```
macro "LN Reporter Regions"
{
//Draw 4 sets of ROI's and place them in the same folder as the image.
//set GFP and RFP thresholds by the negative image

var thresholdGFP = 16;
var thresholdRFP = 23;
var thresholdPE = 4;
var thresholdF480 = 32;

            //get image attributes
            name = getInfo("image.filename");
            original = getImageID();
            originalTitle = getTitle();

            //Set the directory where files will be pulled from and saved to
            OpenDir=getDirectory("Choose the folder where data will be pulled
from");
            DataDir=getDirectory("Choose the folder where data will be saved");
            //convert to 32 bit
            run("32-bit");

            // split channels and threshold the GFP and RFP images.
            //If GFP=0, the ratio image will be 0, and will become NaN during
background/NaN step
            //If RFP=0, the ratio will be x/0, which should return NaN, due to
Misc options
            run("Duplicate...", "title=GFPimage duplicate channels=1");
            GFPImage = getImageID();
            GFPImageTitle = getTitle();
            changeValues(0, thresholdGFP, 0)
            selectImage(original);
            run("Duplicate...", "title=RFPimage duplicate channels=2");
            RFPImage = getImageID();
            RFPImageTitle = getTitle();
            changeValues(0, thresholdRFP, 0)

            // divide GFPImage by RFPImage
            run("Misc...", "divide=NaN");
            imageCalculator("Divide create 32-bit", GFPImage, RFPImage);
            rename("RatioImage");
            RatioImage = getImageID();
            RatioImageTitle = getTitle();

            // generation of membrane mask
            selectImage(original);
            run("Duplicate...", "title=PE duplicate channels=3");
            //run("Median...", "radius=1.0");
            setThreshold(thresholdPE, 255);
            run("Convert to Mask", " black");
            PEImage = getImageID();
            rename("MemMaskImage")
            //to make the mask values either 1 or 0.
            run("Divide...", "value=255.000000000");
            MemMaskImage = getImageID();
```

```
      MemMaskTitle = getTitle();

      // calculate RFP to GFP ratio on membrane.
      imageCalculator("Multiply create 32-bit", RatioImage, MemMaskImage);
      setThreshold(0.0001, 255.0000);
      run("NaN Background");
      rename(originalTitle + "MemRatio");
      MemRatioImage = getImageID();
      MemRatioTitle = getTitle();
      run("Fire");
      saveAs("Tiff", DataDir+MemRatioTitle+".tif");
      setMinAndMax(0.0000, 4.000);
// cleanup and select results table
      //selectImage(original); close();
      selectImage(RFPImage); close();
      selectImage(GFPImage);  close();
      //selectImage(RatioImage); close();

//Ratio measurements
run("Set Measurements...", "area mean display redirect=None decimal=3");
roiManager("Reset");
open(OpenDir + "RoiSetMed.zip");
ROICountMed=roiManager("Count");//=3
for (i=0; i<ROICountMed; i++) {
      roiManager("Select",i);
      roiManager("Rename", "Med");
      }
      //roiManager("Reset");
      open(OpenDir + "RoiSetMedL.zip");
      ROICountMedL=roiManager("Count");
for (i=ROICountMed; i<ROICountMedL; i++) {
      roiManager("Select",i);
      roiManager("Rename", "MedL");
      }
//roiManager("Measure");
//roiManager("Reset");
      open(OpenDir + "RoiSetT.zip");
      ROICountT=roiManager("Count")
for (i=ROICountMedL; i<ROICountT; i++) {
      roiManager("Select",i);
      roiManager("Rename", "T");
      }
      open(OpenDir + "RoiSetSCS.zip");
      ROICountSCS=roiManager("Count")
for (i=ROICountT; i<ROICountSCS; i++) {
      roiManager("Select",i);
      roiManager("Rename", "SCS");
      }
      roiManager("Deselect");
      roiManager("Measure");
      // save the results table as an excel file
      //Adding the settings to the results table:
      i = nResults;    // variable for counting, initialising with 0
      setResult("Image name", i, originalTitle);
```

```
            setResult("thresholdGFP", i, thresholdGFP); // add a "Label" column
to the results table and name the entry "point1"
            setResult("thresholdRFP", i, thresholdRFP);
            setResult("thresholdPE", i, thresholdPE);
            setResult("thresholdF480", i, thresholdF480);
            selectWindow("Results");
            saveAs("Results", DataDir + originalTitle + ".xls");
            }
```

```
macro "LN Reporter transduced primary T cells"
{
var thresholdGFP = 5;
var thresholdRFP = 30;
var thresholdPE = 12;

            //get image attributes
            name = getInfo("image.filename");
            original = getImageID();
            originalTitle = getTitle();
            shortTitle= substring(originalTitle, 0,11)
            CellROI="Cells " + substring(originalTitle, 0,11) + ".zip";
            MedROI="MedROI.zip";
            TROIs="TROI.zip";
            TROI="TROI.zip";
            //Set the directory where files will be pulled from and saved to
            OpenDir=getDirectory("Choose the folder where data will be pulled
from");
            DataDir=getDirectory("Choose the folder where data will be saved");
            //convert to 32 bit
            run("32-bit");

            //split channels and threshold the GFP and RFP images.
            //If GFP=0, the ratio image will be 0, and will become NaN during
background-->NaN
            //If RFP=0, the ratio will be x/0, which should return NaN, as set
in Misc options
            run("Duplicate...", "title=GFPimage duplicate channels=1");
            GFPImage = getImageID();
            GFPImageTitle = getTitle();
            changeValues(0, thresholdGFP, 0)
            selectImage(original);
            run("Duplicate...", "title=RFPimage duplicate channels=2");
            RFPImage = getImageID();
            RFPImageTitle = getTitle();
            changeValues(0, thresholdRFP, 0)

            // divide image1 by image2
            run("Misc...", "divide=NaN");
            imageCalculator("Divide create 32-bit", GFPImage, RFPImage);
            rename("RatioImage");
            RatioImage = getImageID();
            RatioImageTitle = getTitle();


            // generation of membrane mask
            //Mask the PE
            selectImage(original);
            run("Duplicate...", "title=PE duplicate channels=3");
            run("Median...", "radius=.5");
            setThreshold(thresholdPE, 255);
            run("Convert to Mask", " black");
            PEImage = getImageID();
            rename("MemMaskImage")
            //to make the mask values either 1 or 0.
```

```
            run("Divide...", "value=255.000000000");
            MemMaskImage = getImageID();
            MemMaskTitle = getTitle();

            // calculate RFP to GFP ratio on membrane.
            imageCalculator("Multiply create 32-bit", RatioImage, MemMaskImage);
            //run("Threshold...");
            setThreshold(0.0001, 255.0000);
            run("NaN Background");
            rename(originalTitle + "MemRatio");
            MemRatioImage = getImageID();
            MemRatioTitle = getTitle();
            run("Fire");
            setMinAndMax(0.0000, 1.000);
            saveAs("Tiff", DataDir+MemRatioTitle+".tif");
            roiManager("Reset");
            if (File.exists(OpenDir + CellROI) == 1) x=x;
            else {
      setTool("rectangle");
            waitForUser("Draw the T cell ROIs as small as possible");
            roiManager("Save", DataDir + CellROI);
  }
            if (File.exists(OpenDir + MedROI) == 1) x = x;
            else {
      roiManager("Reset");
      setTool("polygon");
            waitForUser("Draw the Medulla ROIs and add to manager");
            ROICount=roiManager("Count");//=3
      for (i=0; i<ROICount; i++) {
            roiManager("Select",i);
            roiManager("Rename", "Med"); }
            roiManager("Save", DataDir + MedROI);
            }
            //here, open all the ROIs, which are named properly now. Then open
the T zone ROI.
            if (File.exists(OpenDir + TROIs) == 1  || File.exists(OpenDir +
TROI)) x = x;
            else {
      roiManager("Reset");
      setTool("polygon");
            waitForUser("Draw the T zone ROIs and add to manager");
            ROICount=roiManager("Count");//=3
      for (i=0; i<ROICount; i++) {
            roiManager("Select",i);
            roiManager("Rename", "T"); }
            roiManager("Save", DataDir + TROI);
            }

//generate a single ROI from all the TROIs
            roiManager("Reset");
            open(DataDir + TROIs);
            ROICount=roiManager("Count");
            roiManager("Deselect");
            roiManager("Combine");
            roiManager("Add");
```

```
            roiManager("Select",ROICount);
            roiManager("Rename", "Whole T");
            a1 = newArray(ROICount);
            for (i=0; i<a1.length; i++)
                  a1[i] = i;
            roiManager("select", a1);
            roiManager("Delete");
//rename the cells that have a centroid inside the T zone
            open(DataDir + CellROI);
            ROICount=roiManager("Count");
            CellStart=1;
            CellEnd=ROICount-1;
            for (i=1; i<ROICount; i++) {
                  roiManager("select", i);
                  Roi.getBounds(x, y, width, height);
                  a= x+width/2;
                  b= y+height/2;
                  roiManager("select", 0);
                        if (Roi.contains(a,b)==1) {
                              roiManager("Select", i);
                              roiManager("Rename", "T zone cell");
                              }
                  }
//open the medulla ROIs and generate a single ROI.
            ROICount1=roiManager("Count"); //9
            open(DataDir + MedROI);
            ROICount2=roiManager("Count"); //11
            array = Array.getSequence(ROICount2);
            Selection=Array.slice(array,ROICount1,ROICount2);
            roiManager("select", Selection);
            roiManager("Combine");
            roiManager("Add");
            roiManager("Select",ROICount2);
            roiManager("Rename", "Whole Med");
//rename the cells that have a centroid inside the medulla
            for (i=1; i<ROICount; i++) {
                  roiManager("select", i);
                  Roi.getBounds(x, y, width, height);
                  a= x+width/2;
                  b= y+height/2;
                  roiManager("select", ROICount2);
                        if (Roi.contains(a,b)==1) {
                              roiManager("Select", i);
                              roiManager("Rename", "Med cell");
                              }
                  }
            ROICountTotal=roiManager("Count"); //11
            arrayTotal = Array.getSequence(ROICountTotal);
            SelectionFinal=Array.slice(arrayTotal,ROICount,ROICountTotal);
            selectWindow(MemRatioTitle+".tif");
            roiManager("select", SelectionFinal);
            roiManager("Delete");
            roiManager("select", 0); //deleting the T zone ROI
            roiManager("Delete");
            //Measure the cell ratios
```

```
            run("Set Measurements...", "area mean display redirect=None
decimal=3");
            roiManager("Deselect");
            roiManager("Measure");

            i = nResults;    // variable for counting, initialising with 0
            setResult("Image name", i, originalTitle);
            setResult("thresholdGFP", i, thresholdGFP); // add a "Label" column
to the results table and name the entry "point1"
            setResult("thresholdRFP", i, thresholdRFP);
            setResult("thresholdPE", i, thresholdPE);
            selectWindow("Results");
            saveAs("Results", DataDir + shortTitle + ".xls");
            }

}
```