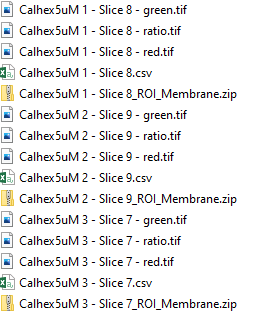
**Processing Zebrafish**

We use four separate scripts, as we experienced to be the best way of monitoring closely each step and keep proper record the data. First an initial processing is made on the Z-stack to explore the images, to establish proper threshold values and to identify the planes where the Notochord / Muscle / Epithelium are most clear. As a general criterion, the first plane where enough cells were visible was used to avoid going into deeper tissue where signal might start to be scattered. Second, the selected planes are extracted and the red, green and ratiometric images are generated for that particular plane. Third, a script is used to automatically detect membrane ROIs, and save the images as TIFF together with the ROI specifying the fish number, the condition and the plane used for the analysis. On this way, we can trace back the origin of each datapoint. This script already measures the Green/Red/ratio channels and save a csv file for each one. Finally, depending on the image quality, expression levels and cell shape, it might be necessary to correct or complement the detected ROIs (eliminate ROIs detected on cells corresponding to a different organ, add ROIs on non-detected cells corresponding to the target organ, etc.). After this is manually performed, the fourth script is executed, the final ROI list as well as the green/red/ratio values are updated.

Output file list example:



The CSV files can be opened and the data extracted into analysis software such as Prism, or compiled using standard python routines.

**Script 1: Exploring the stacks to identify planes containing the target tissue**

name=getTitle()

bname=File.nameWithoutExtension

green="C1-"

red="C2-"

gray="C3-"

run("Split Channels");

selectWindow(gray+name);

rename("gray");

selectWindow(green+name);

rename("green")

selectWindow(red+name);

rename("red")

low\_threshold\_green=0

high\_threshold\_green=255

low\_threshold\_red=5

high\_threshold\_red=255

// < < < < < < < < < < < < < GreenT-EC processing > > > > > > > > > > > > > >

// Noise reduction

selectWindow("green");

run("Median...", "radius=1.5 stack");

// Thresholding noise and saturated pixels

selectWindow("green");

run("Duplicate...", "duplicate");

setThreshold(low\_threshold\_green, high\_threshold\_green);

run("Convert to Mask", "method=Default background=Dark");

run("Divide...", "value=255 stack");

rename("threshold\_green")

imageCalculator("Divide create 32-bit stack", "green","threshold\_green");

rename("processed\_green")

// < < < < < < < < < < < < < mCyRFP1 processing > > > > > > > > > > > > > >

// Noise reduction

selectWindow("red");

run("Median...", "radius=1.5 stack");

// Thresholding

selectWindow("red");

run("Duplicate...", "duplicate");

setThreshold(low\_threshold\_red, high\_threshold\_red);

run("Convert to Mask", "method=Default background=Dark");

run("Divide...", "value=255 stack");

rename("threshold\_red")

imageCalculator("Divide create 32-bit stack", "red","threshold\_red");

rename("processed\_red")

// Getting ratiometric picture

imageCalculator("Divide create 32-bit stack", "processed\_green","processed\_red");

rename("ratio"+bname)

run("fire");

**Script** **2: Opening and processing single image from stack**

// Get images for analysis

File\_name="file name.czi"

Opening\_path="file directory"

first\_slice = 9

last\_slice = 9

first\_channel = 1 // 1: green / 2: red / 3: gray

last\_channel = 2

run("Bio-Formats Importer", "open=["+Opening\_path+File\_name+"] autoscale color\_mode=Default rois\_import=[ROI manager] specify\_range view=Hyperstack stack\_order=XYCZT c\_begin="+first\_channel+" c\_end="+last\_channel+" c\_step=1 z\_begin="+first\_slice+" z\_end="+last\_slice+" z\_step=1");

// Process to get Ratio

name=getTitle()

bname=File.nameWithoutExtension

green="C1-"

red="C2-"

gray="C3-"

result="Result of "

run("Split Channels");

// selectWindow(gray+name);

// rename("gray");

selectWindow(green+name);

rename("green");

selectWindow(red+name);

rename("red");

// < < < < < < < < < < < < < GreenT processing > > > > > > > > > > > > > >

// Noise reduction - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

selectWindow("green");

run("Duplicate...", "duplicate");

run("Gaussian Blur...", "sigma=1.5 stack");

rename("noise\_reduced\_green")

// Thresholding noise and saturated pixels

selectWindow("noise\_reduced\_green");

run("Duplicate...", "duplicate");

setThreshold(low\_threshold\_green, high\_threshold\_green);

run("Convert to Mask", "method=Default background=Dark");

run("Divide...", "value=255 stack");

rename("threshold\_green")

imageCalculator("Divide create 32-bit stack", "noise\_reduced\_green","threshold\_green");

rename("processed\_green")

// < < < < < < < < < < < < < mCyRFP1 processing > > > > > > > > > > > > > >

selectWindow("red");

run("Duplicate...", "duplicate");

// Noise reduction

run("Gaussian Blur...", "sigma=1.5 stack");

rename("noise\_reduced\_red")

// Thresholding noise - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

selectWindow("noise\_reduced\_red");

run("Duplicate...", "duplicate");

setThreshold(low\_threshold\_red, high\_threshold\_red);

run("Convert to Mask", "method=Default background=Dark");

run("Divide...", "value=255 stack");

rename("threshold\_red")

imageCalculator("Divide create 32-bit stack", "noise\_reduced\_red","threshold\_red");

rename("processed\_red")

// Getting ratiometric picture

imageCalculator("Divide create 32-bit stack", "processed\_green","processed\_red");

rename("ratio")

run("Brightness/Contrast...");

setMinAndMax(0.1, 2.8);

run("Fire");

// getting stack for measuring

run("Concatenate...", " title=Concatenated keep image1=processed\_green image2=processed\_red image3=ratio image4=[-- None --]");

**Script** **3: Identify membrane ROI’s**

saving\_path="folder path where data will be saved"

saving\_path\_auxiliary="folder path where auxiliary data will be saved"

// Complete !!!

File\_Original="Name containing Fish number and condition"

slice = Number of plane in Z-stack to be analyzed

// Name to be used for saving ROIs

File\_name=File\_Original+" - Slice "+slice;

// Generating slices for Threshold

selectWindow("red");

run("Duplicate...", " ");

run("Median...", "radius=1");

run("Subtract Background...", "rolling=25");

run("8-bit");

run("Auto Local Threshold", "method=Mean radius=30 parameter\_1=0 parameter\_2=0 white");

run("Convert to Mask", "method=Default background=Dark");

rename("Threshold\_for\_ROI")

// Detecting internal border of the membrane (values of minimun/maximun sizes depend on acquisition parameters)

selectWindow("Threshold\_for\_ROI");

run("Erode");

run("Dilate");

run("Median...", "radius=1");

run("Invert");

run("Analyze Particles...", "size=300-10000 pixel circularity=0.3-1.00 show=Nothing display clear add");

run("ROI Manager...");

if (roiManager("count")!=0) {

roiManager("Save", saving\_path\_auxiliary+File\_name+"\_ROI\_Cytosol.zip");

}

else{

continue;

}

// Enlarge to be sure it reaches the end of the membrane

counts=roiManager("count");

for(i=0; i<counts; i++) {

roiManager("Select", i);

run("Enlarge...", "enlarge=1.3");

roiManager("Update");

}

if (roiManager("count")!=0) {

roiManager("Save", saving\_path\_auxiliary+File\_name+"\_ROI\_Membrane\_Outside.zip");

}

// Reduce to avoid selecting ER signal from the own and other neighbor cells

counts=roiManager("count");

for(i=0; i<counts; i++) {

roiManager("Select", i);

run("Enlarge...", "enlarge=-0.9");

roiManager("Update");

}

if (roiManager("count")!=0) {

roiManager("Save", saving\_path\_auxiliary+File\_name+"\_ROI\_Membrane\_Inside.zip");

}

// Close ROI to avoid problems with old ROIs

selectWindow("ROI Manager");

run("Close");

// Final membrane ROI generation by combination of 2 sets of ROIs

run("ROI Manager...");

roiManager("Open", saving\_path\_auxiliary+File\_name+"\_ROI\_Membrane\_Inside.zip");

roiManager("Open", saving\_path\_auxiliary+File\_name+"\_ROI\_Membrane\_Outside.zip");

roiManager("count");

countsTotal=roiManager("count");

single=countsTotal/2;

for(i=0; i<single; i++) {

roiManager("Select", newArray(i,i+single));

roiManager("XOR");

roiManager("Add");

}

// Closing all ROIs except membrane

final\_count=roiManager("count");

for(i=0; i<final\_count-single; i++) {

roiManager("Select", newArray(1));

roiManager("Delete");

}

counts=roiManager("count");

for(i=0; i<counts; i++) {

roiManager("Select", i);

roiManager("Rename", i+1);

}

roiManager("Deselect");

roiManager("Show All");

roiManager("Save", saving\_path+File\_name+"\_ROI\_Membrane.zip");

// Measure Ratio values in membrane ROIs

selectWindow("Concatenated");

roiManager("Show All");

roiManager("Multi Measure");

saveAs("Results", saving\_path+File\_name+".csv");

// Saving TIFF files

selectWindow("red");

run("Duplicate...", " ");

saveAs("Tiff", saving\_path+File\_name+" - red.tif");

run("Close");

selectWindow("green");

run("Duplicate...", " ");

saveAs("Tiff", saving\_path+File\_name+" - green.tif");

run("Close");

selectWindow("ratio");

run("Duplicate...", " ");

saveAs("Tiff", saving\_path+File\_name+" - ratio.tif");

run("Close");

**Script** **4: Correct ROI selections if necessary and add manually remaining non-detected cells. This was particularly necessary for muscle cells due to the complexity of their shape and the higher inner signal.**

File\_Original="Name containing Fish number and condition" // Same as in script 3!

slice = Number of plane in Z-stack to be analyzed // Same as in script 3!

// Name to be used for saving ROIs

File\_name=File\_Original+" - Slice "+slice;

saving\_path="folder path where data will be saved" // Same as in script 3!

selectWindow("Concatenated");

//Converts 'n' to a string, left padding with zeros

//so the length of the string is 'width'

function leftPad(n, width) {

s =""+n;

while (lengthOf(s)<width)

s = "0"+s;

return s;

}

// It sorts the ROI list according to X-axis from right to left

run("Set Measurements...", " centroid redirect=None decimal=3");

for (i=0 ; i<roiManager("count"); i++) {

roiManager("select", i);

roiManager("Measure");

x = getResult("X",0);

roiManager("Rename", x-100000);

run("Clear Results");

}

roiManager("Show All");

roiManager("sort");

run("Set Measurements...", "mean redirect=None decimal=3");

// Rename

counts=roiManager("count");

for(i=0; i<counts; i++) {

roiManager("Select", i);

roiManager("Rename", i+1);

}

roiManager("Deselect");

roiManager("Show All");

roiManager("Save", saving\_path+File\_name+"\_ROI\_Membrane.zip");

// Measure Ratio values in membrane ROIs

selectWindow("Concatenated");

roiManager("Show All");

roiManager("Multi Measure");

saveAs("Results", saving\_path+File\_name+".csv");

**Time lapsed experiments**

**Script** **5: Due to small movements of zebrafish animals during the acquisition, it is necessary to identify the planes at each time-point where the analyzed cells are located. After identifying the planes, the following script can be used to collect them in a single stack.**

// Select window from which the planes will be extracted before executing

windowsname=getTitle()

// Set the number of planes to duplicate

num\_planes = 14

z1=8

z2=8

z3=8

z4=9

z5=9

z6=9

z7=9

z8=9

z9=9

z10=9

z11=10

z12=10

z13=10

z14=10

// Initialize an array to store the frame values

frames = newArray(z1,z2,z3,z4,z5,z6,z7,z8,z9,z10,z11,z12,z13,z14)

// Loop through the number of planes and duplicate them

for (i = 0; i < num\_planes; i++) {

// Duplicate the current plane and set the number of frames

selectWindow(windowsname);

run("Duplicate...", "duplicate slices="+frames[i]+"-"+frames[i]+" frames="+(i+1));

// Rename the duplicated window

rename("t"+(i+1));

}

// Creates a string for the conctatenate function

concatenate\_string=""

for (i = 1; i < num\_planes+1; i++) {

// Duplicate the current plane and set the number of frames

concatenate\_string= concatenate\_string+" image"+i+"=t"+i;

}

run("Concatenate...", " title=Selected\_Planes"+concatenate\_string+" ");

**Script** **6: After drawing the ROI on the notochord area execute the following macro to save the data.**

save\_name= "Control"

saving\_path="P:\\"

// DRAW roi

// Save ROI

roiManager("Show All");

roiManager("Save", saving\_path+save\_name+"\_ROI.zip");

// Measure Ratio values in membrane ROIs

selectWindow("red");

roiManager("Show All");

roiManager("Multi Measure");

saveAs("Results", saving\_path+save\_name+"\_red.csv");

selectWindow("green");

roiManager("Show All");

roiManager("Multi Measure");

saveAs("Results", saving\_path+save\_name+"\_green.csv");

selectWindow("ratio");

roiManager("Show All");

roiManager("Multi Measure");

saveAs("Results", saving\_path+save\_name+"\_ratio.csv");

// Saving TIFF files

selectWindow("red");

run("Duplicate...", "duplicate");

saveAs("Tiff", saving\_path+save\_name+" - red.tif");

run("Close");

selectWindow("green");

run("Duplicate...", "duplicate");

saveAs("Tiff", saving\_path+save\_name+" - green.tif");

run("Close");

selectWindow("ratio");

run("Duplicate...", "duplicate");

saveAs("Tiff", saving\_path+save\_name+" - ratio.tif");

run("Close");