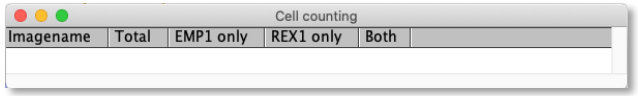
“Co_occurrence_EMP1vREX1.ijm” Macro Workflow

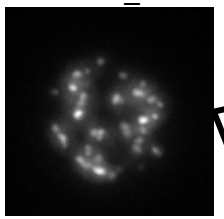
Start → Initialize data table



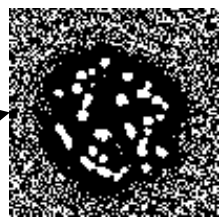
Open image;
Split channels;
Project z-stack
(Maximum Intensity Projection)

Mask generation

‘REX1_MIP’

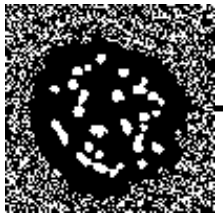


Auto local threshold

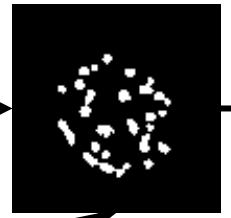


local threshold

Blur



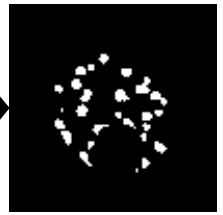
local AND blur



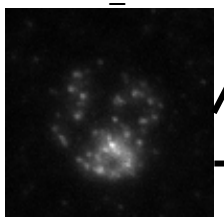
Result AND ‘cellmask’

Result AND ‘cytoplasm’
= ‘REX1mask’

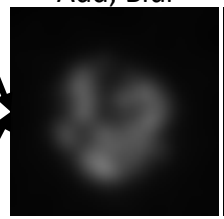
‘REX1mask’ + ‘EMP1mask’
= ‘Bothmask’



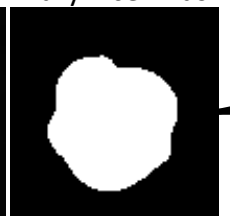
‘EMP1_MIP’



Add; Blur

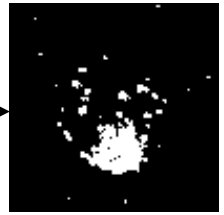


Binary =‘cellmask’

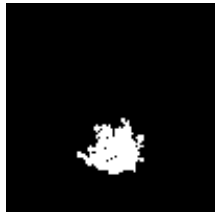


‘cellmask’ - Blur
= ‘cytoplasm’

Auto local threshold



local threshold



largest particle

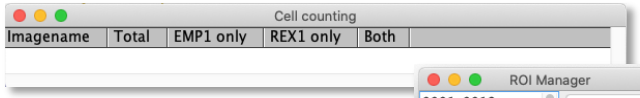
Blur

local threshold
AND ‘cytoplasm’

Particles ≥2 pixels
= ‘EMP1mask’

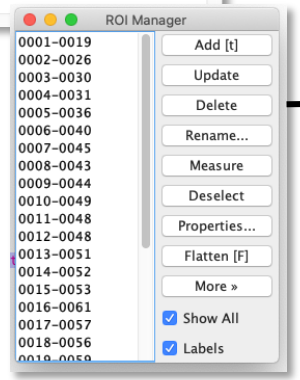
Co-occurrence counts

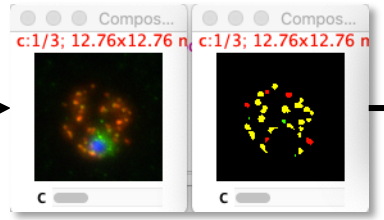
Generate black images for drawing ‘REX1only’, ‘EMP1only’, and ‘Both’ events;
Initiate ‘for’ loop

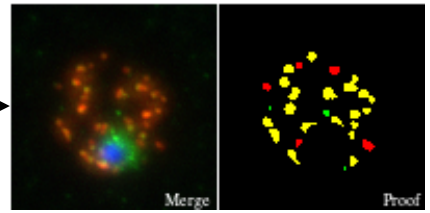


Log data

ROI loop







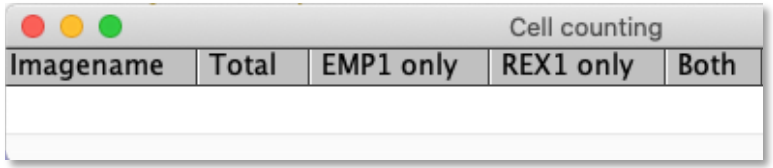
Generate proof: Merge MIPs;
Laterally combine with ‘REX1only’, ‘EMP1only’, and ‘Both’ masks;
Label; Save

Co_occurrence_EMP1vREX1.ijm

```
1
2   //get input and output directories
3   indir = getDirectory("select a source folder");
4   outdir = getDirectory("select a destination folder");
5
6   //get the image list
7   list = getFileList(indir);
8
9   //make result table
10  title1 = "Cell counting";
11  title2 = "["+title1+"]";
12  if (isOpen(title1)) {
13      selectWindow(title1);
14      run("Close");
15  }
16  run("Table...", "name="+title2+" width=600 height=300");
17  print(title2, "\\Headings:Imagename \t Total \t EMP1 only \t REX1 only \t Both");
```

Select input and output directories

Initialize table and assign headings



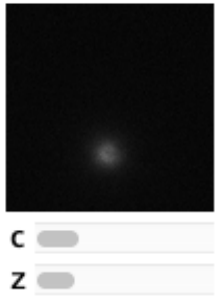
Co_occurrence_EMP1vREX1.ijm

```
18
19    //start batch loop
20    for(i=0; i<list.length; i++){
21
22        run("Close All");
23
24        //open image, grab the file name
25        open(indir + list[i]);
26        nameonly = File.nameWithoutExtension;
27
28        //split channels and rename each channels with static name
29        run("Split Channels");
30        image = "C1-"+ list[i];
31        selectWindow(image);
32        run("Z Project...", "projection=[Max Intensity]");
33        rename("DAPI_MIP");
34
35        image = "C2-"+ list[i];
36        selectWindow(image);
37        run("Z Project...", "projection=[Max Intensity]");
38        rename("EMP1_MIP");
39
40        image = "C3-"+ list[i];
41        selectWindow(image);
42        run("Z Project...", "projection=[Max Intensity]");
43        rename("REX1_MIP");
44
```

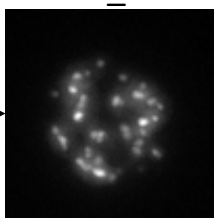
Begin batch loop

Close any open images

Open image (hyperstack)

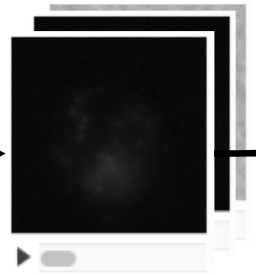


‘REX1_MIP’



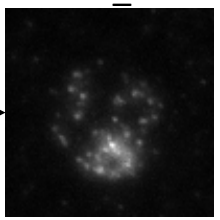
Opened image (hyperstack)

Split channels

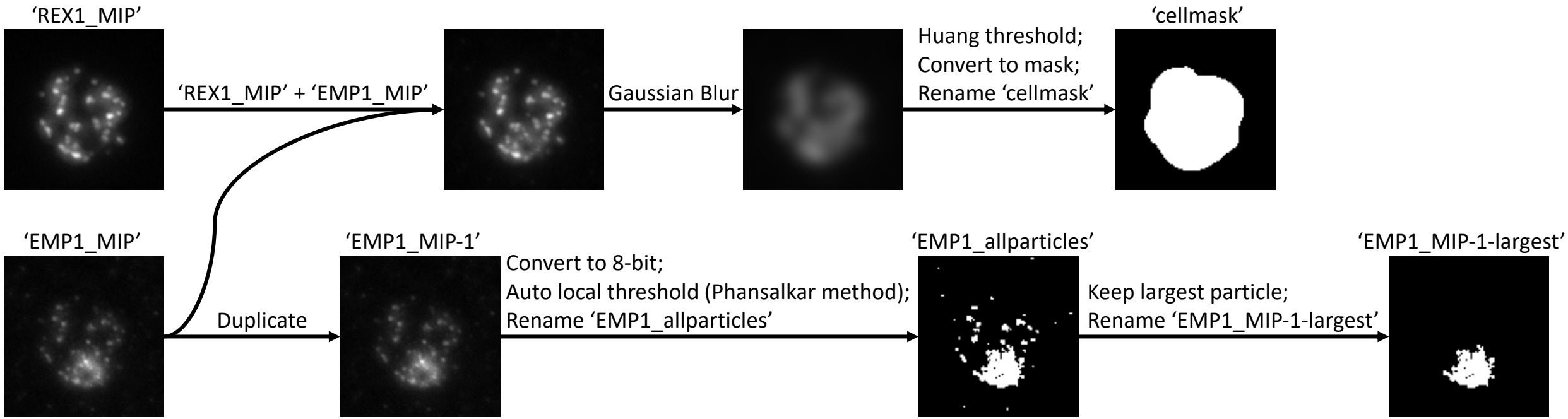


Project z-stack (Maximum Intensity Projection);
Rename

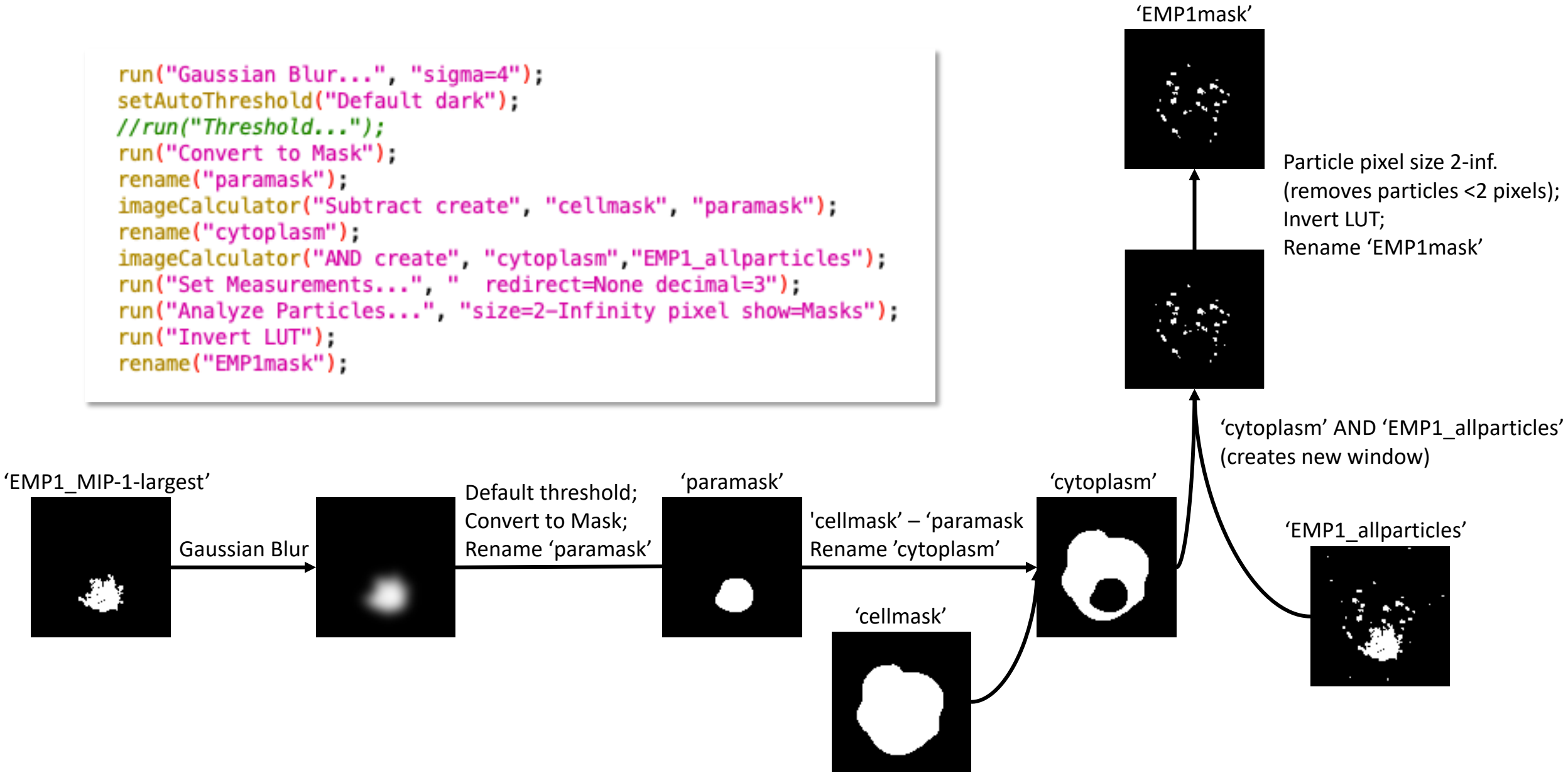
‘EMP1_MIP’



```
//create masks
    //make cell mask w/ gaussian blur
    imageCalculator("Add create", "EMP1_MIP","REX1_MIP");
    run("Gaussian Blur...", "sigma=4");
    setAutoThreshold("Huang dark");
    run("Convert to Mask");
    rename("cellmask");

    //Make parasite mask via EMP1 staining
    selectWindow("EMP1_MIP");
    run("Duplicate...", " ");
    setOption("ScaleConversions", true);
    run("8-bit");
    run("Auto Local Threshold", "method=Phansalkar radius=2 parameter_1=0 parameter_2=0 white");
    rename("EMP1_allparticles");
    run("Keep Largest Region");
    rename("EMP1_MIP-1-largest");
```

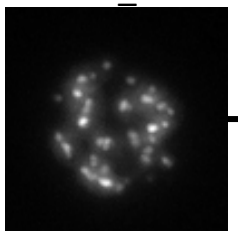


```
run("Gaussian Blur...", "sigma=4");
setAutoThreshold("Default dark");
//run("Threshold...");
run("Convert to Mask");
rename("paramask");
imageCalculator("Subtract create", "cellmask", "paramask");
rename("cytoplasm");
imageCalculator("AND create", "cytoplasm","EMP1_allparticles");
run("Set Measurements...", "  redirect=None decimal=3");
run("Analyze Particles...", "size=2-Infinity pixel show=Masks");
run("Invert LUT");
rename("EMP1mask");
```



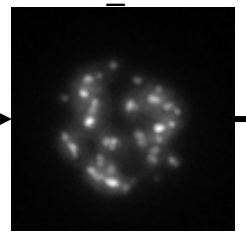
```
//REX1 clefts mask (local threshold)
selectWindow("REX1_MIP");
run("Duplicate...", " ");
setOption("ScaleConversions", true);
run("8-bit");
run("Auto Local Threshold", "method=Niblack radius=5 parameter_1=0 parameter_2=0 white");

rename("REX1mask_clefts");
run("Duplicate...", " ");
run("Gaussian Blur...", "sigma=2");
setAutoThreshold("Default dark");
rename("REX1mask_blur");
//run("Threshold...");
run("Convert to Mask");
imageCalculator("AND create", "REX1mask_clefts","REX1mask_blur");
rename("REX1_binary1");
imageCalculator("AND create", "cellmask","REX1_binary1");
rename("REX1_binary2");
imageCalculator("AND create", "REX1_binary2","cytoplasm");
run("Watershed");
run("Set Measurements...", "  redirect=None decimal=3");
run("Analyze Particles...", "size=5-Infinity pixel show=Masks");
run("Invert LUT");
rename("REX1mask");
```

‘REX1_MIP’



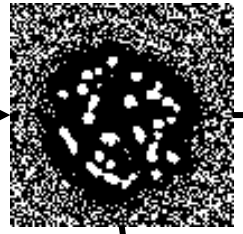
Duplicate

‘REX1_MIP-1’



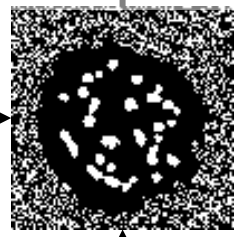
Convert to 8-bit;
Auto local threshold
(Niblack method);
Rename ‘REX1mask_clefts’

‘REX1mask_clefts’

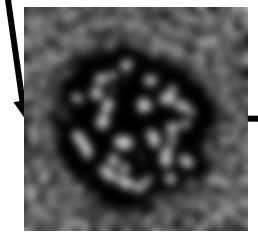


‘REX1mask_clefts’
AND ‘REX1mask_blur’;
Rename ‘REX1_binary1’

‘REX1_binary1’



Duplicate;
Gaussian Blur

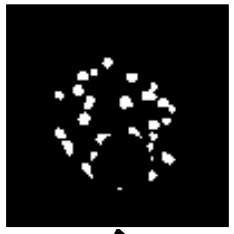


Default threshold;
Rename
‘REX1mask_blur’

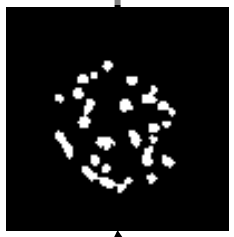
‘REX1mask_blur’

‘REX1_binary1’ AND ‘cytoplasm’;
Watershed;
Particle pixel size 5-inf.
(remove particles >5 pixels);
Invert LUT;
Rename ‘REX1mask’

‘REX1mask’



‘REX1_binary2’

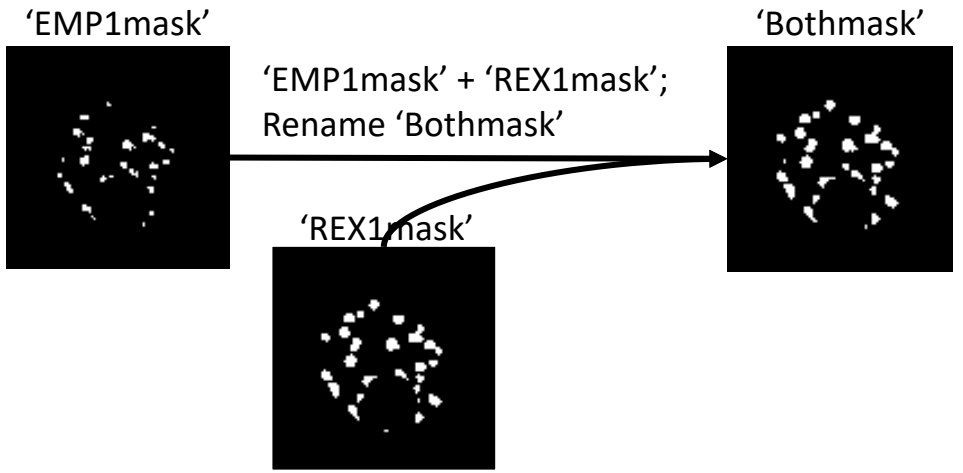


‘cytoplasm’

‘cellmask’ AND ‘REX1_binary1’;
Rename ‘REX1_binary2’

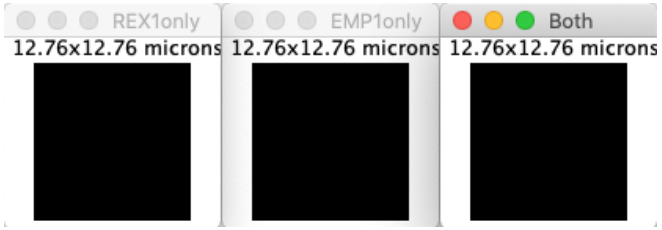
‘cellmask’

```
//create 'both' mask
imageCalculator("Add create", "EMP1mask","REX1mask");
rename("Bothmask");
```




```
//create 3 new black images - Redonly, Greenonly, Both
imageCalculator("Subtract create", "Bothmask","Bothmask");
rename("REX1only");
run("Duplicate...", "title=EMP1only");
run("Duplicate...", "title=Both");


//make a counter
all=roiManager("count");
EMP1only=0;
REX1only=0;
both=0;

//set drawing color to white
setForegroundColor(255, 255, 255);

//run FOR loop for all ROIs
for(a=0;a<all;a++){

    //for each ROI, check if it is positive to green or red mask
    selectWindow("REX1mask");
    roiManager("Select", a);
    List.setMeasurements;
    REX1den = List.getValue("RawIntDen");

    selectWindow("EMP1mask");
    roiManager("Select", a);
    List.setMeasurements;
    EMP1den = List.getValue("RawIntDen");
```

Create 3 new black images; Rename

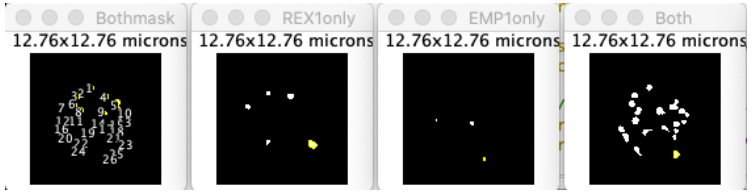


Make a counter for EMP1only, REX1only, and both;
Set drawing color to white

For each region of interest, get Raw Integrated Density

```
    //sort ROI and draw on each image
    if(REX1den>0 && EMP1den>0){
        selectWindow("Both");
        roiManager("Select", a);
        run("Fill", "slice");
        both++;
    }else if(REX1den>0 && EMP1den==0){
        selectWindow("REX1only");
        roiManager("Select", a);
        run("Fill", "slice");
        REX1only++;
    }else if(REX1den==0 && EMP1den>0){
        selectWindow("EMP1only");
        roiManager("Select", a);
        run("Fill", "slice");
        EMP1only++;
    }
}

//log counting to table
print(title2, list[i] + "\t" + all + "\t" + EMP1only + "\t" + REX1only + "\t" + both);

//make proof image
run("Merge Channels...", "c1=REX1_MIP c2=EMP1_MIP c3=DAPI_MIP create keep");
run("Stack to RGB");
rename("rawimage");
run("Select All");
setForegroundColor(255, 255, 255);
run("Draw", "slice");
```

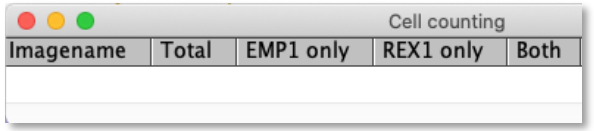
If Raw Integrated Density is >0 on both REX1mask AND EMP1 mask, then add count for 'Both';
Draw ROI on 'Both'



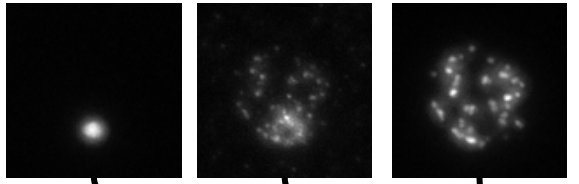
If Raw Integrated Density is >0 on REX1mask only, then add count for 'REX1only';
Draw ROI on 'REX1only''

If Raw Integrated Density is >0 on EMP1mask only, then add count for 'EMP1only';
Draw ROI on 'EMP1only''

Log counts to table

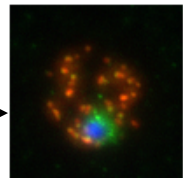


'DAPI_MIP' 'EMP1_MIP' 'REX1_MIP'

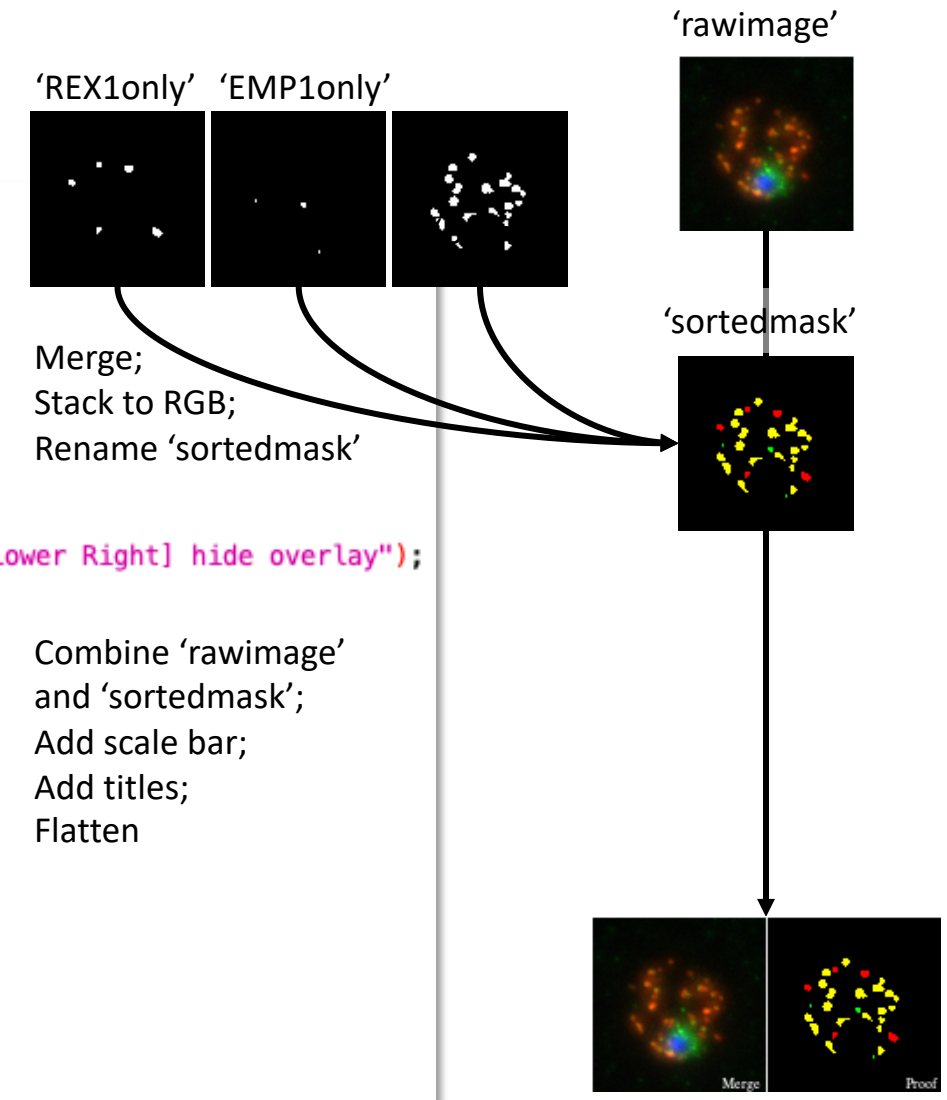


Merge;
Stack to RGB;
Rename 'rawimage'

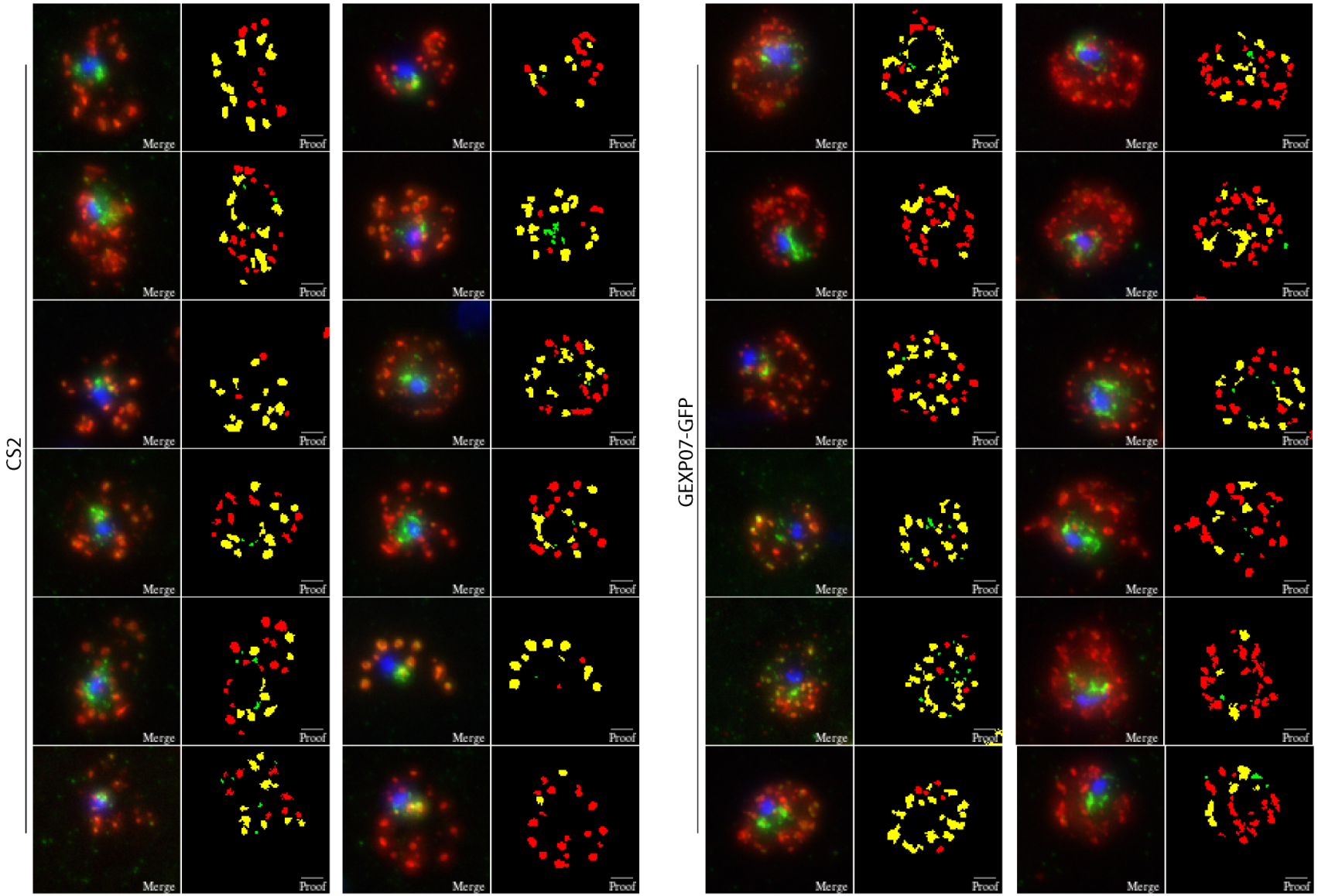
'rawimage'



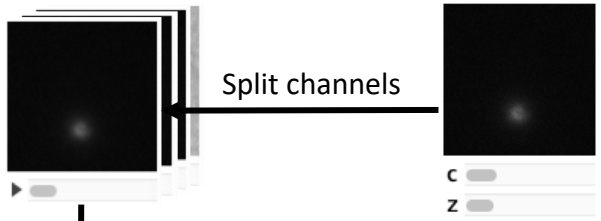
```
//run("Merge Channels...", "c1=REX1mask c2=EMP1mask c7=Both create keep");
run("Merge Channels...", "c1=REX1only c2=EMP1only c7=Both create keep");
run("Stack to RGB");
rename("sortedmask");
run("Select All");
setForegroundColor(255, 255, 255);
run("Draw", "slice");
run("Combine...", "stack1=rawimage stack2=sortedmask");

//Draw scale bar and labels
run("Scale Bar...", "width=2 height=1 font=6 color=Gray background=None location=[Lower Right] hide overlay");
setFont("Serif",8 , "antiliased");
setColor(255, 255, 255);
x=75; y=100;
drawString("Merge", x, y);
setColor(255, 255, 255);
x=180; y=100;
drawString("Proof", x, y);
run("Flatten");

saveAs("tif", outdir+nameonly+"_count_proof");
close();

//clear up workspace, memory
run("Close All");
run("Collect Garbage");
```



Co-occurrence_EMP1vREX1.ijm proofs



“Co_occurrence_EMP1vREX1_MFI.ijm” Macro Workflow



Project z-stack (MIP)

Mask generation

‘REX1_MIP’

Auto local threshold

local threshold

Blur

local AND blur

AND ‘cellmask’

Add; Blur

Binary =‘cellmask’

‘EMP1_MIP’

local threshold

largest particle

Blur; ‘parasite’

Auto local threshold

‘cellmask’

AND ‘cytoplasm’

‘REX1mask’

‘cytoplasm’

‘cytoplasm-clefts’

AND ‘cytoplasm’

‘EMP1mask’

Measure fluorescence of EMP1_MIP reference for each mask;
Summarize results in table

EMP1 fluorescence measurements

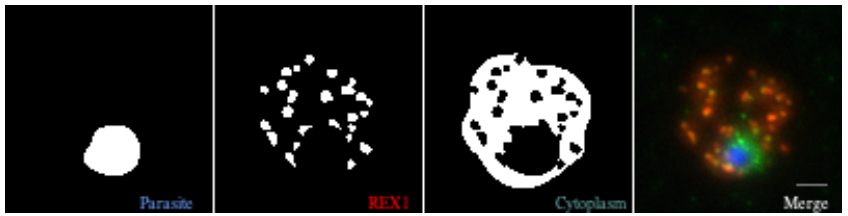
‘EMP1_MIP’

Summary

Slice	Count	Total Area	Average Size	%Area	Mean
6.9.18_CS2_wella1_10_R3D-2.dv_stack_REX1	23	6.692	0.291	4.110	352.432
6.9.18_CS2_wella1_10_R3D-2.dv_stack_para	1	8.369	8.369	5.140	540.782
6.9.18_CS2_wella1_10_R3D-2.dv_stack_cytoplasm	1	34.159	34.159	20.980	247.127

'REX1mask’ 'paramask’ 'cytoplasm-clefts’

Generate proof:
Merge MIP; Laterally combine with masks



```
//get input and output directories
indir = getDirectory("select a source folder");
outdir = getDirectory("select a destination folder");

//get the image list
list = getFileList(indir);
```

Select input and output directories

```
//start batch loop
for(i=0; i<list.length; i++){

    run("Close All");

    //open image, grab the file name
    open(indir + list[i]);
    nameonly = File.nameWithoutExtension;

    //split channels and rename each channels with static name
    run("Split Channels");
    image = "C1-"+ list[i];
    selectWindow(image);
    run("Z Project...", "projection=[Max Intensity]");
    rename("DAPI_MIP");
```

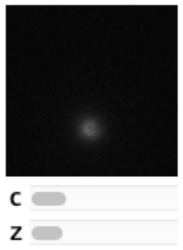
Begin batch loop

Close any open images

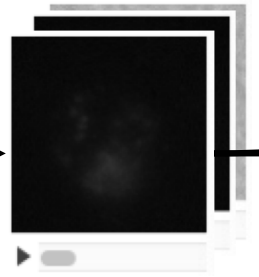
Open image (hyperstack)


```
    image = "C2-"+ list[i];
    selectWindow(image);
    run("Z Project...", "projection=[Max Intensity]");
    rename("EMP1_MIP");

    image = "C3-"+ list[i];
    selectWindow(image);
    run("Z Project...", "projection=[Max Intensity]");
    rename("REX1_MIP");
```

Opened image (hyperstack)

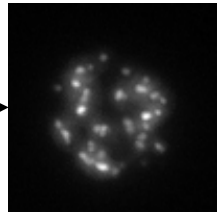


Split channels

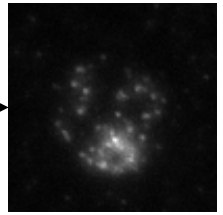


Project z-stack (Maximum Intensity Projection); Rename

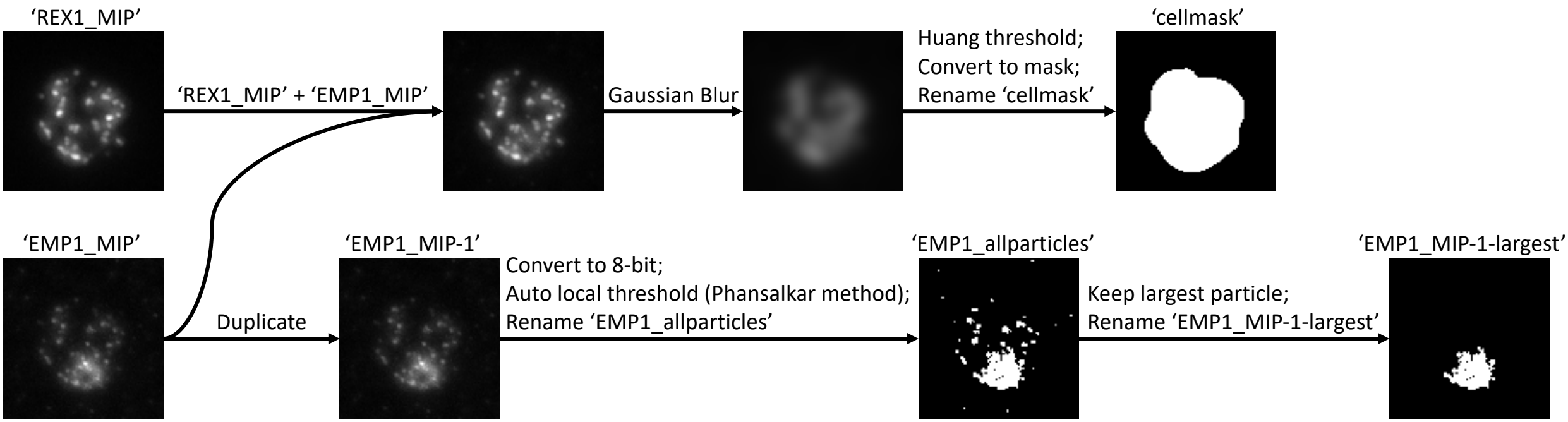
‘REX1_MIP’



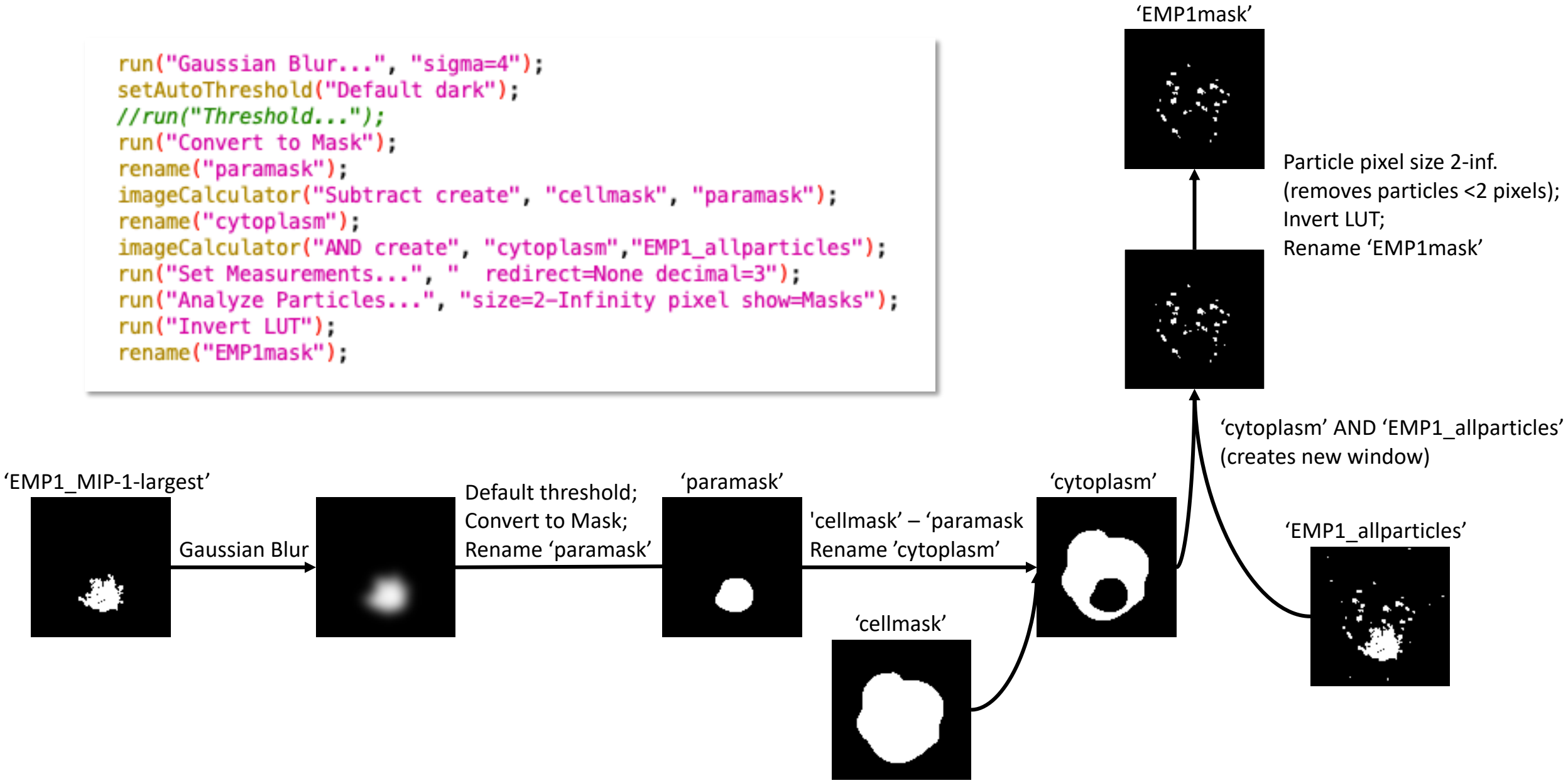
‘EMP1_MIP’



```
//create masks
    //make cell mask w/ gaussian blur
    imageCalculator("Add create", "EMP1_MIP","REX1_MIP");
    run("Gaussian Blur...", "sigma=4");
    setAutoThreshold("Huang dark");
    run("Convert to Mask");
    rename("cellmask");

    //Make parasite mask via EMP1 staining
    selectWindow("EMP1_MIP");
    run("Duplicate...", " ");
    setOption("ScaleConversions", true);
    run("8-bit");
    run("Auto Local Threshold", "method=Phansalkar radius=2 parameter_1=0 parameter_2=0 white");
    rename("EMP1_allparticles");
    run("Keep Largest Region");
    rename("EMP1_MIP-1-largest");
```

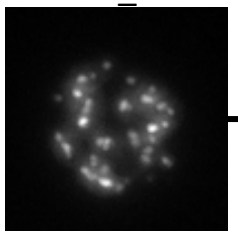


```
run("Gaussian Blur...", "sigma=4");
setAutoThreshold("Default dark");
//run("Threshold...");
run("Convert to Mask");
rename("paramask");
imageCalculator("Subtract create", "cellmask", "paramask");
rename("cytoplasm");
imageCalculator("AND create", "cytoplasm","EMP1_allparticles");
run("Set Measurements...", "  redirect=None decimal=3");
run("Analyze Particles...", "size=2-Infinity pixel show=Masks");
run("Invert LUT");
rename("EMP1mask");
```



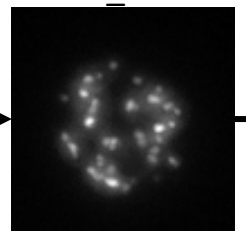
```
//REX1 clefts mask (local threshold)
selectWindow("REX1_MIP");
run("Duplicate...", " ");
setOption("ScaleConversions", true);
run("8-bit");
run("Auto Local Threshold", "method=Niblack radius=5 parameter_1=0 parameter_2=0 white");

rename("REX1mask_clefts");
run("Duplicate...", " ");
run("Gaussian Blur...", "sigma=2");
setAutoThreshold("Default dark");
rename("REX1mask_blur");
//run("Threshold...");
run("Convert to Mask");
imageCalculator("AND create", "REX1mask_clefts","REX1mask_blur");
rename("REX1_binary1");
imageCalculator("AND create", "cellmask","REX1_binary1");
rename("REX1_binary2");
imageCalculator("AND create", "REX1_binary2","cytoplasm");
run("Watershed");
run("Set Measurements...", "  redirect=None decimal=3");
run("Analyze Particles...", "size=5-Infinity pixel show=Masks");
run("Invert LUT");
rename("REX1mask");
```

‘REX1_MIP’



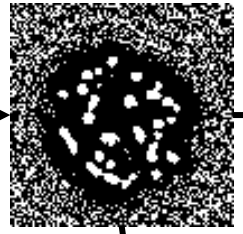
Duplicate

‘REX1_MIP-1’



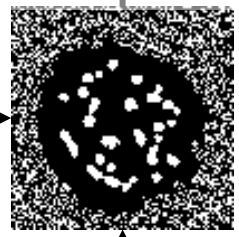
Convert to 8-bit;
Auto local threshold
(Niblack method);
Rename ‘REX1mask_clefts’

‘REX1mask_clefts’



‘REX1mask_clefts’
AND ‘REX1mask_blur’;
Rename ‘REX1_binary1’

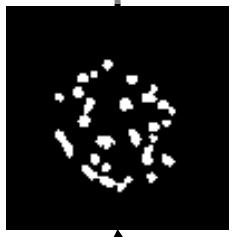
‘REX1_binary1’



‘cellmask’ AND ‘REX1_binary1’;
Rename ‘REX1_binary2’

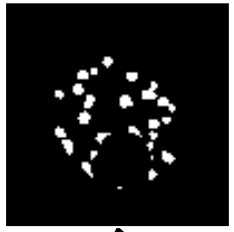
‘cellmask’

‘REX1_binary2’



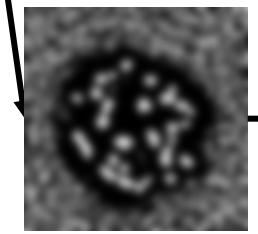
‘cytoplasm’

‘REX1mask’



‘REX1_binary2’ AND ‘cytoplasm’;
Watershed;
Particle pixel size 5-inf.
(remove particles >5 pixels);
Invert LUT;
Rename ‘REX1mask’

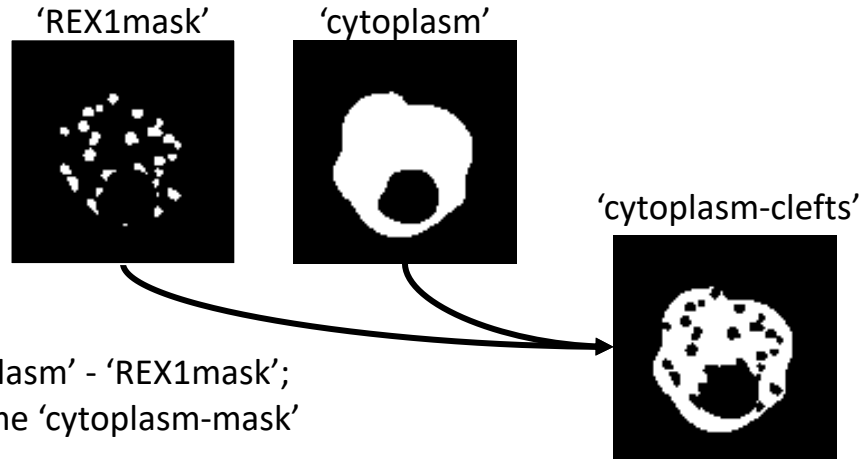
Duplicate;
Gaussian Blur



Default threshold;
Rename
‘REX1mask_blur’

‘REX1mask_blur’

```
imageCalculator("Subtract create", "cytoplasm","REX1mask");
rename("cytoplasm-clefts");
```

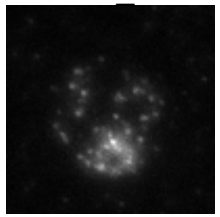


'cytoplasm' - 'REX1mask';
Rename 'cytoplasm-mask'

```
//measuring MFI and StDev
//measure REX1mask
selectWindow("REX1mask");
rename(nameonly+"_REX1");
run("Set Measurements...", "area mean redirect=EMP1_MIP decimal=3");
run("Analyze Particles...", "size=0-Infinity pixel show=Nothing summarize");
rename(nameonly+"REX1mask_stack_");

//measure paramask
selectWindow("paramask");
rename(nameonly+"_para");
run("Analyze Particles...", "size=0-Infinity pixel show=Nothing summarize");
rename(nameonly+"paramask_stack_");

//measure cellmask
selectWindow("cytoplasm-clefts");
rename(nameonly+"_cytoplasm");
run("Analyze Particles...", "size=0-Infinity pixel show=Nothing summarize");
rename(nameonly+"cytoplasm_stack_");
```

Measure EMP1_MIP reference for each mask;
Summarize results

'EMP1_MIP'

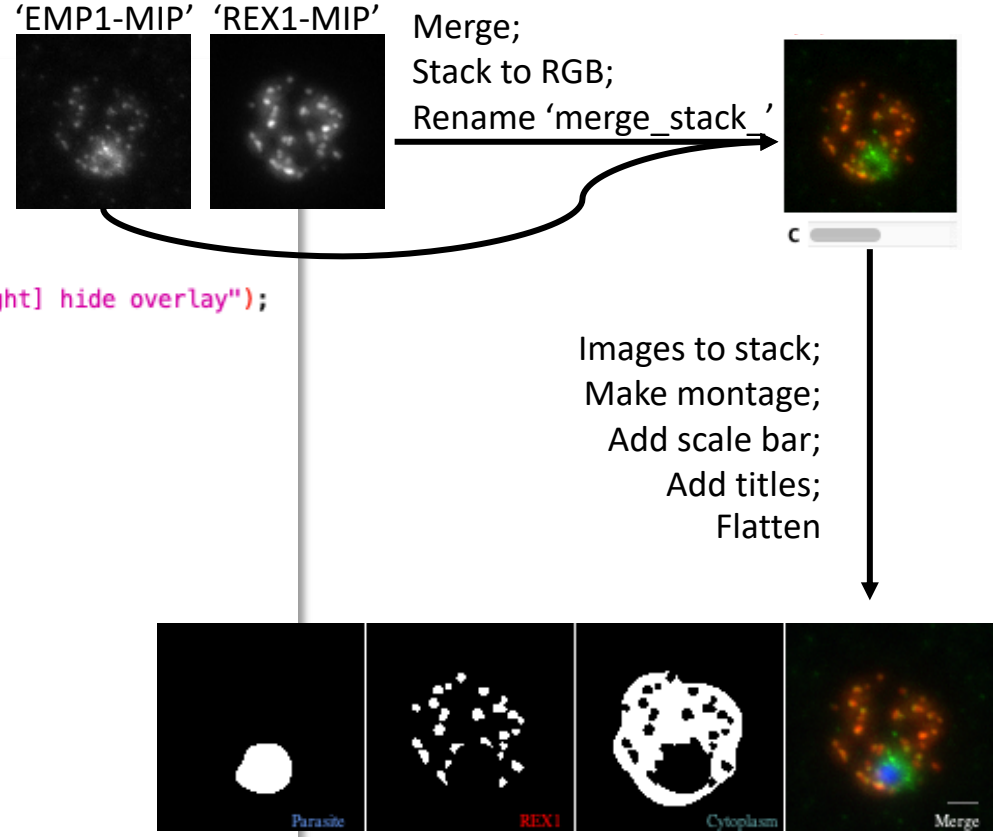


Summary

Slice	Count	Total Area	Average Size	%Area	Mean
6.9.18_CS2_wella1_10_R3D-2.dv_stack_REX1	23	6.692	0.291	4.110	352.432
6.9.18_CS2_wella1_10_R3D-2.dv_stack_para	1	8.369	8.369	5.140	540.782
6.9.18_CS2_wella1_10_R3D-2.dv_stack_cytoplasm	1	34.159	34.159	20.980	247.127

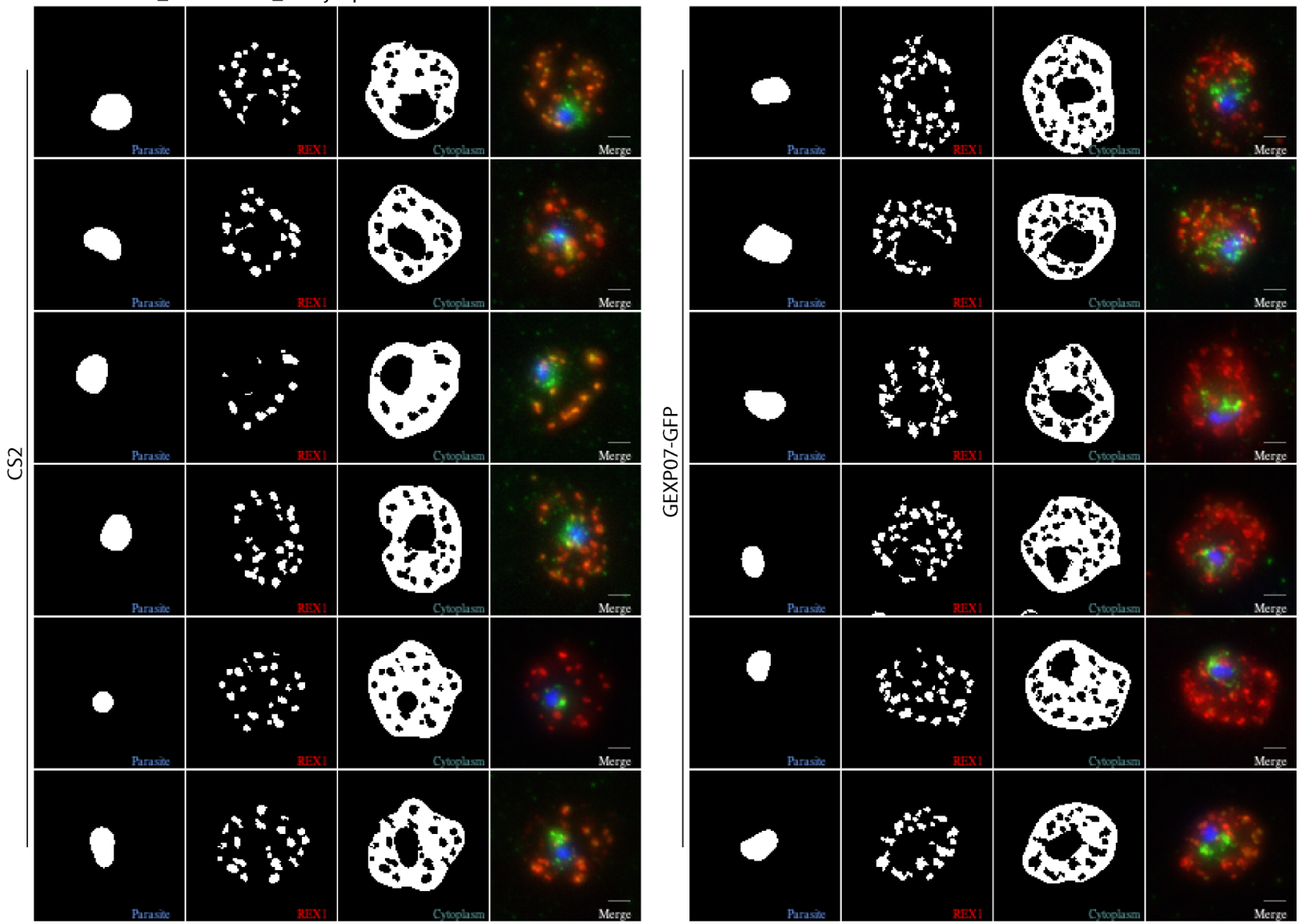
'REX1mask' 'paramask' 'cytoplasm-clefts'

```
    //make 'measure' proof
    run("Merge Channels...", "c1=REX1_MIP c2=EMP1_MIP c3=DAPI_MIP create keep");
    run("Stack to RGB");
    rename("merge_stack_");

    run("Images to Stack", "name=[] title=_stack_ use");
    run("Make Montage...", "columns=4 rows=1 scale=1 border=1");
    run("Scale Bar...", "width=2 height=1 font=6 color=Gray background=None location=[Lower Right] hide overlay");
    setFont("Serif",8 , "antiliased");
    setColor(100,149,237);
    x=65; y=100;
    drawString("Parasite", x, y);
    setColor(255, 0, 0);
    x=175; y=100;
    drawString("REX1", x, y);
    setColor(95,158,160);
    x=265; y=100;
    drawString("Cytoplasm", x, y);
    setColor(255, 255, 255);
    x=375; y=100;
    drawString("Merge", x, y);
    run("Flatten");
    saveAs("tif", outdir+nameonly+"_MFI_proof");

    //clear up workspace, memory
    run("Close All");
    run("Collect Garbage");
}
```



Save proof to ouput directory

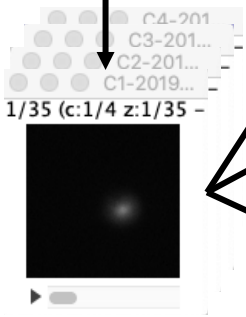
Co-occurrence_EMP1vREX1_MFI.ijm proofs



“Surface_protein_v_cleft_counting” Macro Workflow

Start

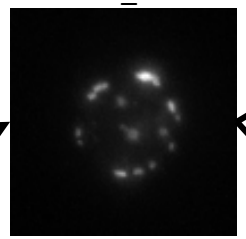
Open image (hyperstack); Split channels



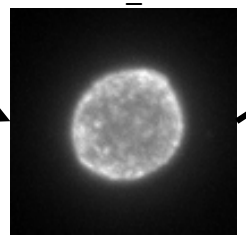
Project Z-stack

‘DAPI_MIP’

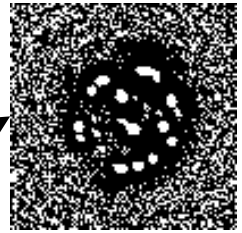
‘REX1_MIP’



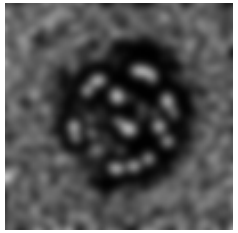
‘EMP3_MIP’



Local threshold

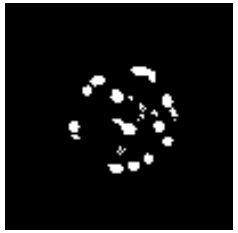


Blur

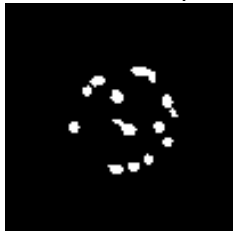


Convert to Mask

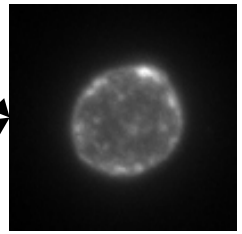
AND



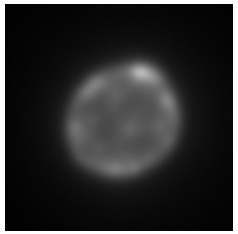
Particles 5-inf. pixels



Add

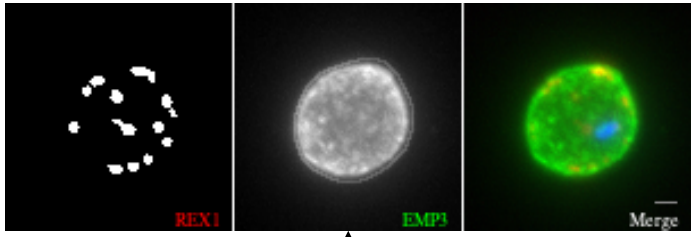


Blur

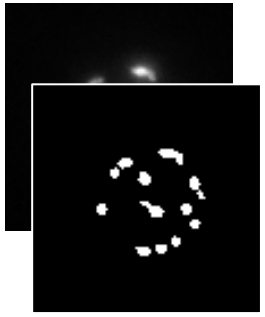


Convert to Mask

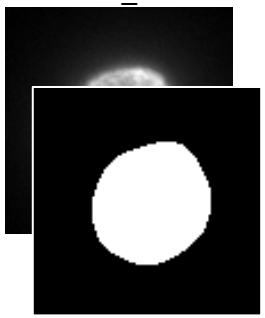
Proof

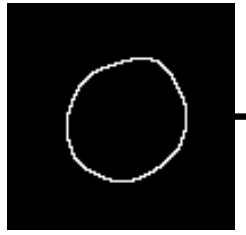


‘REX1_MIP’



‘EMP3_MIP’



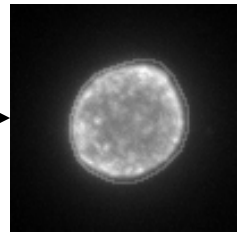


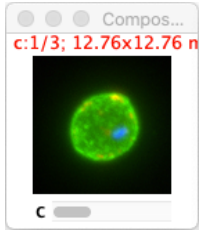
Analyze particles; summarize results; show outline; Invert LUT; Convert to Mask

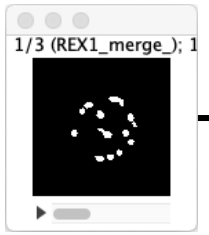
‘EMP3_MIP’ + ‘EMP3outline

Summary

Slice	Count	Total Area	Average Size	%Area	Mean
2019.12.11_rep1_cs2_mEMP3_rREX1_02_R3D-2.dv_stackREX1mask	12	4.608	0.384	2.830	750.700
2019.12.11_rep1_cs2_mEMP3_rREX1_02_R3D-2.dv_stackEMP3mask	1	35.738	35.738	21.950	1272.072







Surface_protein_v_cleft_counting.ijm

```
1  //BEFORE RUNNING, CHECK THAT SURFACE PROTEIN IS CORRECT (command+F)
2  //get input and output directories
3  indir = getDirectory("select a source folder");
4  outdir = getDirectory("select a destination folder");
5
6  //get the image list
7  list = getFileList(indir);
8
9  //start batch loop
10 for(i=0; i<list.length; i++){
11
12     run("Close All");
13
14     //open image, grab the file name
15     open(indir + list[i]);
16     nameonly = File.nameWithoutExtension;
17
18     //split channels and rename each channels with static name
19     run("Split Channels");
20
21     image = "C1-"+ list[i];
22     selectWindow(image);
23     run("Z Project...", "projection=[Max Intensity]");
24     rename("DAPI_MIP");
25
26     image = "C2-"+ list[i];
27     selectWindow(image);
28     run("Z Project...", "projection=[Max Intensity]");
29     rename("REX1_MIP");
30
31     image = "C3-"+ list[i];
32     selectWindow(image);
33     run("Z Project...", "projection=[Max Intensity]");
34     rename("EMP3_MIP");
35
```

Select input and output directories

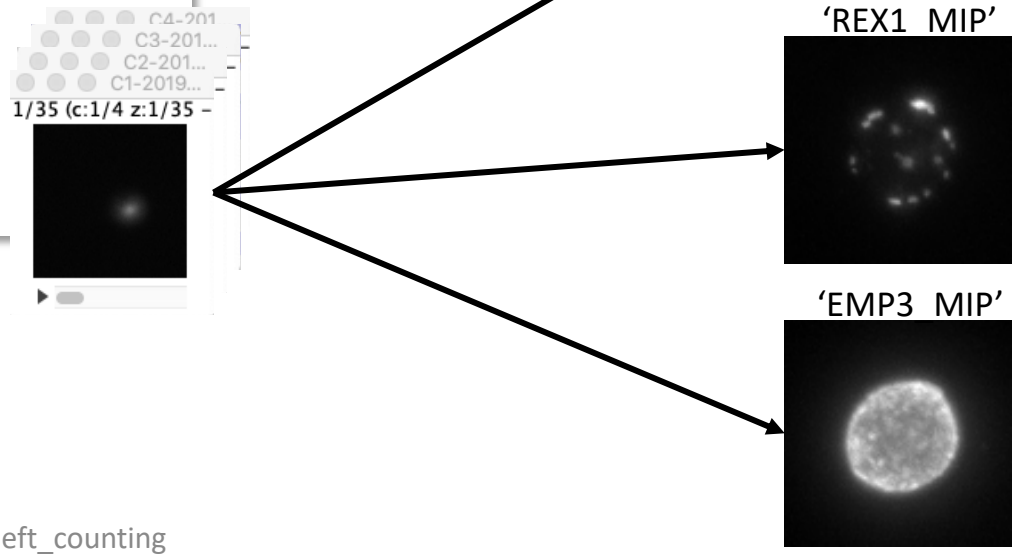
Begin batch loop

Close any open images

Open image (hyperstack)

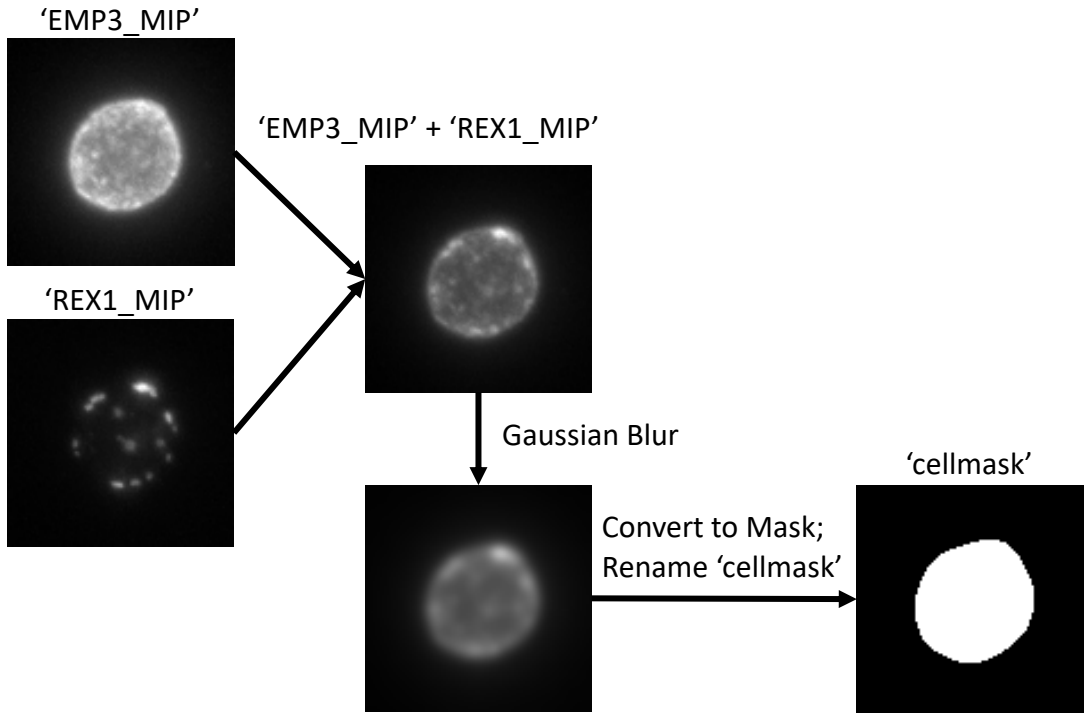
Split channels

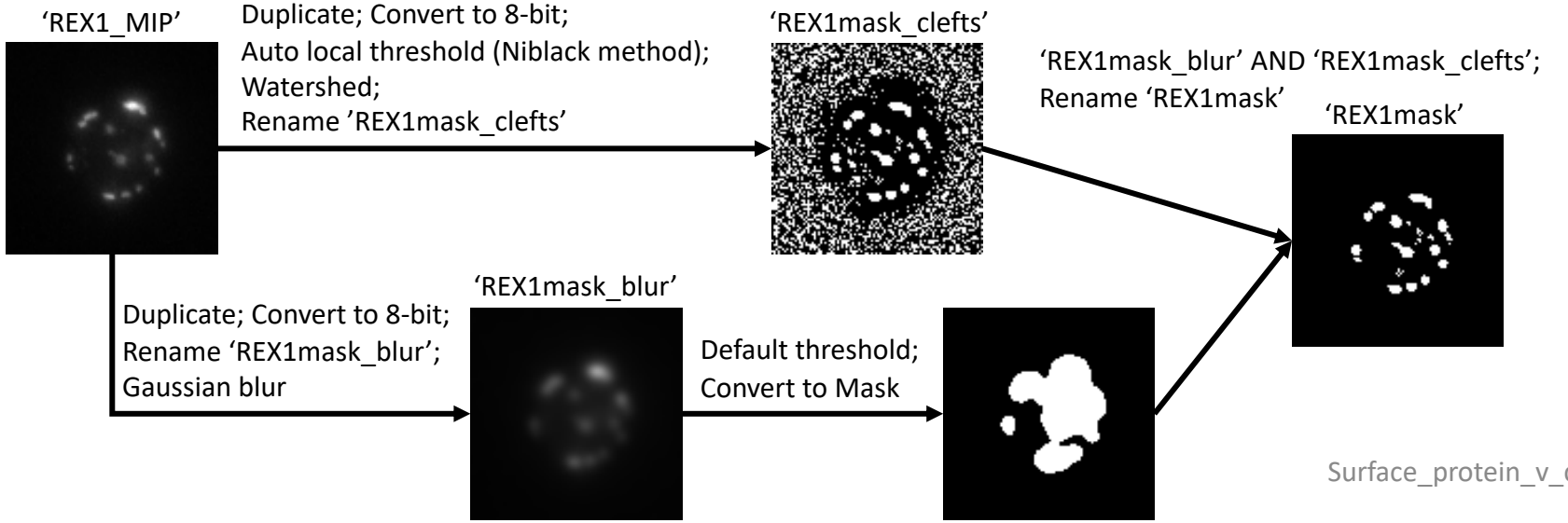
Project z-stack
(Maximum Intensity Projection);
Rename



Surface_protein_v_cleft_counting.ijm

```
36      //create masks
37          //make cell mask w/ gaussian blur
38          imageCalculator("Add create", "EMP3_MIP","REX1_MIP");
39          run("Gaussian Blur...", "sigma=2");
40          setAutoThreshold("Default dark");
41          run("Convert to Mask");
42          rename("cellmask");
43
44          //REX1 clefts mask (local threshold)
45          selectWindow("REX1_MIP");
46          run("Duplicate...", " ");
47          setOption("ScaleConversions", true);
48          run("8-bit");
49          run("Auto Local Threshold", "method=Niblack radius=5 parameter_1=0 parameter_2=0 white");
50          run("Watershed");
51          rename("REX1mask_clefts");
52
53          //REX1 clefts mask (default threshold to remove noise from local threshold)
54          selectWindow("REX1_MIP");
55          run("Duplicate...", " ");
56          run("8-bit");
57          rename("REX1mask_blur");
58          run("Gaussian Blur...", "sigma=2");
59          setAutoThreshold("Default dark");
60          //run("Threshold...");
61          run("Convert to Mask");
62          imageCalculator("AND create", "REX1mask_clefts","REX1mask_blur");
63          selectWindow("Result of REX1mask_clefts");
64          rename("REX1mask");
65
```





Surface_protein_v_cleft_counting.ijm

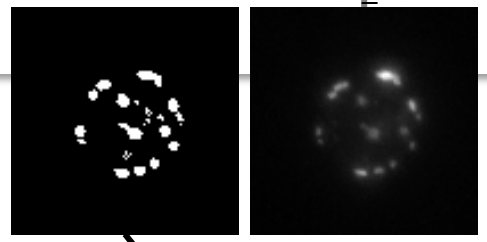
```
65
66      //on the original image, make a binary, watershed, analyse particles
67      selectWindow("REX1mask");
68      rename(nameonly+"REX1mask");
69      run("Set Measurements...", "area mean redirect=REX1_MIP decimal=3");
70      run("Analyze Particles...", "size=10-Infinity pixel show=[Masks] summarize");
71      run("Invert LUT");
72      run("Convert to Mask");
73      rename("REX1_merge_");
74
75      selectWindow("cellmask");
76      rename(nameonly+"EMP3mask");
77      run("Set Measurements...", "area mean redirect=EMP3_MIP decimal=3");
78      run("Analyze Particles...", "show=[Bare Outlines] summarize");
79      run("Invert LUT");
80      run("Convert to Mask");
81      rename("EMP3outline");
82      imageCalculator("Add", "EMP3_MIP","EMP3outline");
83      rename("EMP3_merge_");
84
```

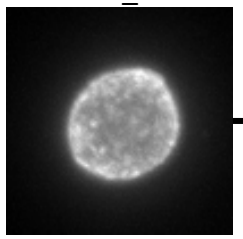
Summary

Slice	Count	Total Area	Average Size	%Area	Mean
2019.12.11_rep1_cs2_mEMP3_rREX1_02_R3D-2.dv_stackREX1mask	12	4.608	0.384	2.830	750.700
2019.12.11_rep1_cs2_mEMP3_rREX1_02_R3D-2.dv_stackEMP3mask	1	35.738	35.738	21.950	1272.072

Set measurements; Analyze particles

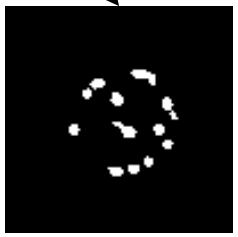
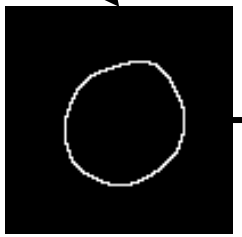
‘REX1mask’ ‘REX1_MIP’ ‘cellmask’ ‘EMP3_MIP’



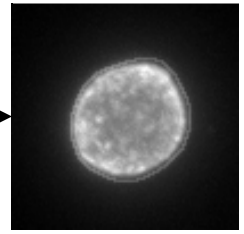
Analyze particles 5-inf. Pixels;
show mask; Invert LUT; Convert to Mask;
Rename ‘REX1_merge_’

Analyze particles; show outline;
Invert LUT; Convert to Mask;
Rename ‘EMP3outline’

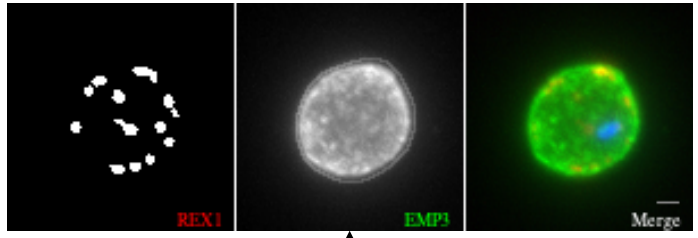



‘EMP3_MIP’
+ ‘EMP3outline;
Rename ‘EMP3_merge’

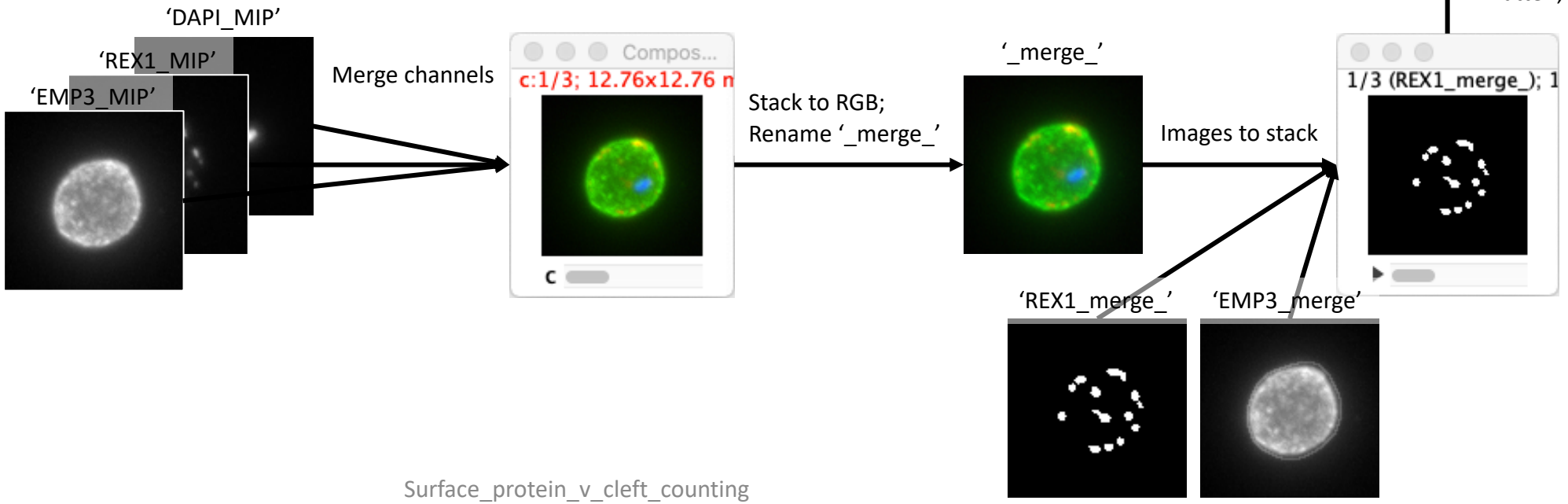
‘EMP3_merge’



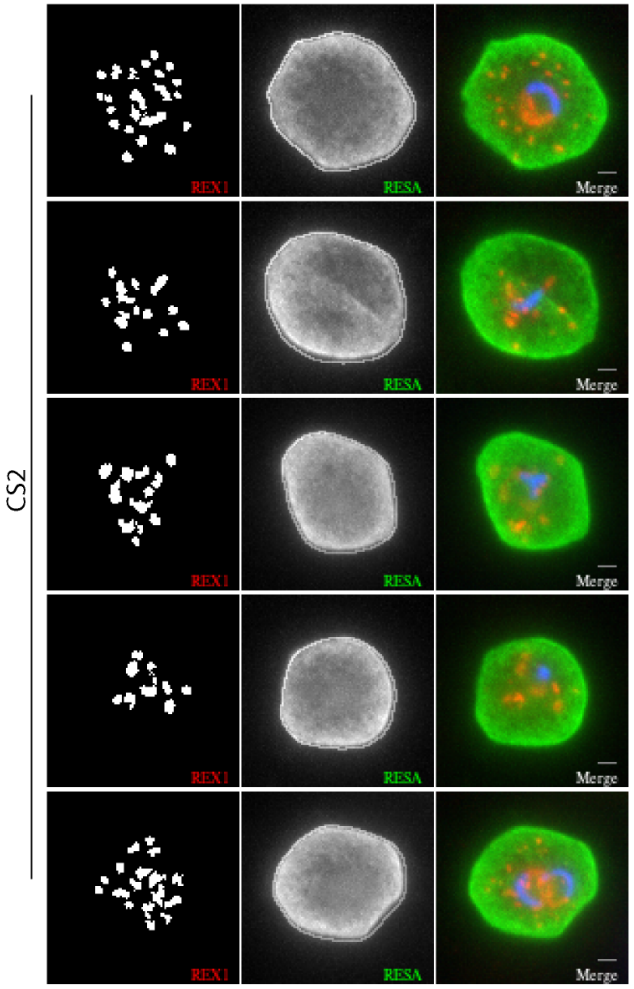
```
        //make proof
        run("Merge Channels...", "c1=REX1_MIP c2=EMP3_MIP c3=DAPI_MIP create keep");
        run("Stack to RGB");
        rename("_merge_");

        run("Images to Stack", "name=[] title=_merge_ use");
        run("Make Montage...", "columns=3 rows=1 scale=1 border=1");
        run("Scale Bar...", "width=10 height=1 font=6 color=Gray background=None location=[Lower Right] hide overlay");
        setFont("Serif",8 , "antiliased");
        setColor(0, 255, 0);
        x=175; y=100;
        drawString("EMP3", x, y);
        setColor(255, 0, 0);
        x=75; y=100;
        drawString("REX1", x, y);
        setColor(255, 255, 255);
        x=275; y=100;
        drawString("Merge", x, y);
        run("Flatten");
        saveAs("tif", outdir+nameonly+"_proof");

        //clear up workspace, memory
        run("Close All");
        run("Collect Garbage");
    }
```

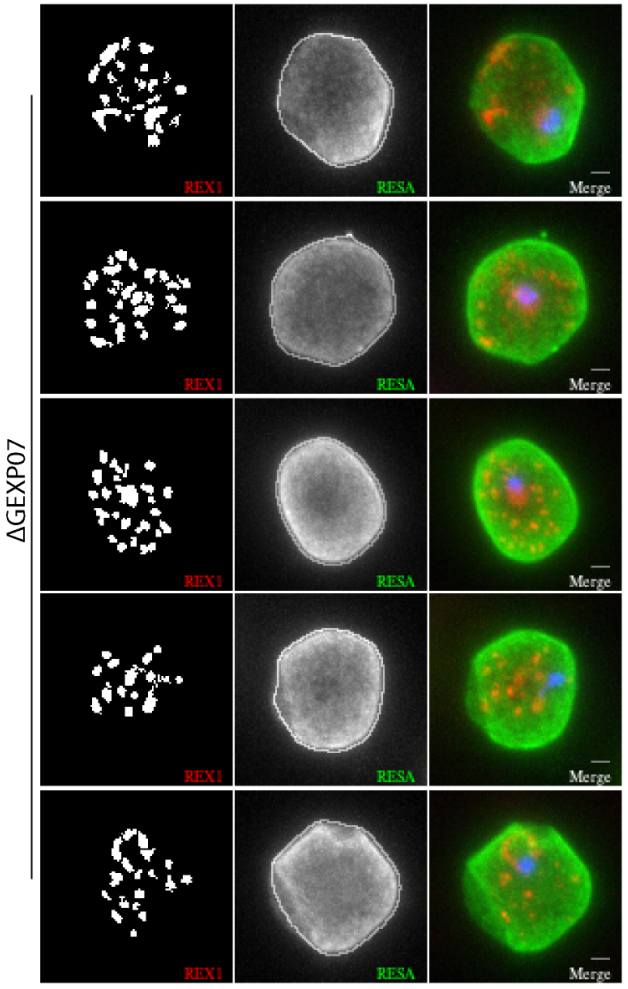


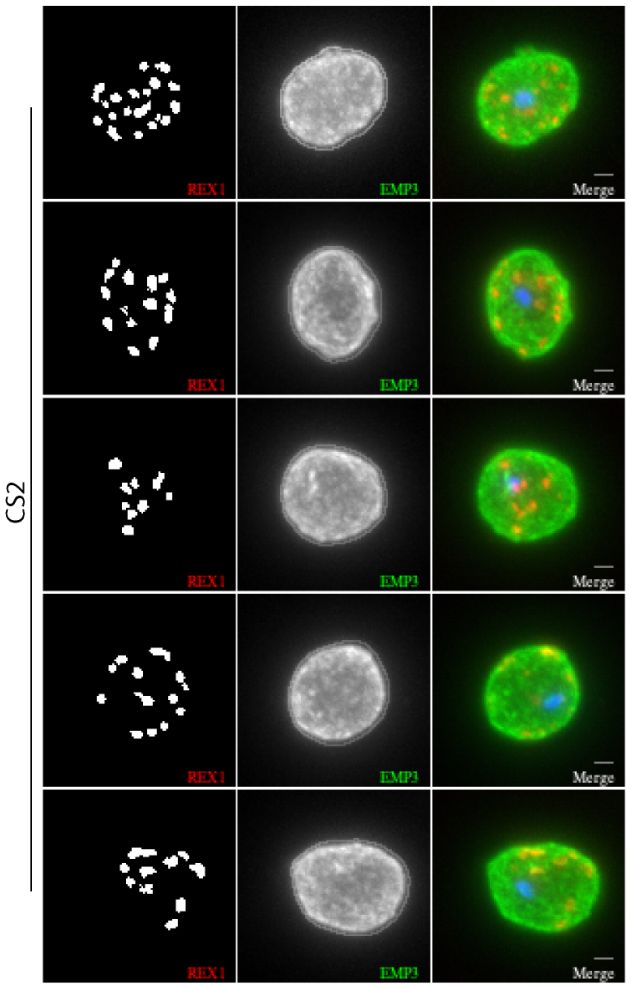
Stack to Montage;
Print scale bar;
Print labels;
Flatten; Save

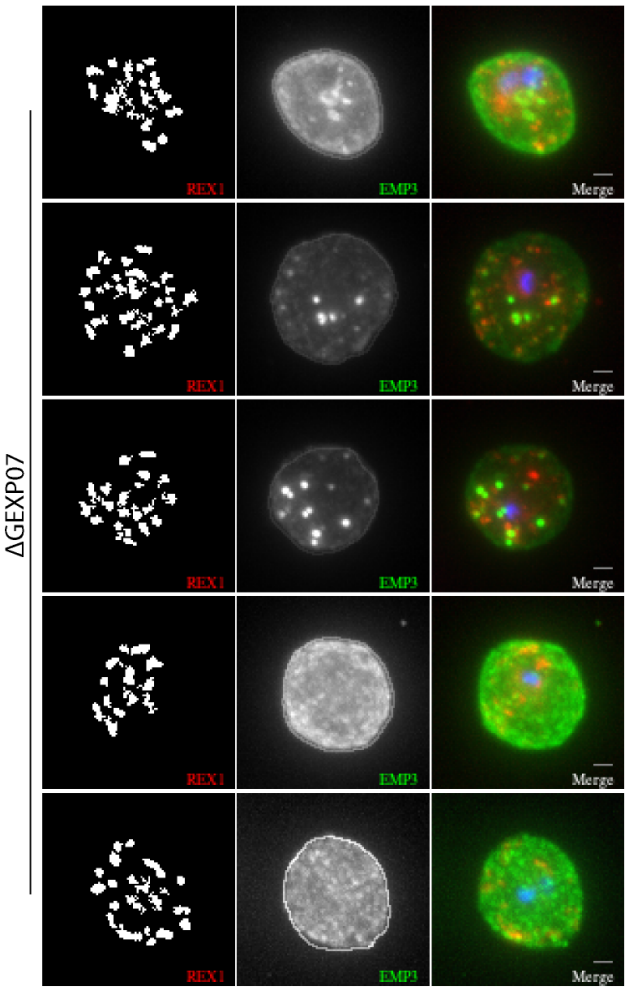


Surface_protein_v_cleft_counting.ijm proofs







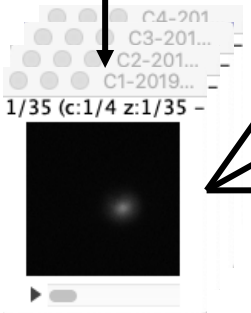


“Surface_protein_v_surface_protein .ijm” Macro

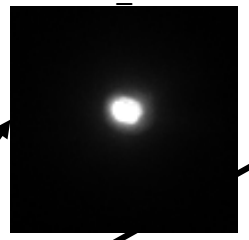
Mask generation

Start

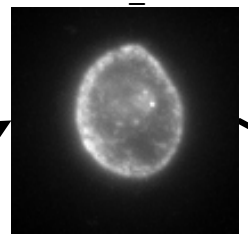
Open image (hyperstack); Split channels



‘DAPI_MIP’

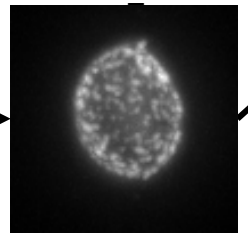


‘MESA_MIP’

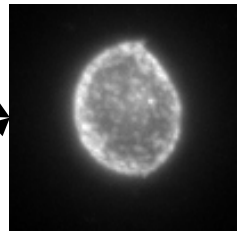


Project Z-stack

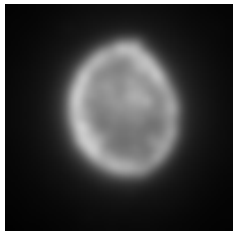
‘KAHRP_MIP’



Add



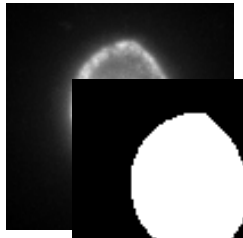
Blur



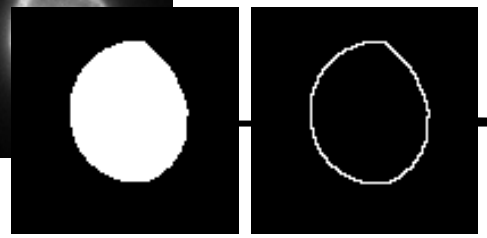
Convert to Mask

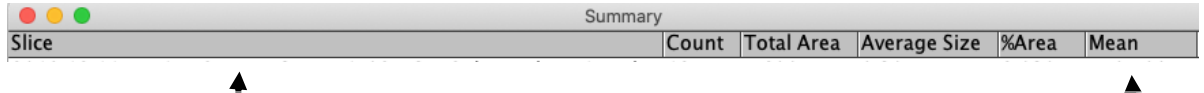
Measurements

‘MESA_MIP’

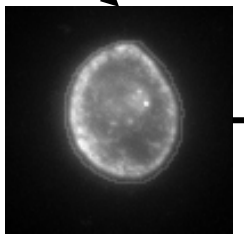


Analyze particles; summarize results; show outline; Invert LUT; Convert to Mask

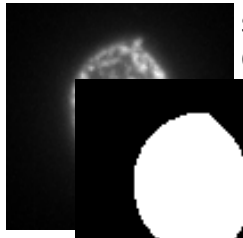




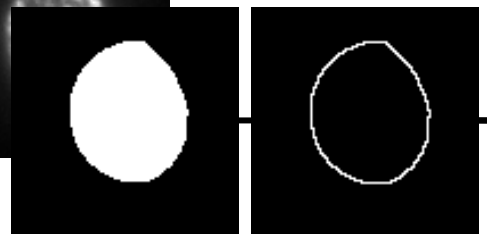
‘MESA_MIP’ + ‘MESAoutline



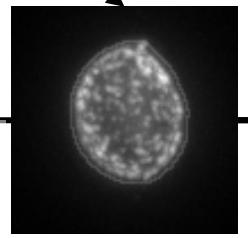
‘KAHRP_MIP’



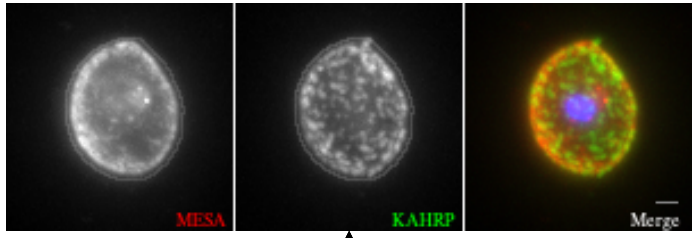
Analyze particles; summarize results; show outline; Invert LUT; Convert to Mask



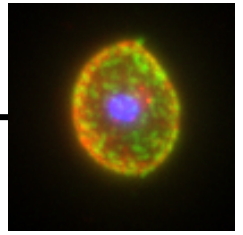
‘KAHRP_MIP’ + ‘KAHRPoutline



Proof



‘_merge_’



Surface_protein_v_surface_protein.ijm

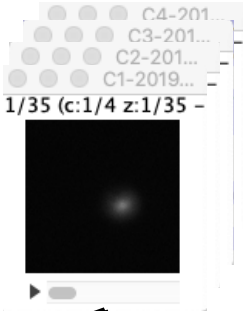
```
//BEFORE RUNNING, CHECK THAT SURFACE PROTEIN IS CORRECT (command+F)
//get input and output directories
indir = getDirectory("select a source folder");
outdir = getDirectory("select a destination folder");

//get the image list
list = getFileList(indir);

//start batch loop
for(i=0; i<list.length; i++){

    run("Close All");

    //open image, grab the file name
    open(indir + list[i]);
    nameonly = File.nameWithoutExtension;

    //split channels and rename each channels with static name
    run("Split Channels");

    image = "C1-"+ list[i];
    selectWindow(image);
    run("Z Project...", "projection=[Max Intensity]");
    rename("DAPI_MIP");

    image = "C2-"+ list[i];
    selectWindow(image);
    run("Z Project...", "projection=[Max Intensity]");
    rename("MESA_MIP");

    image = "C3-"+ list[i];
    selectWindow(image);
    run("Z Project...", "projection=[Max Intensity]");
    rename("KAHRP_MIP");
```

Select input and output directories

Begin batch loop

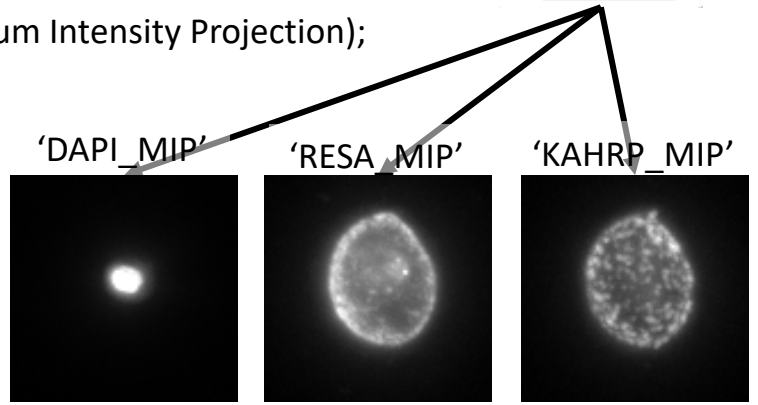
Close any open images

Open image (hyperstack)

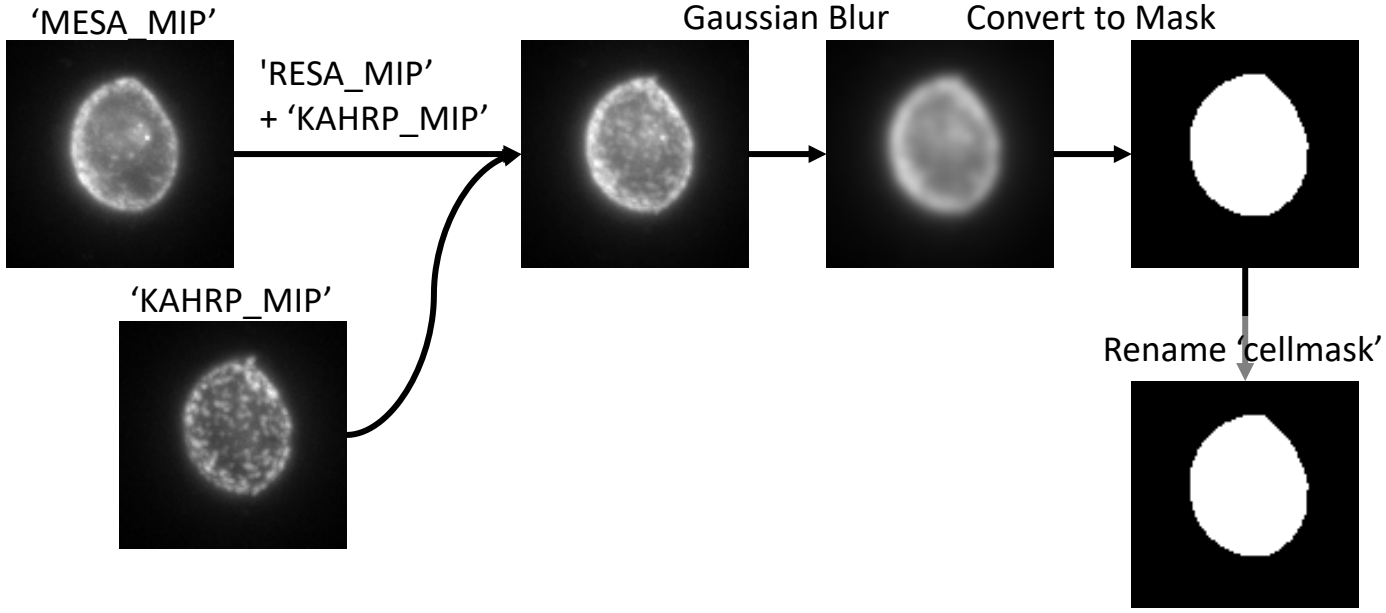


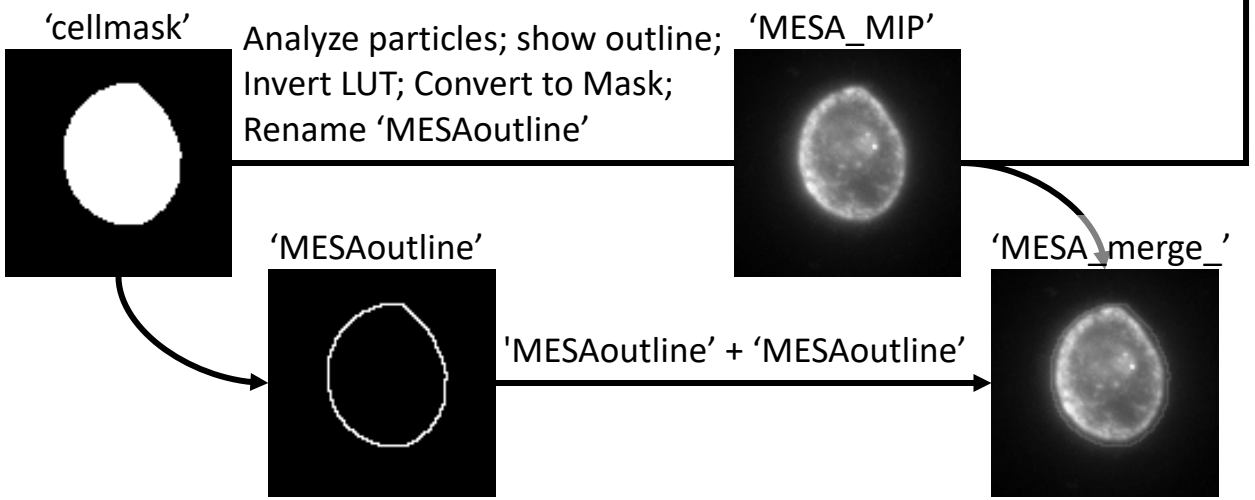
Split channels

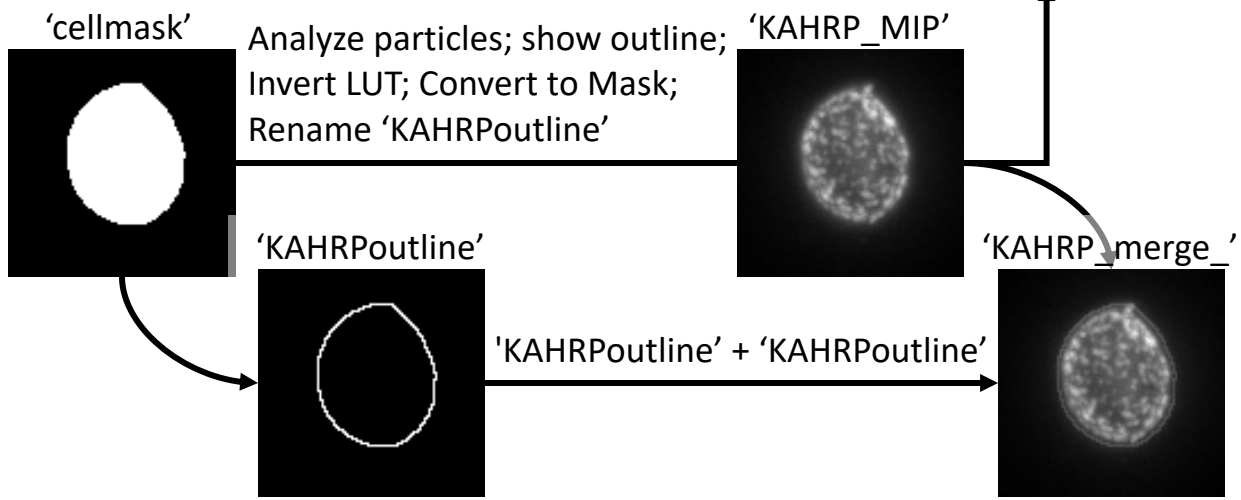
Project z-stack
(Maximum Intensity Projection);
Rename



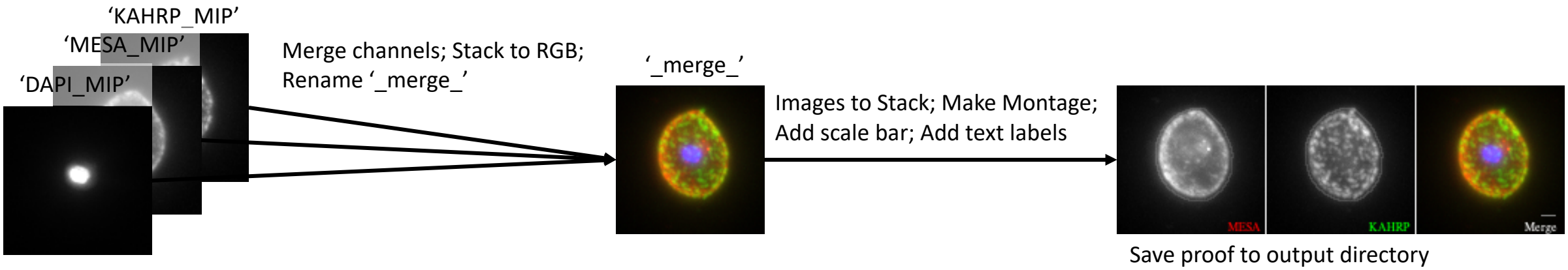
```
//create masks
    //make cell mask w/ gaussian blur
    imageCalculator("Add create", "MESA_MIP","KAHRP_MIP");
    run("Gaussian Blur...", "sigma=2");
    setAutoThreshold("Default dark");
    run("Convert to Mask");
    rename("cellmask");

//on the original image, make a binary, watershed, analyse particles
selectWindow("cellmask");
run("Duplicate...", " ");
rename(nameonly+"_MESAmeasure");
run("Set Measurements...", "area mean redirect=MESA_MIP decimal=3");
run("Analyze Particles...", "show=[Bare Outlines] summarize");
run("Invert LUT");
run("Convert to Mask");
rename("MESAoutline");
imageCalculator("Add", "MESA_MIP","MESAoutline");
rename("MESA_merge_");

selectWindow("cellmask");
run("Duplicate...", " ");
rename(nameonly+"_KAHRPmeasure");
run("Set Measurements...", "area mean redirect=KAHRP_MIP decimal=3");
run("Analyze Particles...", "show=[Bare Outlines] summarize");
run("Invert LUT");
run("Convert to Mask");
rename("KAHRPoutline");
imageCalculator("Add", "KAHRP_MIP","KAHRPoutline");
rename("KAHRP_merge_");
```





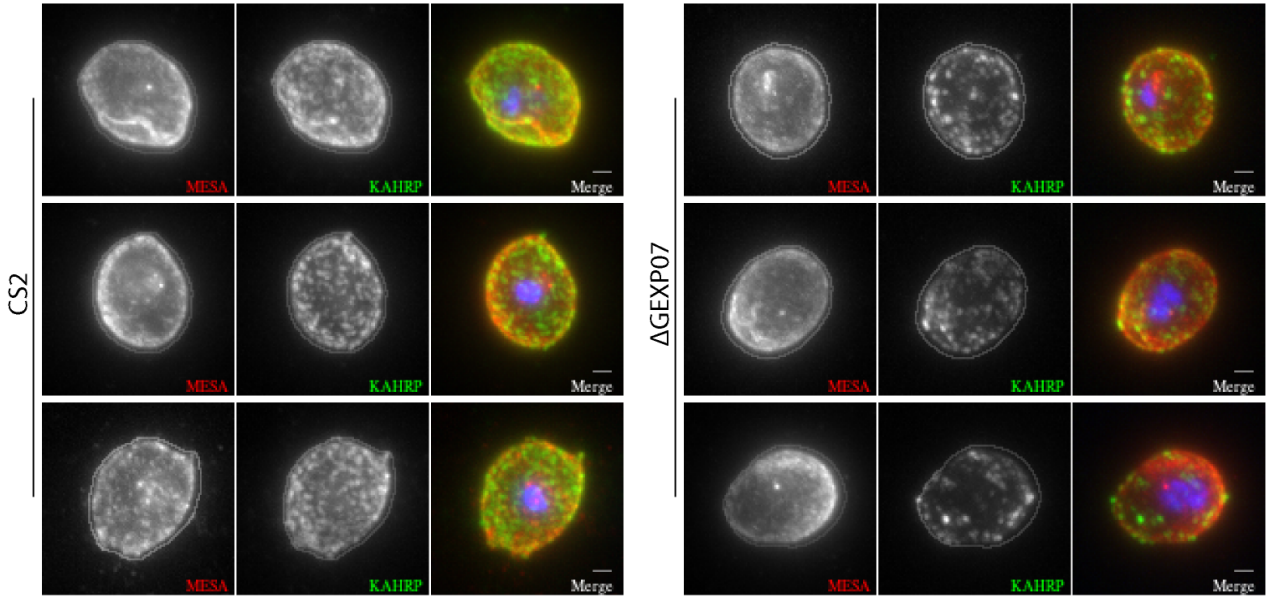


```
    //make proof
    run("Merge Channels...", "c1=MESA_MIP c2=KAHRP_MIP c3=DAPI_MIP create keep");
    run("Stack to RGB");
    rename("_merge_merge_");

    run("Images to Stack", "name=[] title=_merge_ use");
    run("Make Montage...", "columns=3 rows=1 scale=1 border=1");
    run("Scale Bar...", "width=10 height=1 font=6 color=Gray background=None location=[Lower Right] hide overlay");
    setFont("Serif",8 , "antiliased");
    setColor(0, 255, 0);
    x=170; y=100;
    drawString("KAHRP", x, y);
    setColor(255, 0, 0);
    x=75; y=100;
    drawString("MESA", x, y);
    setColor(255, 255, 255);
    x=275; y=100;
    drawString("Merge", x, y);
    run("Flatten");
    saveAs("tif", outdir+nameonly+"_proof");

    //clear up workspace, memory
    run("Close All");
    run("Collect Garbage");
}
```



Save proof to output directory

Surface_protein_v_surface_protein.ijm proofs

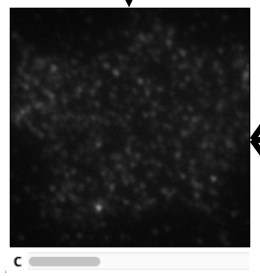


“Sheared_membrane_masks.ijm” Macro Workflow

Mask generation

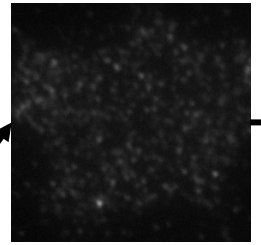
Start

Open image



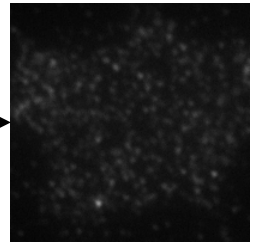
Split channels

C1

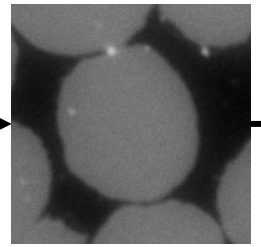


Duplicate;
Rename ‘_GFP’

‘_GFP’



C2



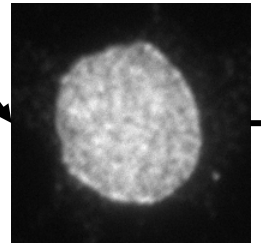
Duplicate; Huang threshold;
Convert to binary;
Rename ‘cellmembranes’

‘cellmembranes’

Invert LUT; Convert to mask;
Rename ‘_Background’

‘_Background’

C3



Duplicate; Huang threshold;
Convert to binary;
Rename ‘_iRBC’

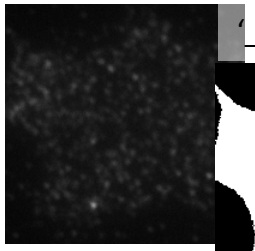
‘_iRBC’

Subtract

‘_uRBC’

Measurements

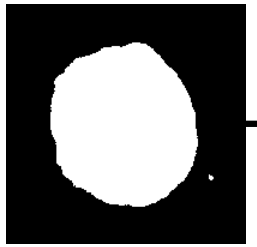
‘_GFP’



‘_Background’

‘_uRBC’

‘_iRBC’

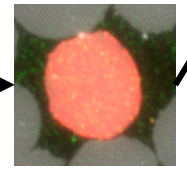


Summary

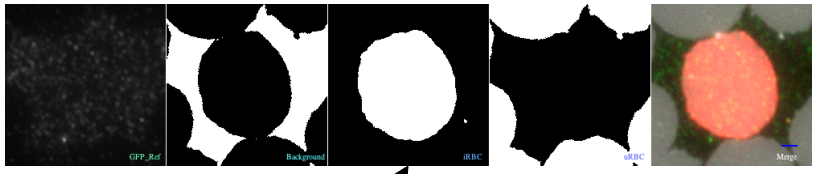
Slice	Count	Total Area	Average Size	%Area	Mean	Median
OC30_30_R3D-1.dv_stack_Background	6	53.884	8.981	33.095	24.393	24.000
OC30_30_R3D-1.dv_stack_uRBC	6	59.388	9.898	36.475	14.889	14.333
OC30_30_R3D-1.dv_stack_iRBC	2	53.543	26.771	32.885	43.251	46.500

Measure masks, redirecting
to reference image ’_GFP’

Merge C1,C2,C3



Make montage with masks

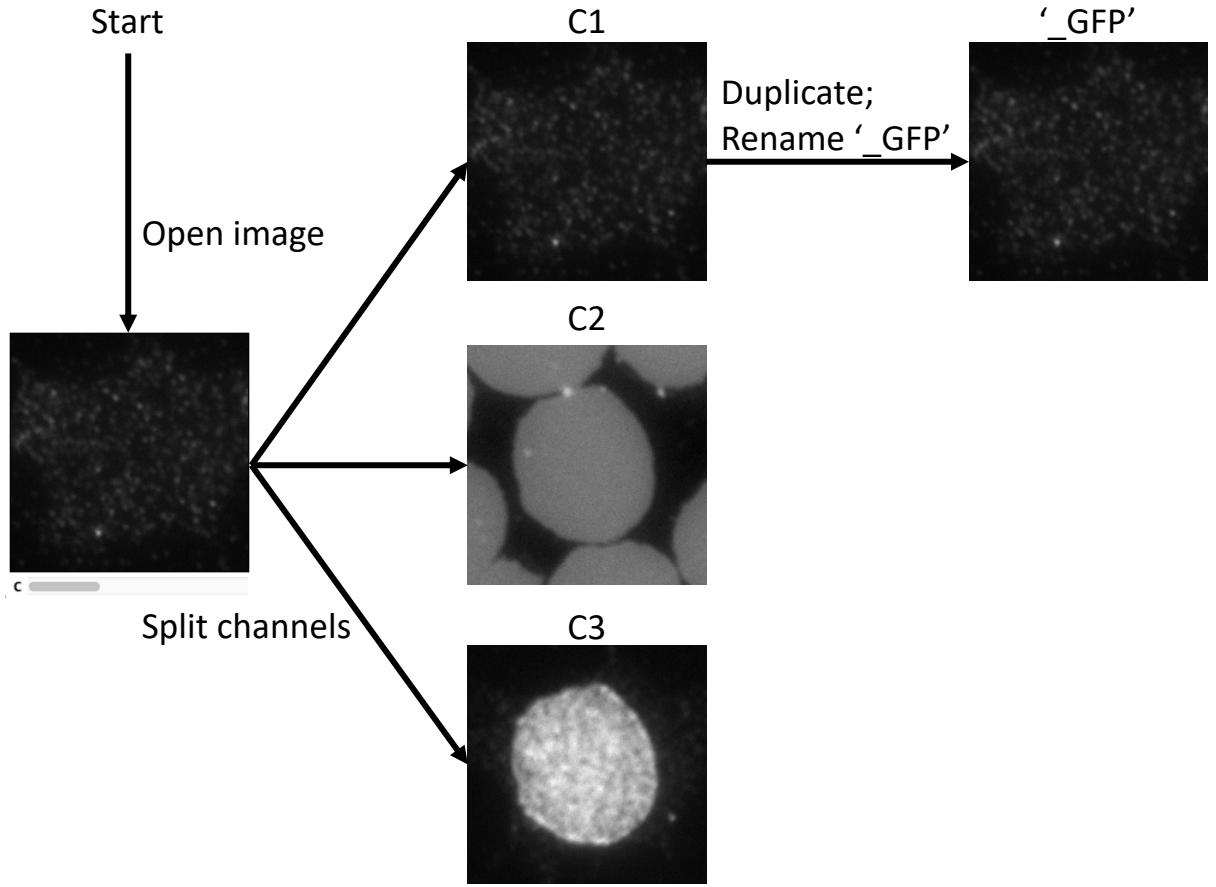


Sheared_membrane_masks.ijm

```
indir=getDirectory("Choose a Directory");
outdir=getDirectory("Choose a Directory");

//retrieving stacks, splitting, and merging z-projections
list1=getFileList(indir);
list2=Array.copy(list1);
a=0;
for (i = 0; i < list1.length; i++) {
    if (endsWith(list1[i], "stack.tif")) {
    list2[a]=list1[i];a++;
    //Array.print(list2);
        }
    }
list2=Array.trim(list2, a);
for (b = 0; b < list2.length; b++) {
    run("Close All");
    open(indir+list2[b]);
    nameonly = File.nameWithoutExtension;

    //split channels and rename each channels with static name
    run("Split Channels");
    C1 = "C1-"+ list2[b];
    C2 = "C2-"+ list2[b];
    C3 = "C3-"+ list2[b];

    //convert reference to 8-bit
    selectWindow(C1); rename("C1");
    run("Duplicate...", " ");
    setOption("ScaleConversions", true);
    run("8-bit");
    rename("_GFP");

```

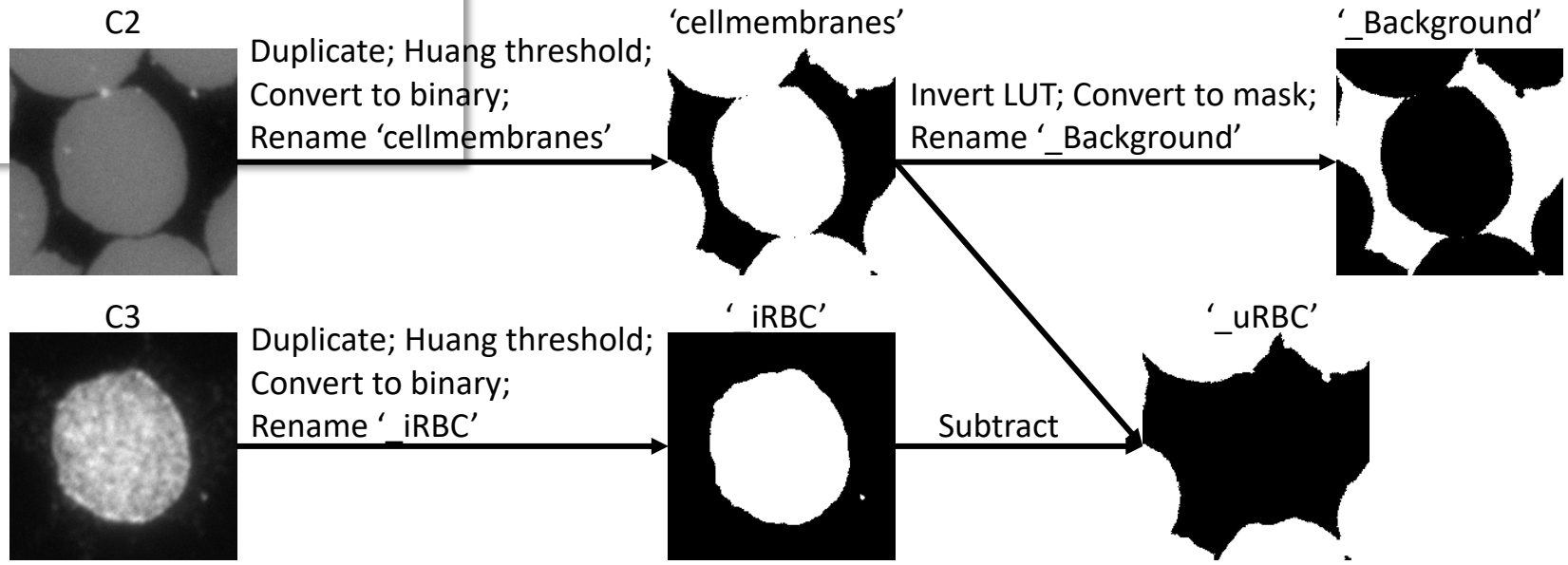
Establish input and output directories

Initialize ‘for’ loop



Sheared_membrane_masks.ijm

```
32
33    //Threshold and make masks
34    selectWindow(C2); rename("C2");
35    run("Duplicate...", " ");
36    setAutoThreshold("Huang dark");
37    setOption("BlackBackground", true);
38    run("Make Binary");
39    rename("cellmembranes");
40
41    selectWindow(C3); rename("C3");
42    run("Duplicate...", " ");
43    setAutoThreshold("Huang dark");
44    run("Make Binary");
45    rename("_iRBC");
46
47    imageCalculator("Subtract create", "cellmembranes","_iRBC");
48    rename("_uRBC");
49
50    selectWindow("cellmembranes");
51    run("Invert LUT");
52    run("Convert to Mask");
53    rename("_Background");
54
```



Sheared_membrane_masks.ijm

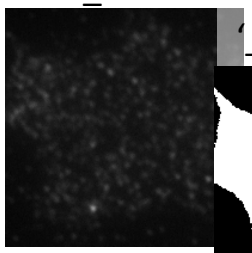
```
54
55  //Measure
56  run("Set Measurements...", "area mean standard median area_fraction limit display redirect=_GFP decimal=3");
57  selectWindow("_Background");
58  rename(nameonly+"_Background");
59  run("Analyze Particles...", "  show=Nothing summarize");
60
61  selectWindow("_uRBC");
62  rename(nameonly+"_uRBC");
63  run("Analyze Particles...", "  show=Nothing summarize");
64
65  selectWindow("_iRBC");
66  rename(nameonly+"_iRBC");
67  run("Analyze Particles...", "  show=Nothing summarize");
68
69      //save images and make proof
70      run("Merge Channels...", "c1=C3 c2=C1 c4=C2 keep");
71      rename("_merge");
72      run("Images to Stack", "name=[] title=_ use");
73      run("Make Montage...", "columns=5 rows=1 scale=1 border=1");
74      run("Scale Bar...", "width=20 height=2 font=6 color=Blue background=None location=[Lower Right] hide overlay");
75
```

```
76      //print label onto proof
77      setFont("Serif",10 , "antiliased");
78      setColor(102,255,178);
79      x=155; y=195;
80      drawString("GFP_Ref", x, y);
81      setColor(102,255,255);
82      x=350; y=195;
83      drawString("Background", x, y);
84      setColor(102,178,255);
85      x=570; y=195;
86      drawString("iRBC", x, y);
87      setColor(102,102,255);
88      x=770; y=195;
89      drawString("uRBC", x, y);
90      setColor(255, 255, 255);
91      x=960; y=195;
92      drawString("Merge", x, y);
93      run("Flatten");
94      saveAs("tif", outdir+nameonly+"_masks_proof");
95
96  run("Close All");
97  run("Collect Garbage");
98  }
99
```

Summary

Slice	Count	Total Area	Average Size	%Area	Mean	Median
OC30_30_R3D-1.dv_stack_Background	6	53.884	8.981	33.095	24.393	24.000
OC30_30_R3D-1.dv_stack_uRBC	6	59.388	9.898	36.475	14.889	14.333
OC30_30_R3D-1.dv_stack_iRBC	2	53.543	26.771	32.885	43.251	46.500

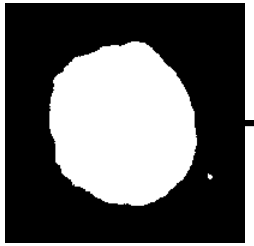
'_GFP'



'_Background'

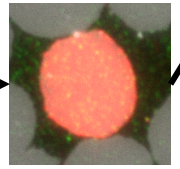
' uRBC'

' iRBC'



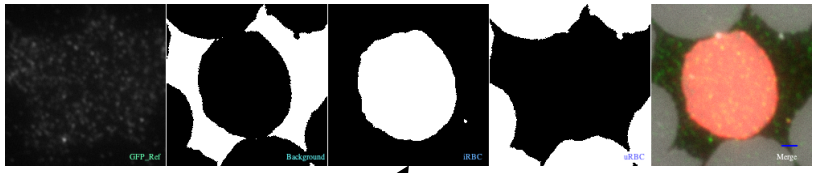
Measure masks, redirecting to reference image '_GFP'

Merge C1,C2,C3



Make montage with masks; Print scale bar; print labels; Flatten; Save into output directory

Proof



Sheared_membrane_masks.ijm proofs