

Find foci imagej script

Set parameters

```
// Parameter
    diameterROI = 6;
//below fmaxnoise is noise
    FMaxNoise = 6000;

    gDir = getDirectory("Choose a Directory") ;
    Fdir = gDir;

    diameterCell = 60; //
    DotsMax = 10000; //

    ori_ID = getTitle();
```

Main macro contains all of the functions

```
// main macro contains all processes
macro main {

// run("Options...", "iterations=1 count=1 black");
    setOption("black background", true); // set black background

    ori_ID = getTitle();

// Mask from Slice1-8 (Average)
    Slice1_8_Ave();

// Mask Check

// ROI manager > reset
    roiManager("reset");
// Mask > analyze particles > Result ON
    run("Select All");
    run("Analyze Particles...", "size=600-Infinity circularity=0.70-1.00 display
exclude clear include add");

IJ.log("Check1");

//waitForUser("Wait for Check1");

// Find maxima on ROI manager
    selectWindow(ori_ID);
    run("Select All");
```

```
FindMaxima_on_ROI();

} // End of main function
```

ROI check

```
// Cell Mask > analyze particles > XYCenter + diameterROI circle > ori_ID
function Cell_ROI_Check(){

    ROI_Mask_ID = getTitle();

    run("Watershed");

// Mask > analyze particles > Result ON
    run("Select All");
    run("Analyze Particles...", "size=600-Infinity circularity=0.70-1.00 display
exclude clear include add");
    IJ.log("Analyze Particles");

// FirstROI_x, y, nSli
    FirstROI_x = newArray();
    FirstROI_y = newArray();
    FirstROI_nSli = newArray();

// FirstROI_x, y, nSli loop
    for (i=0; i<nResults; i++) {

        FirstROI_x = Array.concat(FirstROI_x, getResult('XM', i));
        FirstROI_y = Array.concat(FirstROI_y, getResult('YM', i));
        FirstROI_nSli = Array.concat(FirstROI_nSli, getResult('Slice', i));
    }

    Array.show("CheckROI", FirstROI_x, FirstROI_y, FirstROI_nSli);

    IJ.renameResults("CheckROI_" + getTitle());

// ROI center (XM, YM) > diameterCell(= 60) Resize
    selectWindow(ori_ID);
    ROItoCellROI(FirstROI_x, FirstROI_y, FirstROI_nSli);

    selectWindow(ROI_Mask_ID);

} // End of function
```

Remove ROI

Filtering of ROI depending on position, size and circularity

```
function RemoveROI(){

    run("Select All");
    run("Duplicate...", " ");
    run("Analyze Particles...", "size=600-Infinity circularity=0.70-1.00 display
exclude clear include add");
    run("Select All");
    setForegroundColor(0, 0, 0);
    run("Fill", "slice");
    roiManager("Show None");
    roiManager("Show All");
    setForegroundColor(255, 255, 255);
    roiManager("Fill");

} // End of function
```

Find maxima on ROI

Find the foci

```
function FindMaxima_on_ROI(){

    run("Select None");

    OriImageID = getImageID();
    ori_ID = getTitle();
    IJ.log("ori_ID: "+ ori_ID);

    // ResultsのX, Y, Sliceの数値を配列に書きこみ。X, Yが中心。半径はdiameterROI/2。
    // FirstROI_x, y, nSli
    FirstROI_x= newArray();
    FirstROI_y= newArray();
    FirstROI_nSli = newArray();

    // FirstROI_x, y, nSli loop
    for (i=0; i<nResults; i++) {    //for (i=0; i<10; i++) {

        FirstROI_x = Array.concat(FirstROI_x, getResult('XM', i));
        FirstROI_y = Array.concat(FirstROI_y, getResult('YM', i));
        FirstROI_nSli = Array.concat(FirstROI_nSli, getResult('Slice', i));
    } // endo for

    Array.show("FirstROI", FirstROI_x, FirstROI_y, FirstROI_nSli);

    IJ.renameResults("FirstXY" + getTitle());

    // ROI center (XM, YM) > diameterCell(= 60) Resize
    ROItoCellROI(FirstROI_x, FirstROI_y, FirstROI_nSli);
```

```

// Gaussian Kernel image
Convolv_ID = Gauss_Convolve();
print(Convolv_ID);
// Save convolved image
saveAs("tif", Fdir +File.separator+ getTitle());
Convolv_ID = getTitle(); // Save tifで TIF > tif でエラーが出るためファイル名再取得

ROI_Num = 0;
roiManager("Select", ROI_Num);

// ROI manager の数だけFindMax実行
for(ROI_Num = 0; ROI_Num < roiManager("count") ; ROI_Num ++){

    // Noiseを引数にしてFind Max。結果はResultsに表示。
    FindMax(FMaxNoise, ROI_Num);
}

IJ.log("FindMax nResults:"+ nResults);

// ROI manager > clear
roiManager("Deselect");
roiManager("Delete");
// FirstROI > Add at 6th Slice < Cell Circle
for (k = 0; k < FirstROI_x.length; k++){
    setSlice(6);
    makeOval(FirstROI_x[k]-diameterCell/2, FirstROI_y[k]-diameterCell/2,
diameterCell, diameterCell);
    roiManager("Add");
}

//roiManager("Measure");

// FindMax() のResultsのX, Y, Sliceの数値を配列に書きこみ。X, Yが中心。半径は
diameterROI/2。

x= newArray();
y= newArray();
nSli = newArray();

for (i=0; i<nResults; i++) {    //for (i=0; i<10; i++) {

    x = Array.concat(x, getResult('XM', i));
    y = Array.concat(y, getResult('YM', i));
    // y = getResult('Y', i); // IJ.log(x +", "+ y);
    nSli = Array.concat(nSli, getResult('Slice', i)); // getResult('Slice',
i);
    // IJ.log("nSli: "+nSli);
} // endo for

// x, y, nSliの表示
Array.show(x, y, nSli);

// (X, Y, nSli, diameterROI) > ROI maneger > measure
// Convolv_ID

```

```

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

        selectImage(Convolve_ID);

        setSlice(nSli[j]);
        makeOval(x[j]-diameterROI/2, y[j]-diameterROI/2, diameterROI,
diameterROI);

        getStatistics(GSarea, GSmean, GSmin, GSmax, GSstd, GShistogram);

// Max > 8000 輝度の最大値で足り
        if(GSmax > DotsMax){
            print("over", GSmax);
            roiManager("Add");

        } else {

            //print("under", GSmax);

        }
        //roiManager("Measure");

    } // end for


selectWindow("Results");
run("Close");


// ROI manager > Stack (Convolve) > montage
selectImage(Convolve_ID);

ori_IM = getTitle();

ROItoStack(120);

columns_Nm = 5;

run("Enhance Contrast", "saturated=0.35");
row = nSlices/5 +1;
run("Make Montage...", "columns=" + columns_Nm + " rows=" + row + " scale=1
font=20 label use");
rename("Montage_" + ori_IM);
saveAs("tif", Fdir +File.separator+ getTitle());
selectWindow (ori_IM);


// ori_ID >ROI manager > All ROI > Measure

selectImage(ori_ID);

roiManager("Deselect");
roiManager("Show All");
roiManager("Measure");

```

```

    run("Select None");

// Results rename
    selectWindow("Results");

    ResName = ori_ID + "-roi-" + diameterROI + "-FMax-" + FMaxNoise + "_" + DotsMax;

    IJ.renameResults(ResName);

// Track mate analysis wait
//  waitForUser("Save Results? \n" + ResName);

// Save Results
    saveAs("Results", Fdir +File.separator+ ResName + ".csv");
// Save ROI.zip
    roiManager("Save", Fdir +File.separator+ ResName + ".zip");

} // end FindMaxima_on_ROI()

```

Gaussian kernel function

```

function Gauss_Convolve() {

    run("Select All");

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

    // GaussFit
    run("Convolve...", "text1=-0.09 -0.08  -0.07  -0.08  -0.09\n-0.08  0.05
0.28  0.05  -0.08\n-0.07  0.28  0.91  0.28  -0.07\n-0.08  0.05
0.28  0.05  -0.08\n-0.09  -0.08  -0.07  -0.08  -0.09\n stack");

    rename("Convolv_" + getTitle());

    return getTitle();

} // End of function

```

ROItoCellROI

Creates a circle ROI on the cell for counting the foci with the specified diameter: `diameterCell`

```

function ROItoCellROI(X_array, Y_array, Sli_array){

// ROI manager > reset
    roiManager("reset");

// FirstROI > Add at 6th Slice < Cell Circle

```

```

    for (k = 0; k < X_array.length; k++){
        // setSlice(6);
        makeOval(X_array[k]-diameterCell/2, Y_array[k]-diameterCell/2,
diameterCell, diameterCell);
        roiManager("Add");
    }

    roiManager("Deselect");
    roiManager("Remove Slice Info");

} // End of function

```

Find maxima

```

function FindMax(Noise, ROI_ID){
    // Zscan_Cell_Mask_Center();

    setSlice(1);

    // ROI manager (ROI_ID)を計測
    roiManager("Select", ROI_ID);
    // run("Measure");

    // ROI manager (ROI_ID)でFind Maxima > Point > measure
    for(i = 0; i<nSlices; i++){

        roiManager("Select", ROI_ID);

        run("Find Maxima...", "noise="+Noise+" output=[Point Selection] exclude");

        // Measure if Point selection == (10) --> point
        if(selectionType() == 10){
            run("Measure");
        }

        run("Next Slice [>]");

    } // End for find maxima

} // End function

```

ROI to stack

Export ROI as stack Images

```

// ROI Manager > select > Stack (120 pixs)
//
function ROItoStack(Image_size){

```

```

    Square = Image_size; // ROI size

    ROI_Num = roiManager("count");

    ori_ID = getTitle();
    run("Select All");

// Stack image
    newImage("particle_" + ori_ID, "16-bit black", Square, Square,
    roiManager("count"));
    partID = getImageID();

    setBatchMode(true);

    nSli_o = 0;

// Loop Results > ROIs > paste
    for (i=0; i < ROI_Num; i++) { //for (i=0; i<10; i++) {

        selectImage(ori_ID);
        roiManager("Select", i);
        run("Enlarge...", "enlarge=60");
        run("Copy");

        selectImage(partID);
        setSlice(i+1);
        run("Paste");

// Dot at Slice Change

        run("Select All");
    }

    rename(getTitle + "_" + Image_size + "_" + ".tif");
    run("Enhance Contrast", "saturated=0.35");

    saveAs("tif", Fdir +File.separator+ getTitle());

    setBatchMode(false);

} // ROItoStack()
////////////////////

```