*Image analysis and macro scripts for rsCaMPARI photoswitching*

Macro scripts in Fiji software1 were used to analyze fluorescence intensities from imaging experiments. For dissociated neuron experiments, channels corresponding to rsCaMPARI green, mRuby3 red, and ChrimsonR-HaloTag(JF635) far-red were deinterleaved. ROIs were either manually drawn around neuronal somatas or automatically selected using the Trainable Weka Segmentation plugin2 using a classifier trained on segmenting neuronal somatas. Images were rolling ball background subtracted3 and mean pixel values within ROIs were saved in Excel. For zebrafish experiments, a script was generated to calculate cosine similarity between image stacks. Code of the Fiji macro scripts used are reproduced below.

//This is a macro for analysis of rsCaMPARI two channel images from field stimulation

//assays acquired on a Nikon widefield microscope.

//It separates channels, selects ROIs, subtracts background, measures intensities,

//and saves the images.

setBatchMode(false);

topdir = getDirectory("select top directory of dataset");

rawdir = topdir + "raw/";

savedir = topdir + "process/";

file\_list = getFileList(rawdir);

var filename\_array = newArray(0);

var donkey = newArray(1);

for (i=0; i<file\_list.length; i++) {

showProgress(i, file\_list.length);

run("Bio-Formats Importer", "open=" + rawdir + file\_list[i] + " color\_mode=Default rois\_import=[ROI manager] view=Hyperstack stack\_order=XYCZT");

filename=getTitle();

donkey[0] = filename;

filename\_array = Array.concat(filename\_array,donkey[0]);

run("Make Substack...", "channels=1 slices=2-201,216-415,430-629");

saveAs("tiff", savedir + filename + "\_green\_stack");

rename("green\_stack");

selectWindow(filename);

run("Make Substack...", "channels=2 slices=1");

saveAs("tiff", savedir + filename + "\_red");

rename("pre\_red");

selectWindow("pre\_red");

run("Trainable Weka Segmentation");

wait(3000);

selectWindow("Trainable Weka Segmentation v3.2.32");

call("trainableSegmentation.Weka\_Segmentation.loadClassifier", topdir + "classifier\_v2.model");

call("trainableSegmentation.Weka\_Segmentation.getResult");

while (!isOpen("Classified image")) {

wait(100);

}

selectWindow("Classified image");

saveAs("tiff", savedir + filename + "\_classified");

run("8-bit");

setThreshold(0, 120);

run("Convert to Mask");

run("Watershed");

run("Analyze Particles...", "size=50 exclude add");

roiManager("Save", savedir + filename + "\_RoiSet.zip");

selectWindow("green\_stack");

run("Subtract Background...", "rolling=50");

roiManager("Multi Measure");

selectWindow("pre\_red");

run("Subtract Background...", "rolling=50");

roiManager("multi-measure measure\_all one append");

saveAs("Results", savedir + filename + "\_intensities.csv");

run("Close All");

selectWindow("ROI Manager");

run("Close");

selectWindow("Results");

run("Close");

}

selectWindow("Log");

run("Close");

//This is a macro for analysis of rsCaMPARI two channel images from spontaneous

//recovery assay acquired on a Nikon widefield microscope.

//It separates channels, selects ROIs, subtracts background, measures intensities,

//and saves the images.

setBatchMode(false);

topdir = getDirectory("select top directory of dataset");

rawdir = topdir + "raw\_thermal/";

savedir = topdir + "process\_thermal/";

file\_list = getFileList(rawdir);

var filename\_array = newArray(0);

var donkey = newArray(1);

for (i=0; i<file\_list.length; i++) {

showProgress(i, file\_list.length);

run("Bio-Formats Importer", "open=" + rawdir + file\_list[i] + " color\_mode=Default rois\_import=[ROI manager] view=Hyperstack stack\_order=XYCZT");

filename=getTitle();

donkey[0] = filename;

filename\_array = Array.concat(filename\_array,donkey[0]);

run("Make Substack...", "channels=1 slices=204,206,208,210,212,214,216,218,220,222,224,226,228,230");

run("Set Label...", "label=T0");

run("Next Slice [>]");

run("Set Label...", "label=T1");

run("Next Slice [>]");

run("Set Label...", "label=T2");

run("Next Slice [>]");

run("Set Label...", "label=T3");

run("Next Slice [>]");

run("Set Label...", "label=T4");

run("Next Slice [>]");

run("Set Label...", "label=T5");

run("Next Slice [>]");

run("Set Label...", "label=T6");

run("Next Slice [>]");

run("Set Label...", "label=T7");

run("Next Slice [>]");

run("Set Label...", "label=T8");

run("Next Slice [>]");

run("Set Label...", "label=T9");

run("Next Slice [>]");

run("Set Label...", "label=T10");

run("Next Slice [>]");

run("Set Label...", "label=T11");

run("Next Slice [>]");

run("Set Label...", "label=T12");

run("Next Slice [>]");

run("Set Label...", "label=reset");

saveAs("tiff", savedir + filename + "\_thermal\_stack");

rename("thermal\_stack");

selectWindow(filename);

run("Make Substack...", "channels=2 slices=1");

saveAs("tiff", savedir + filename + "\_pre\_red");

rename("pre\_red");

selectWindow(filename);

run("Make Substack...", "channels=1 slices=2");

saveAs("tiff", savedir + filename + "\_pre\_green");

rename("pre\_green");

selectWindow("pre\_red");

run("Trainable Weka Segmentation");

wait(3000);

selectWindow("Trainable Weka Segmentation v3.2.28");

call("trainableSegmentation.Weka\_Segmentation.loadClassifier", topdir + "classifier\_v2.model");

call("trainableSegmentation.Weka\_Segmentation.getResult");

while (!isOpen("Classified image")) {

wait(100);

}

selectWindow("Classified image");

saveAs("tiff", savedir + filename + "\_classified");

run("8-bit");

setThreshold(0, 120);

run("Convert to Mask");

run("Watershed");

run("Analyze Particles...", "size=50 exclude add");

roiManager("Save", savedir + filename + "\_RoiSet.zip");

selectWindow("thermal\_stack");

run("Subtract Background...", "rolling=50");

roiManager("Multi Measure");

selectWindow("pre\_red");

run("Subtract Background...", "rolling=50");

roiManager("multi-measure measure\_all one append");

saveAs("Results", savedir + filename + "\_intensities.csv");

run("Close All");

selectWindow("ROI Manager");

run("Close");

selectWindow("Results");

run("Close");

}

Array.print(filename\_array);

for (i=0; i<file\_list.length; i++) {

setResult("Label", i, filename\_array[i]);

updateResults();

}

selectWindow("Results");

saveAs("Text", savedir + "mean\_redgreenratio\_within\_mask");

run("Close");

selectWindow("Log");

run("Close");

//This is a macro for analysis of rsCaMPARI three channel images from channelrhodopsin

//assays acquired on a Nikon widefield microscope.

//It separates channels, creates composite images, subtracts background, measures

//intensities, and saves the images.

//Images are manually segmented based on the first red image and the ROIs are saved.

//ROIs that have ChrimsonR value > 10 are considered ChrimsonR positive.

//ROIs that have ChrimsonR value < 4 are considered ChrimsonR negative.

setBatchMode(false);

topdir = getDirectory("select top directory of dataset");

rawdir = topdir + "raw/";

savedir = topdir + "process/";

file\_list = getFileList(rawdir);

for (i=0; i<file\_list.length; i += 2) {

showProgress(i, file\_list.length);

run("Bio-Formats Importer", "open=" + rawdir + file\_list[i] + " color\_mode=Default view=[Standard ImageJ] stack\_order=Default");

filename=getTitle();

selectWindow(filename);

run("Make Substack...", " slices=2,299,453,454,458,755,909,910,914,1211,1365,1366");

run("Set Label...", "label=C1\_pre\_green");

run("Next Slice [>]");

run("Set Label...", "label=C1\_post\_green");

run("Next Slice [>]");

run("Set Label...", "label=C1\_red");

run("Next Slice [>]");

run("Set Label...", "label=C1\_ChrimsonR");

run("Next Slice [>]");

run("Set Label...", "label=C2\_pre\_green");

run("Next Slice [>]");

run("Set Label...", "label=C2\_post\_green");

run("Next Slice [>]");

run("Set Label...", "label=C2\_red");

run("Next Slice [>]");

run("Set Label...", "label=C2\_ChrimsonR");

run("Next Slice [>]");

run("Set Label...", "label=C3\_pre\_green");

run("Next Slice [>]");

run("Set Label...", "label=C3\_post\_green");

run("Next Slice [>]");

run("Set Label...", "label=C3\_red");

run("Next Slice [>]");

run("Set Label...", "label=C3\_ChrimsonR");

saveAs("tiff", savedir + filename + "\_cleanup");

run("Stack to Images");

selectWindow("C1\_red");

saveAs("tiff", savedir + filename + "\_C1\_red");

rename("C1\_red");

selectWindow("C1\_ChrimsonR");

saveAs("tiff", savedir + filename + "\_C1\_ChrimsonR");

rename("C1\_ChrimsonR");

selectWindow("C1\_pre\_green");

saveAs("tiff", savedir + filename + "\_C1\_pre\_green");

rename("C1\_pre\_green");

selectWindow("C1\_post\_green");

saveAs("tiff", savedir + filename + "\_C1\_post\_green");

rename("C1\_post\_green");

selectWindow("C2\_red");

saveAs("tiff", savedir + filename + "\_C2\_red");

rename("C2\_red");

selectWindow("C2\_ChrimsonR");

saveAs("tiff", savedir + filename + "\_C2\_ChrimsonR");

run("Enhance Contrast", "saturated=0.01");

run("RGB Color");

saveAs("tiff", savedir + filename + "\_RGB\_C2\_ChrimsonR");

rename("C2\_ChrimsonR");

selectWindow("C2\_pre\_green");

saveAs("tiff", savedir + filename + "\_C2\_pre\_green");

rename("C2\_pre\_green");

selectWindow("C2\_post\_green");

saveAs("tiff", savedir + filename + "\_C2\_post\_green");

rename("C2\_post\_green");

selectWindow("C3\_red");

saveAs("tiff", savedir + filename + "\_C3\_red");

rename("C3\_red");

selectWindow("C3\_ChrimsonR");

saveAs("tiff", savedir + filename + "\_C3\_ChrimsonR");

rename("C3\_ChrimsonR");

selectWindow("C3\_pre\_green");

saveAs("tiff", savedir + filename + "\_C3\_pre\_green");

rename("C3\_pre\_green");

selectWindow("C3\_post\_green");

saveAs("tiff", savedir + filename + "\_C3\_post\_green");

rename("C3\_post\_green");

run("Merge Channels...", "c2=C1\_pre\_green c6=C1\_red create keep");

selectWindow("Composite");

setMinAndMax(350, 2000);

run("Next Slice [>]");

setMinAndMax(350, 9000);

saveAs("tiff", savedir + filename + "\_composite\_C1\_pre");

run("RGB Color");

saveAs("tiff", savedir + filename + "\_composite\_RGB\_C1\_pre");

close();

close();

run("Merge Channels...", "c2=C1\_post\_green c6=C1\_red create keep");

selectWindow("Composite");

setMinAndMax(350, 2000);

run("Next Slice [>]");

setMinAndMax(350, 9000);

saveAs("tiff", savedir + filename + "\_composite\_C1\_post");

run("RGB Color");

saveAs("tiff", savedir + filename + "\_composite\_RGB\_C1\_post");

close();

close();

run("Merge Channels...", "c2=C2\_pre\_green c6=C2\_red create keep");

selectWindow("Composite");

setMinAndMax(350, 2000);

run("Next Slice [>]");

setMinAndMax(350, 9000);

saveAs("tiff", savedir + filename + "\_composite\_C2\_pre");

run("RGB Color");

saveAs("tiff", savedir + filename + "\_composite\_RGB\_C2\_pre");

close();

close();

run("Merge Channels...", "c2=C2\_post\_green c6=C2\_red create keep");

selectWindow("Composite");

setMinAndMax(350, 2000);

run("Next Slice [>]");

setMinAndMax(350, 9000);

saveAs("tiff", savedir + filename + "\_composite\_C2\_post");

run("RGB Color");

saveAs("tiff", savedir + filename + "\_composite\_RGB\_C2\_post");

close();

close();

run("Merge Channels...", "c2=C3\_pre\_green c6=C3\_red create keep");

selectWindow("Composite");

setMinAndMax(350, 2000);

run("Next Slice [>]");

setMinAndMax(350, 9000);

saveAs("tiff", savedir + filename + "\_composite\_C3\_pre");

run("RGB Color");

saveAs("tiff", savedir + filename + "\_composite\_RGB\_C3\_pre");

close();

close();

run("Merge Channels...", "c2=C3\_post\_green c6=C3\_red create keep");

selectWindow("Composite");

setMinAndMax(350, 2000);

run("Next Slice [>]");

setMinAndMax(350, 9000);

saveAs("tiff", savedir + filename + "\_composite\_C3\_post");

run("RGB Color");

saveAs("tiff", savedir + filename + "\_composite\_RGB\_C3\_post");

close();

close();

roiManager("Open", rawdir + file\_list[i+1]);

selectWindow("C1\_ChrimsonR");

roiManager("Multi Measure");

selectWindow("C1\_red");

run("Subtract Background...", "rolling=50");

roiManager("multi-measure measure\_all one append");

selectWindow("C1\_pre\_green");

run("Subtract Background...", "rolling=50");

roiManager("multi-measure measure\_all one append");

selectWindow("C1\_post\_green");

run("Subtract Background...", "rolling=50");

roiManager("multi-measure measure\_all one append");

saveAs("Results", savedir + filename + "\_intensities\_C1.csv");

selectWindow("C1\_ChrimsonR");

roiManager("Multi Measure");

selectWindow("C2\_red");

run("Subtract Background...", "rolling=50");

roiManager("multi-measure measure\_all one append");

selectWindow("C2\_pre\_green");

run("Subtract Background...", "rolling=50");

roiManager("multi-measure measure\_all one append");

selectWindow("C2\_post\_green");

run("Subtract Background...", "rolling=50");

roiManager("multi-measure measure\_all one append");

saveAs("Results", savedir + filename + "\_intensities\_C2.csv");

selectWindow("C1\_ChrimsonR");

roiManager("Multi Measure");

selectWindow("C3\_red");

run("Subtract Background...", "rolling=50");

roiManager("multi-measure measure\_all one append");

selectWindow("C3\_pre\_green");

run("Subtract Background...", "rolling=50");

roiManager("multi-measure measure\_all one append");

selectWindow("C3\_post\_green");

run("Subtract Background...", "rolling=50");

roiManager("multi-measure measure\_all one append");

saveAs("Results", savedir + filename + "\_intensities\_C3.csv");

selectWindow("ROI Manager");

run("Close");

selectWindow("Results");

run("Close");

run("Close All");

}

//This is a macro for computation of pixel-based cosine similarity between image

//stacks.

setBatchMode(true);

topdir = getDirectory("select top directory of dataset");

rawdir = topdir + "raw/";

savedir = topdir + "process/";

file\_list = getFileList(rawdir);

setOption("ExpandableArrays", true);

root\_squared\_sums = newArray;

norm\_cross\_coeff = newArray;

for (i=1; i<=file\_list.length; i += 1) {

showProgress(i, file\_list.length);

run("Bio-Formats Importer", "open=" + rawdir + file\_list[i-1] + " color\_mode=Default rois\_import=[ROI manager] view=Hyperstack stack\_order=XYCZT");

rename("C" + i);

//Calculate summed squares

imageCalculator("Multiply create 32-bit stack", "C" + i,"C" + i);

selectWindow("Result of C" + i);

run("Measure Stack...");

myArray = newArray(nResults);

for ( j=0; j<nResults; j++ ) {

myArray[j] = getResult("RawIntDen", j);

}

NoNaN = Array.deleteValue(myArray, NaN);

total = 0;

for (k=0; k<lengthOf(NoNaN); k++) {

total=total+NoNaN[k];

}

root\_squared\_sums[i-1] = sqrt(total);

Array.show(root\_squared\_sums);

close("Result of C" + i);

close("Results");

}

//compute cross-correlation and then normalize by the root sum of squares calculated previously

i=1;

for (j=1; j<=file\_list.length; j += 1) {

for(k=1; k<=file\_list.length; k += 1) {

imageCalculator("Multiply create 32-bit stack", "C" + j,"C" + k);

selectWindow("Result of C" + j);

run("Measure Stack...");

myArray = newArray(nResults);

for ( l=0; l<nResults; l++ ) {

myArray[l] = getResult("RawIntDen", l);

}

NoNaN = Array.deleteValue(myArray, NaN);

Array.show(myArray);

selectWindow("myArray");

saveAs("Results", savedir + "C" + j + "\_C" + k + ".csv");

total = 0;

for (l=0; l<lengthOf(NoNaN); l++) {

total=total+NoNaN[l];

}

norm\_cross\_coeff[i-1] = total / root\_squared\_sums[j-1] / root\_squared\_sums[k-1];

i=i + 1;

Array.show(norm\_cross\_coeff);

close("Result of C" + j);

close("Results");

close("C" + j + "\_C" + k + ".csv");

}

norm\_cross\_coeff[i-1] = "column" + j + 1;

i=i+1;

}

selectWindow("norm\_cross\_coeff");

saveAs("Results", savedir + "norm\_cross\_coeff.csv");

selectWindow("root\_squared\_sums");

saveAs("Results", savedir + "root\_squared\_sums.csv");

run("Close All");

**References**

1 Schindelin, J. *et al.* Fiji: an open-source platform for biological-image analysis. *Nat Methods* **9**, 676-682, doi:10.1038/nmeth.2019 (2012).

2 Arganda-Carreras, I. *et al.* Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics* **33**, 2424-2426, doi:10.1093/bioinformatics/btx180 (2017).

3 Sternberg, S. R. Biomedical Image-Processing. *Computer* **16**, 22-34 (1983).