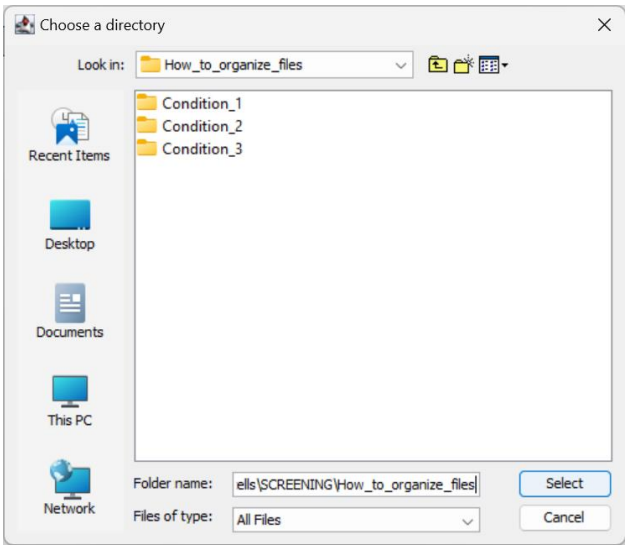


Here, the working folder is called “How_to_organize_files”. In each “Condition_n” subfolder (sample folder) should the TIFF image with the 3 channels saved and the segmented label image in PNG.

Note: If you want to run the macro on only one condition, you need to put your sample folder into a working folder. The macro will report an error if it is running directly in the sample folder.

2- CWRegenerationQuantification (Fiji macro)

When starting the macro, a pop-up window opens. You can choose the working folder. In the example here “How_to_organize_files”.

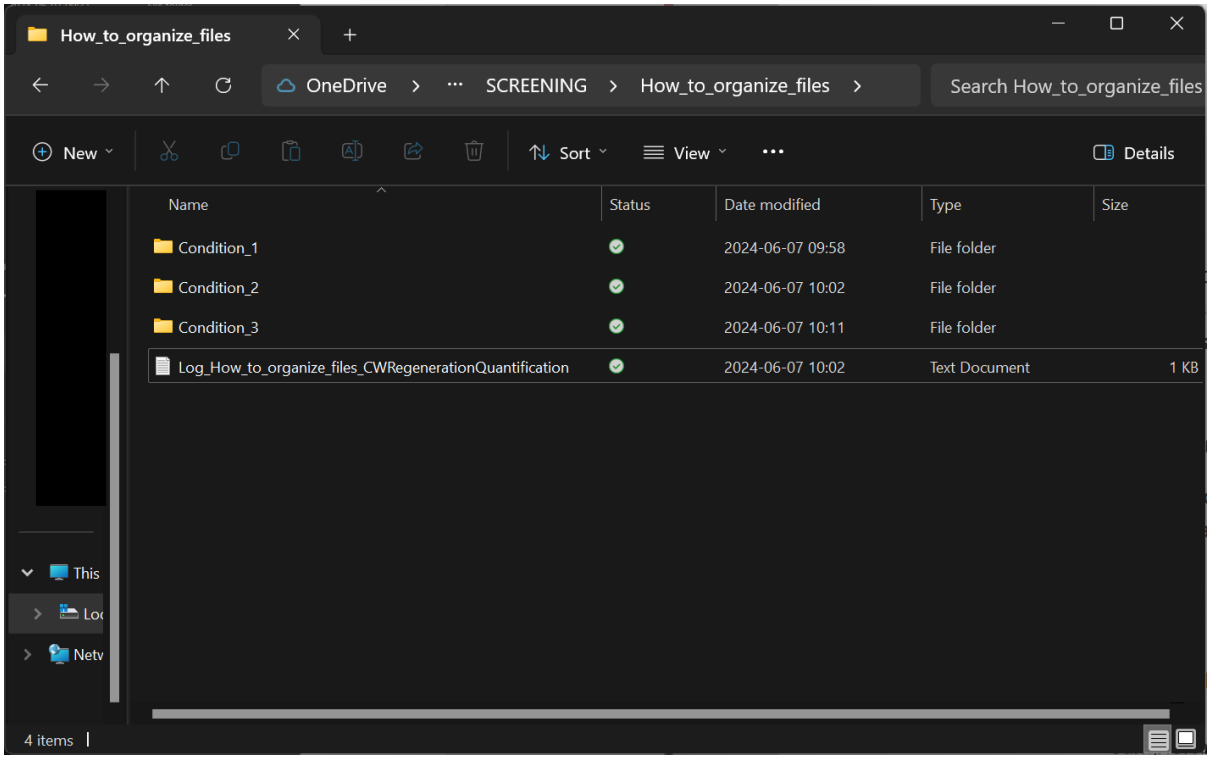


When the macro is running, a Log window allows to follow at which step of the process we are.

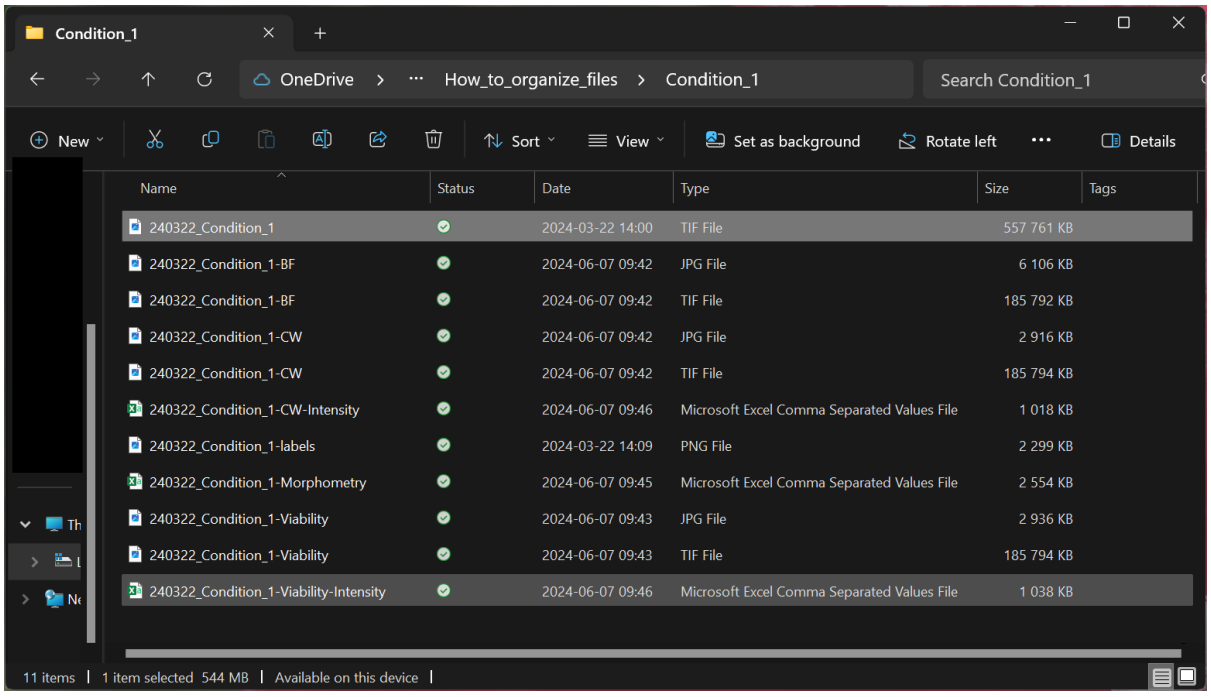
At the end of the macro, the Log window will also mention it.

Note: when running the macro, try to not use the computer, especially the keyboard because pop-up windows will come on top of all other windows opened and you might terminate the process by mistake.

The log is also saved into the working folder. It can be checked to verify all processed conditions.



The sample folders will contain many new files:

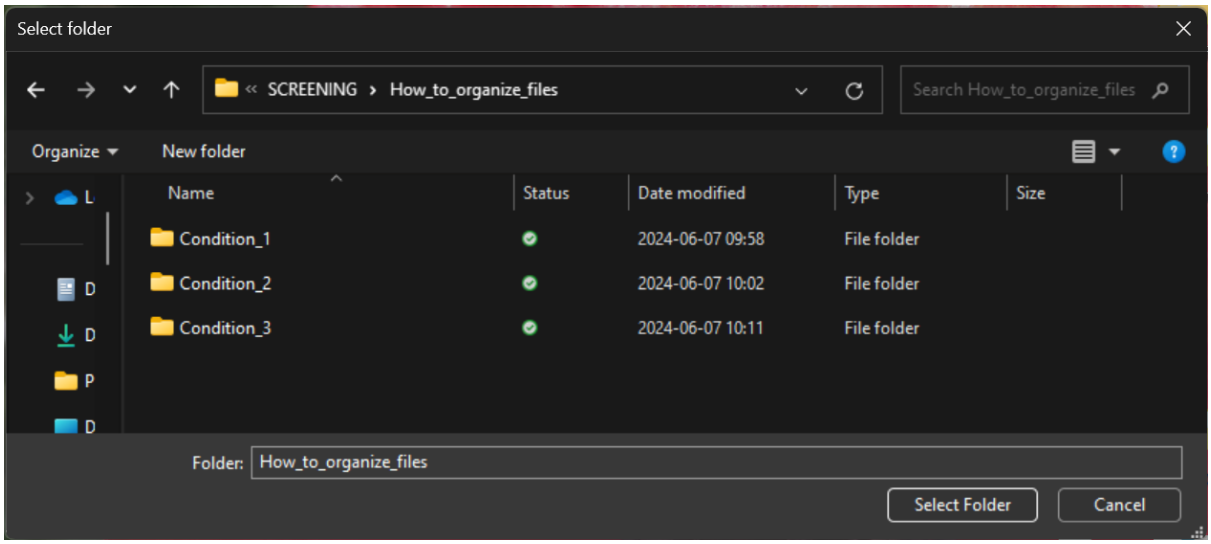


The macro saves each channel from the TIFF image in TIFF (for later potential use) and in JPG to be used in the Q-Warg-shinyapp. The data concerning morphometry (size, circularity, position), fluorescence intensity for vitality and cell wall of each cell are saved in 3 different CSV files. They will be processed by the R script.

3- Q-Warg analysis (R)

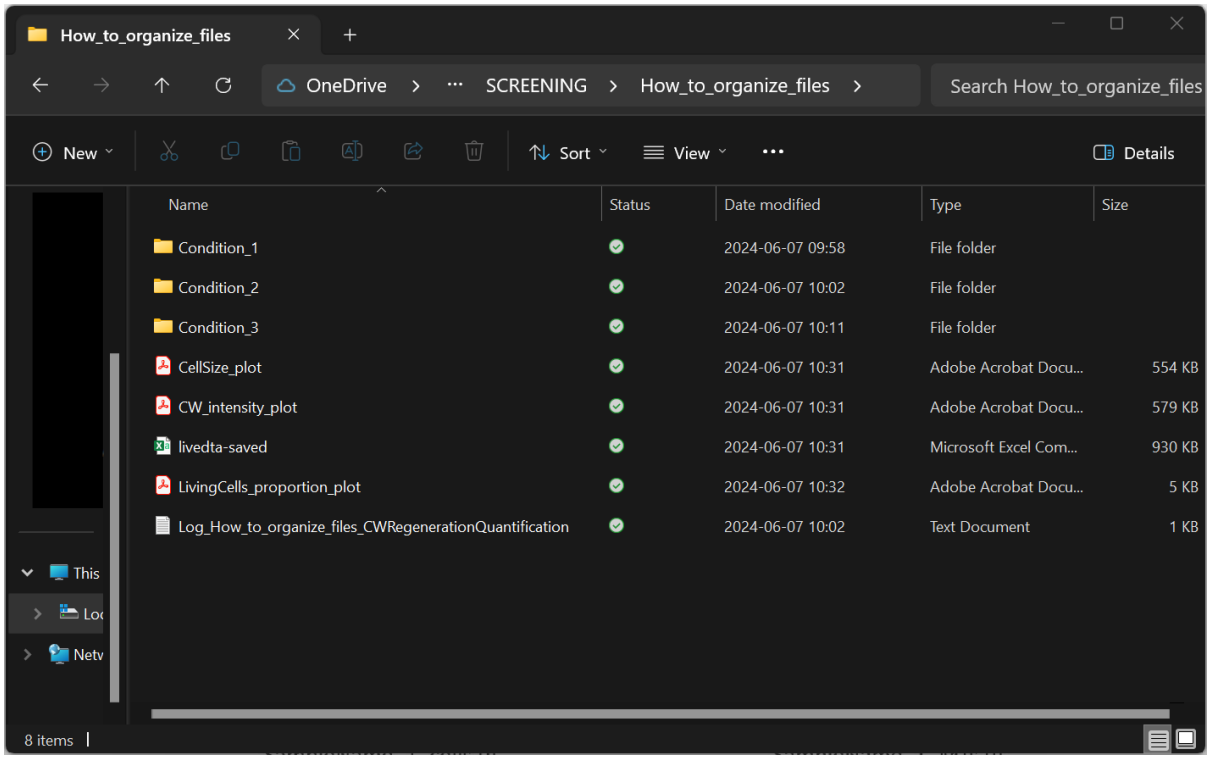
With this first script, the csv files obtained from the macro are combined and reorganized to obtain information about the living cells that have been segmented. Detailed information about the script can be found as comments directly in the script file Q-Warg -analysis.R.

The first step is to give the working folder, a pop-up window will open allowing to browse on the computer and find the folder. It should be the same as the one used for the previous macro.



To determine if the segmented objects are living cells, the script uses the viability staining and the circularity of the cells. A threshold of 500 is applied for the cell viability staining below this value, the cells are considered dead. This threshold is dependent on the viability staining and the imaging setup so it should be adjusted if needed. To ensure that the object is a cell, a high threshold is used for the circularity. Indeed, after protoplasting, the cells lose their original shape and are spherical. In absence of directional force on the cell, they stay spherical while regenerating their cell wall. Both parameters can be modified at line 54.

```
51  # filter out non Viability positive cells and non circular elements (that woul
52  ########### Depending on the staining, the Viability mean might need to be adj
53  wdta$cellalive <- "no"
54  wdta$cellalive[wdta$Circularity >= 0.8 & wdta$meanViability >= 500] <- "yes"
```



Q-Warg-analysis.R save the data table used to plot the data as a CSV file “livedta-saved.csv” in the working folder. This can be used to plot differently the data if needed. The data frame looks like:

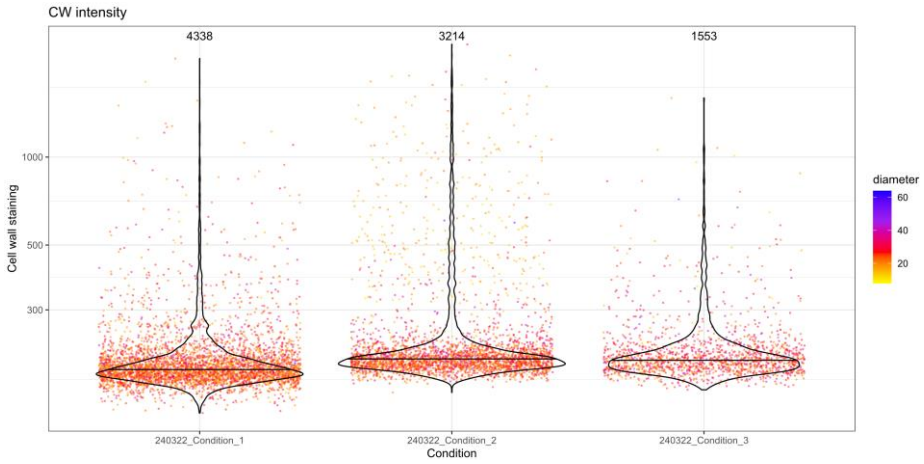
	cell	medium	diameter	Circularity	meanCW	meanViability	cooX	cooY	cellalive
1	240322_Condition_1-100	240322_Condition_1	36.69158	0.986	155.192	681.747	675.339	111.483	yes
2	240322_Condition_1-1000	240322_Condition_1	27.90336	1.001	178.394	1428.671	2164.137	1040.369	yes
3	240322_Condition_1-10001	240322_Condition_1	17.82535	0.986	263.421	1538.608	2771.289	10304.394	yes
4	240322_Condition_1-10014	240322_Condition_1	20.87222	0.986	194.312	860.608	2891.682	10311.533	yes
5	240322_Condition_1-10015	240322_Condition_1	19.21032	1.000	208.722	1594.619	2036.606	10311.209	yes
6	240322_Condition_1-10016	240322_Condition_1	23.38464	0.975	194.149	1008.568	1494.507	10314.740	yes
7	240322_Condition_1-10017	240322_Condition_1	27.16743	1.014	216.020	641.632	3025.054	10318.023	yes

One row for one cell:

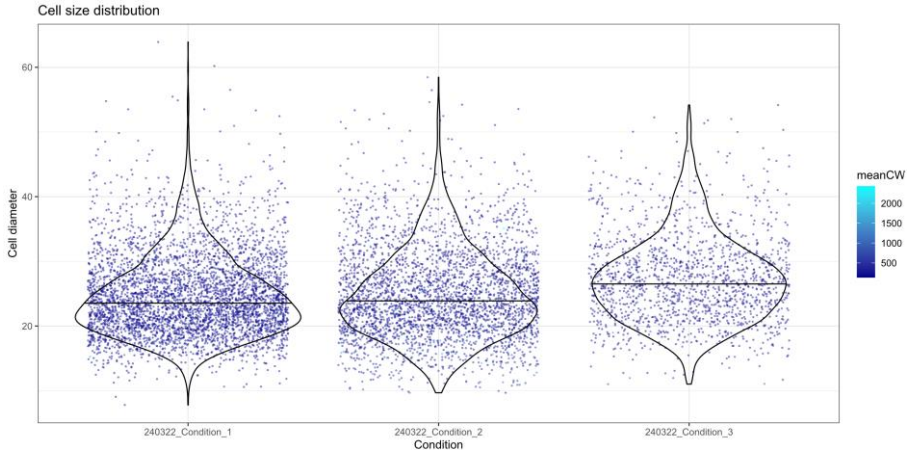
- Medium: condition of the sample
- Diameter: based on the perimeter measured by the macro in the morphometry csv file
- Circularity: measured by the macro in the morphometry csv file
- meanCW: mean intensity of the cell wall measured by the macro in the CW-Intensity csv file
- meanViability: mean intensity of the cell wall measured by the macro in the Viability-Intensity csv file
- cooX: coordinate of the cell in the image on the X axis (used for the Q-Warg-shinyapp)
- cooY: coordinate of the cell in the image on the Y axis (used for the Q-Warg-shinyapp)
- cellalive: yes for all (filtered from a previous data table).

3 plots are also saved in pdf (width and height can be adjusted line 91, 111 and 131 respectively):

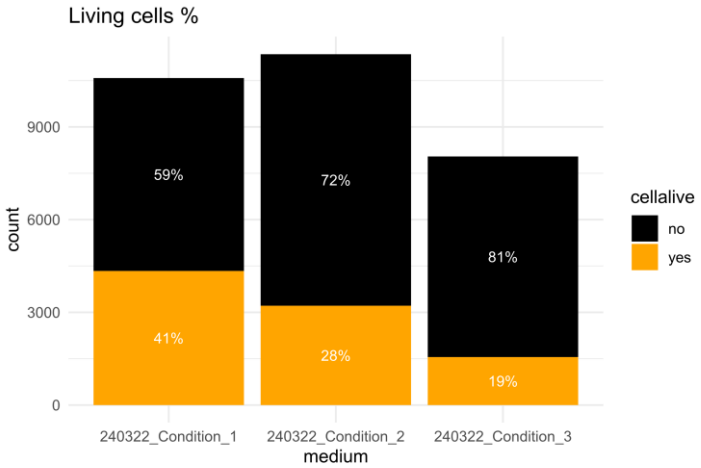
- Cell wall staining intensity with the size of the cells as color gradient. The above number is the total number of living cells.



- Size of the cells with the cell wall staining as a color gradient



- The living cells % contained into all segmented objects.

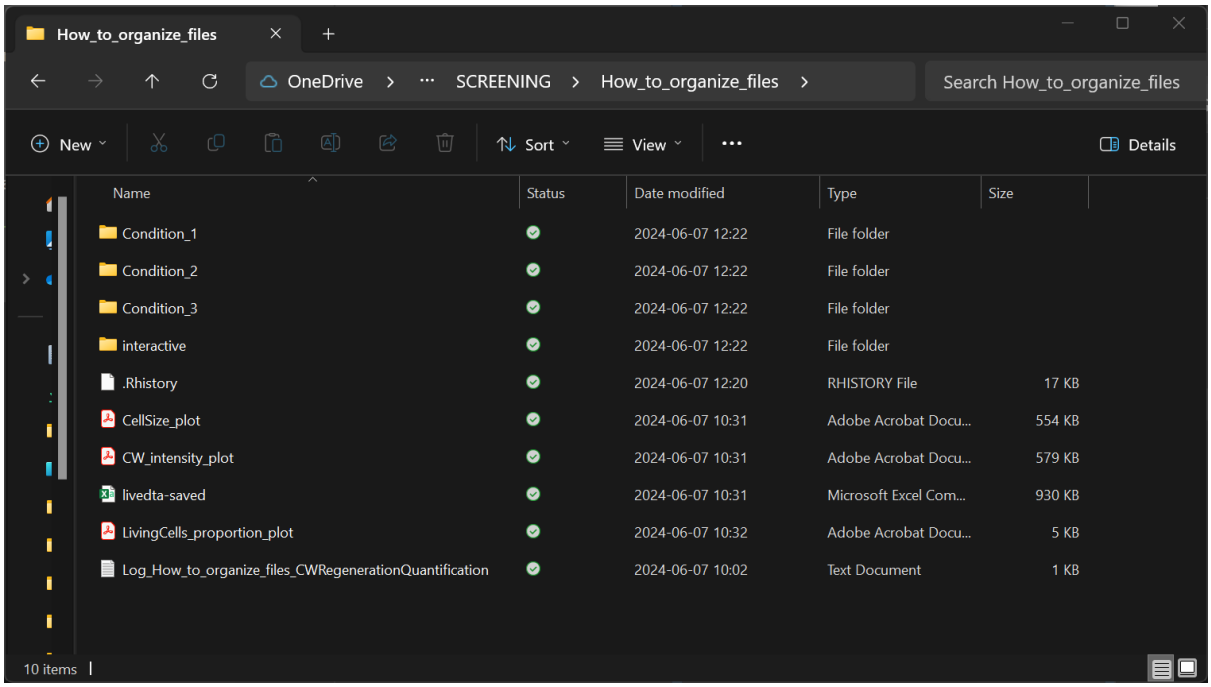


In the 3rd graph, the black percentage cannot be interpreted as a number of dead cells but non-living objects. While dying, a cell might have burst into several parts that have been segmented.

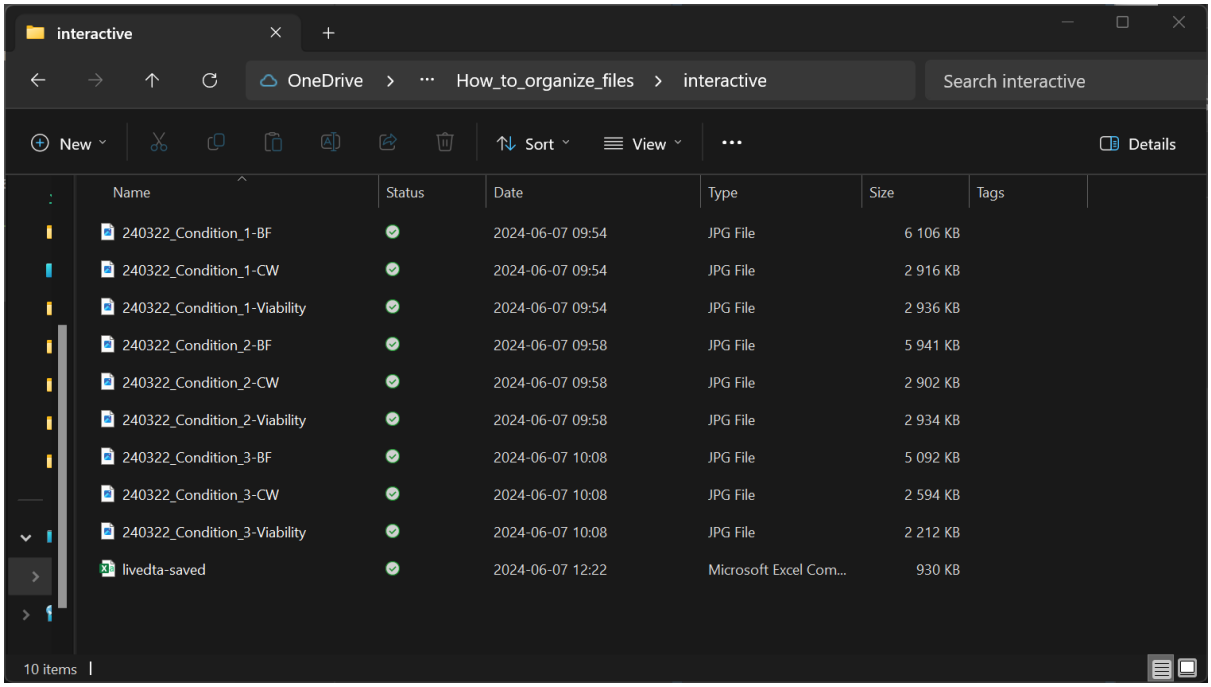
4- Q-Warg shiny app

The aim of this app is to check the good function of the analysis pipeline: each point of the plot corresponds to a living cell. It allows you to load the livedta file obtained with the previous script and click on each point of the graph to see the corresponding cell in the image in the 3 channels of the original picture.

To access both the images and the data table, the data must be reorganized. All the JPG images and the livedta-saved should be in one folder here called “interactive”.



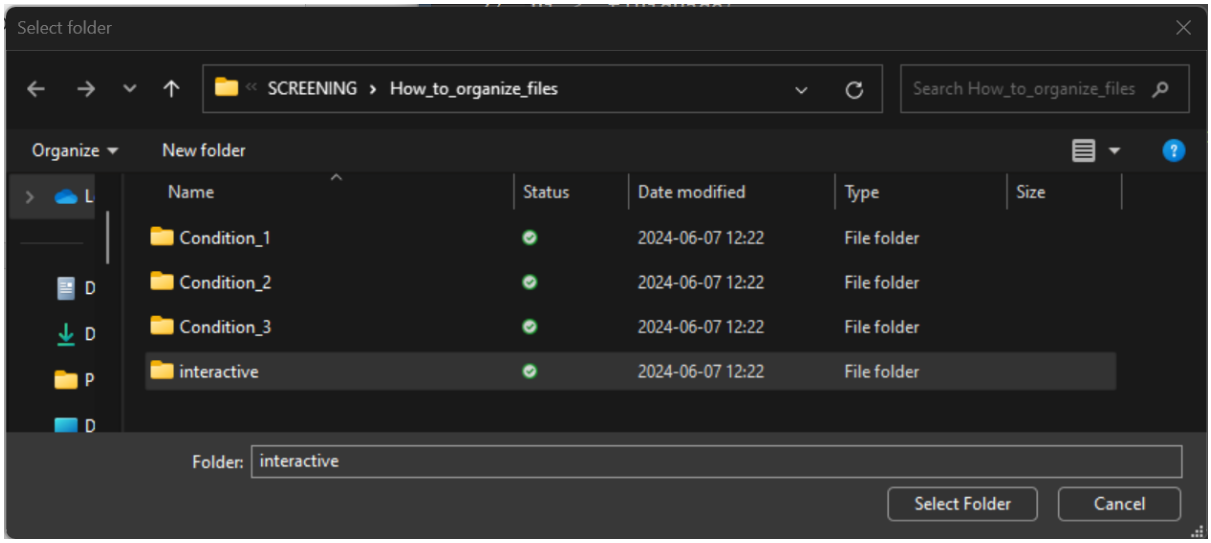
Here is the organization inside of the interactive folder:



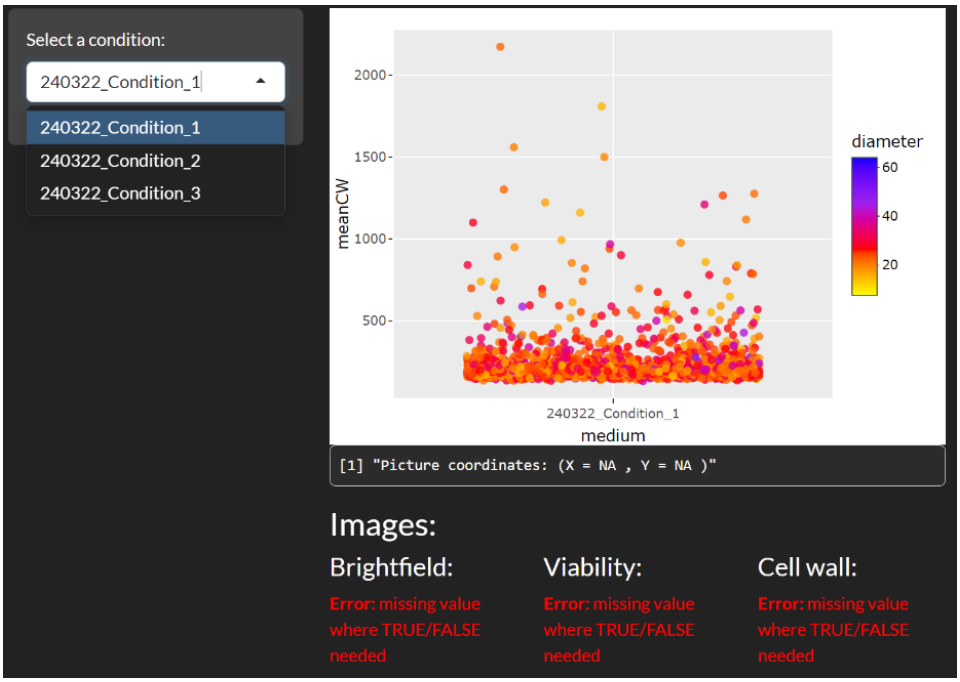
It is possible to do this reorganization by hand or use the R script Q-Warg-interactive-folder. The script is creating a new directory in the main working folder, moves inside all JPG files and copy the livedta-saved csv file.

The interactive folder is ready to be used in the Q-Warg-shinyapp.

When starting the script, a pop-up window opens to choose the directory:



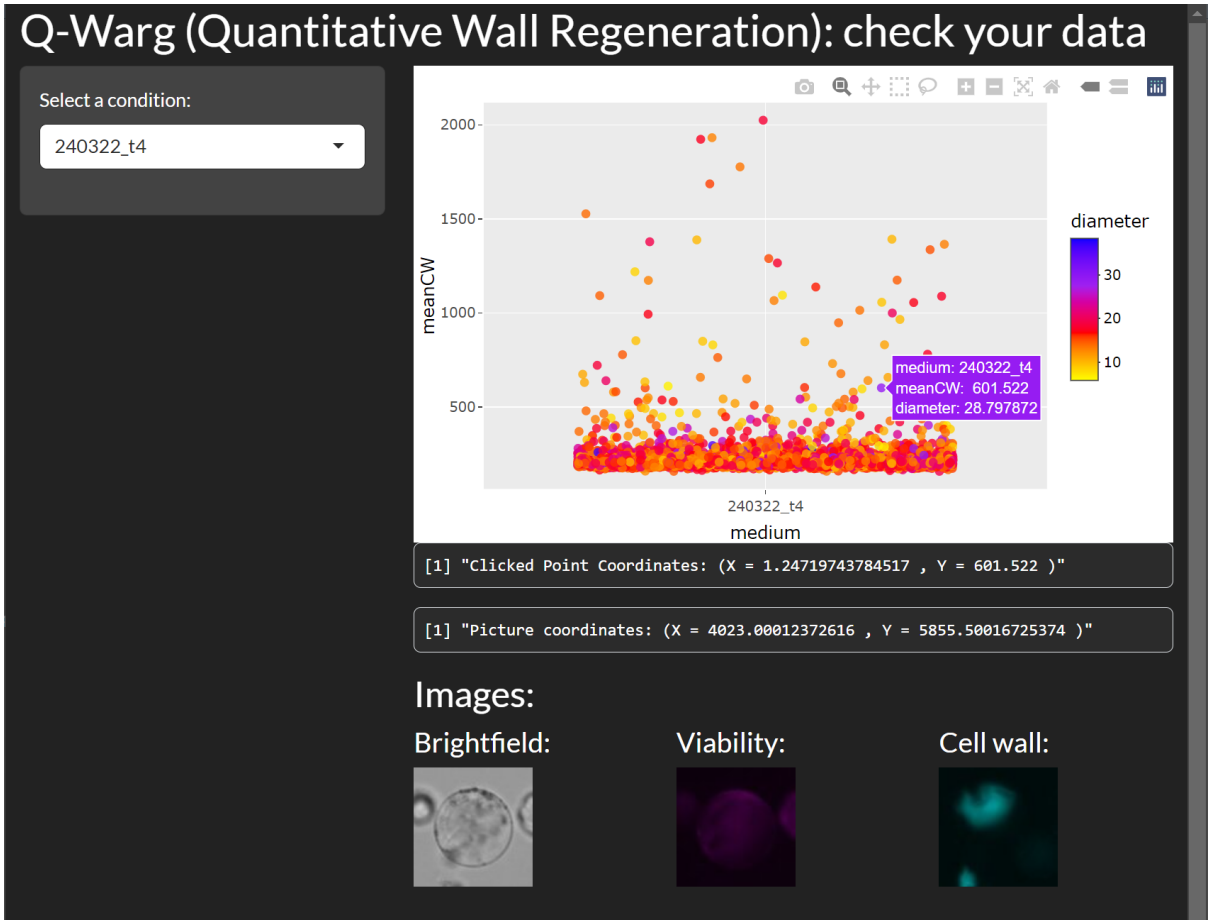
The shiny app looks like:



On the left panel, a rolling menu allows to choose which condition to check.

The graph shows the mean of the cell wall intensity for the chosen condition.

Below the graph, picture coordinates is NA because no point in the graph has been chosen yet. The red error message is there until a point is selected to show the corresponding cell.



“Clicked Point Coordinates” correspond to the coordinates of the point in the plot. It is based on the Y value that the script will find back the right coordinates in the jpg image corresponding to the chosen condition. “Picture coordinates” correspond to the cooX and cooY columns of the livedta-saved table. The cell corresponding to the point is shown at the center of the image.

When overlaying the mouse on top of the graph, information about each point is displayed.