Code for

Photoswitchable Single-Stranded DNA-Peptide Coacervate Formation as a Dynamic System for Reaction Control

Wen Ann Wee, Hiroshi Sugiyama*, and Soyoung Park*

*Corresponding author. Email: oleesy@kuchem.kyoto-u.ac.jp (S.P.), hs@kuchem.kyoto-u.ac.jp (H.S.)

ImageJ macro for droplet counting

Images obtained from microscope observation were exported to TIF files and the following ImageJ macro (adapted from Grishagin, 2015) was used to obtain the droplet counts for all samples. Typical output images are as shown in **Figure S1D**. Output images were visually inspected to verify the validity of droplet counts.

```
macro "DropletCount--BrightField--Threshold--QD--BGSubtract" {

    //show prompt for selection of source directory
    dir = getDirectory("Choose source directory");

    //get the file list
    list = getFileList(dir);
    resultsFileName = "Droplet Counting Results (Threshold).txt"

    //make directory to store processed images
    File.makeDirectory(dir+"Processed Images");

    //hide all the details from user
    setBatchMode(true);

    //process every file...
    for (i=0; i<list.length; i++) {

              //...that has a .tif extension
              if (endsWith(toUpperCase(list[i]), ".TIF")) {

                        open(dir+list[i]);
                        fileNoExt = split(list[i], ".");

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

             //Flatten image
              run("8-bit");
              setAutoThreshold("Default dark");

                 //run("Threshold...")
              setThreshold(50, 255);
              run("Convert to Mask");

saveAs("Tiff", dir+"Processed Images"+File.separator+fileNoExt[0]+" Threshold");

                        //analyze and show as [Bare Outline]
run("Analyze Particles...", "size=0-100 pixel show=[Bare Outlines] clear");
saveAs("Tiff", dir+"Processed Images"+File.separator+fileNoExt[0]+" Outline");

                        //create overlay with original image and outline
    call("ij.plugin.frame.ThresholdAdjuster.setMode", "Red");
                        setAutoThreshold("Default");

                        open(dir+list[i]);
                        selectWindow(fileNoExt[0]+" Outline.tif");
run("Add Image...", "image="+"["+list[i]+"]"+" x=0 y=0 opacity=75");
```

```
                    run("Flatten");
saveAs("Tiff", dir+"Processed Images"+File.separator+fileNoExt[0]+" Outline Overlay");

                    close();
                    close();
                    close();

                    //analyze and show as [Overlay]
run("Analyze Particles...", " size=0-100 pixel show=Overlay clear summarize");
saveAs("Tiff", dir+"Processed Images"+File.separator+fileNoExt[0]+" Overlay");


                    //close all windows
                    selectWindow(fileNoExt[0]+" Overlay"+".tif");
                    close();
            }
    }

    //copy all relevant contents from the Summary window
    selectWindow("Summary");
    text = getInfo("window.contents");
    lines = split(text, "\n");

//create a text file with counting results, output cell count, total area and average size
    //copy the results into clipboard
    //reset string buffer
    String.resetBuffer;

    for (i=0; i<lines.length; i++) {

            if (i==0){
                    File.saveString("",dir+resultsFileName);
                    f = File.open(dir+resultsFileName);
            }

            labels = split(lines[i], "\t");
        labelNoExt = split(labels[0], ".");
        resultsList = newArray(labelNoExt[0]+"\t"+labels[1]+"\t"+labels[2]+"\t"+labels[3]);
            print(f,resultsList[0]);

//append another value to the string
            String.append(resultsList[0]+"\n");
    }

    File.close(f);

    //open the summary file, close the Summary window
    open(dir+resultsFileName);
    selectWindow("Summary");
    run("Close");

    //copy all values into the clipboard
    String.copy(String.buffer);

setBatchMode(false);

}
```