

https://doi.org/10.1071/RD20143_AC

Supplementary Material

Time-lapse confocal imaging-induced calcium ion discharge from the cumulus–oocyte complex at the time of cattle oocyte activation

Hanna J. McLennan^{A,B,C,E}, Melanie L. Sutton-McDowall^{A,B,C}, Sabrina Heng^{A,C,D}, Andrew D. Abell^{A,C,D} and Jeremy G. Thompson^{A,B,C}

^AAustralian Research Council Centre of Excellence for Nanoscale BioPhotonics, Australia.

^BRobinson Research Institute, Adelaide Medical School, University of Adelaide, SA 5005, Australia.

^CInstitute for Photonics and Advanced Sensing, School of Physical Sciences, University of Adelaide, SA 5005, Australia.

^DDepartment of Chemistry, School of Physical Sciences, University of Adelaide, SA 5005, Australia.

^ECorresponding author. Email: hanna.mclennan@adelaide.edu.au

File S1. Supplementary Videos, and Supplementary Information

Supplementary Videos

The supplementary videos that accompany this publication can be accessed via the following link:

<https://www.dropbox.com/sh/bm9r5trrr5o78j1/AACxCGViAVRpRHi9GbHMkf3Ra?dl=0>

Supplementary Information

Fiji macros were specifically developed for the image analysis conducted within this study. The macros used for each step of analysis are presented below.

Step 1: Formatting raw Olympus image files (.oif)

These two macros open raw '.oif' files with two and three channels respectively, split the channels, convert them to 8-bit greyscale and save the separate channels as individual stacks of images for analysis.

Formatting macro (2 channels)

```
//This macro will format .oif files taken on an
//Olympus Confocal Microscope for later image
//analysis. It will also separate the channel(s)
//from the DIC (or phase contrast) image. The
//final stack(s) will be saved as an 8-bit greyscale
//.tiff file in a directory chosen by the user.
```

```
//Global variables
var dir1="Temp";
var dir2="Temp";
var list = 0;
var nameStore="Temp";

fileFunction();
```

```
//loop the formatting code for all .oif stacks in the directory
for(i=0; i<list.length; i++) {
    filename = dir1 + list[i];
    if (endsWith(filename, ".oif")) {
        openFunction();

        run("8-bit");
        channelFunction();
        saveFunction();

        run("Close All");
        run("Collect Garbage");
    }
}

//END OF MACRO

//Functions
function fileFunction() {
    //Ask the user to select the source directory
    dir1 = getDirectory("Select the source directory");

    //Ask the user to select the destination directory
    dir2 = getDirectory("Select the destination directory");

    //Generate a list of the files in the selected directory
    list = getFileList(dir1);
    Array.sort(list);
}

function openFunction() {
```

```
//Open the .oif files in greyscale and as Hyperstacks
run("Bio-Formats Importer", "open=[filename]
autoscale color_mode=Grayscale view=Hyperstack
stack_order=XYCZT");
}

function channelFunction() {
    //Store the initial image name into the nameStore
variable
    nameStore = getTitle();

    //Split the channels into stain(s) and DIC
run("Split Channels");
}

function saveFunction() {
    //Select the first channel stack
selectWindow("C1-"+nameStore);

    //Save the stack as a .tiff image stack for later
analysis
saveAs("Tiff", dir2+nameStore+" - Ca(Fluo4AM).tif");

    //Select the second channel stack
selectWindow("C2-"+nameStore);

    //Save the stack as a .tiff image stack for later
analysis
saveAs("Tiff", dir2+nameStore+" - PC.tif");
}
```

Formatting macro (3 channels)

```
//This macro will format .oif files taken on an
//Olympus Confocal Microscope for later image
//analysis. It will also separate the channel(s)
//from the DIC (or phase contrast) image. The
//final stack(s) will be saved as an 8-bit greyscale
//.tiff file in a directory chosen by the user and the
//the DIC image will be saved in a new folder generated
//by this macro called PC.

//Global variables
var dir1="Temp";
var dir2="Temp";
var saveDir="Temp";
var fs="Temp";
var list = 0;
var nameStore="Temp";

fileFunction();

//loop the formatting code for all .oif stacks in the directory
for(i=0; i<list.length; i++) {
    filename = dir1 + list[i];
    if (endsWith(filename, ".oif")) {
        openFunction();

        run("8-bit");
        channelFunction();
        saveFunction();
    }
}
```

```
        run("Close All");
        run("Collect Garbage");
    }
}

//END OF MACRO

//Functions
function fileFunction() {
    //Ask the user to select the source directory
    dir1 = getDirectory("Select the source directory");

    //Ask the user to select the destination directory
    dir2 = getDirectory("Select the destination directory");

    //Define and create another output directory for PC
images
    saveDir = dir2+"PC";
    File.makeDirectory(saveDir);
    fs = File.separator;

    //Generate a list of the files in the selected directory
    list = getFileList(dir1);
    Array.sort(list);
}

function openFunction() {
    //Open the .oif files in greyscale and as Hyperstacks
    run("Bio-Formats Importer", "open=[filename] autoscale
color_mode=Grayscale view=Hyperstack stack_order=XYZCT");
}
```

```
function channelFunction() {  
    //Store the initial image name into the nameStore  
variable  
    nameStore = getTitle();  
  
    //Split the channels into stain(s) and DIC  
    run("Split Channels");  
}  
  
function saveFunction() {  
    //Select the first channel stack  
    selectWindow("C1-"+nameStore);  
  
    //Save the stack as a .tiff image stack for later  
analysis  
    saveAs("Tiff", dir2+nameStore+" - Ca(Fluo4AM).tif");  
  
    //Select the second channel stack  
    selectWindow("C2-"+nameStore);  
  
    //Save the stack as a .tiff image stack for later  
analysis  
    saveAs("Tiff", dir2+nameStore+" - DeadRed(Red).tif");  
  
    //Select the third channel stack  
    selectWindow("C3-"+nameStore);  
  
    //Save the stack as a .tiff image stack for later  
analysis  
    saveAs("Tiff", saveDir+fs+nameStore+" - PC.tif");  
}
```

Step 2: Surrounding media reference area check

This macro draws 100x100 pixel squares in each corner of the image. Each area was then manually checked for any cellular fluorescence, leading to a false positive fluorescence reading, and whether the edge of a media droplet was present, leading to a false negative fluorescence reading, before it was used as a measure of surrounding media fluorescence.

```
//This macro draws the four control squares on a stack
//so the user can then check through the stack to make
//sure the COC never enters it or that the square is
//outside the media bubble

//Get the image size
w = getWidth();
h = getHeight();
w1 = w -100;
h1 = h -100;

//Draw the control squares as store them as a single ROI
run("ROI Manager...");

//Top Left Square
makeRectangle(0, 0, 100, 100);
setKeyDown("shift");

//Top Right Square
makeRectangle(w1, 0, w, 100);
setKeyDown("shift");

//Bottom Left Square
makeRectangle(0, h1, 100, h);
```



```
setKeyDown("shift");

//Bottom Right Square
makeRectangle(w1, h1, w, h);
roiManager("Add");

//Now play through the stack and make sure the oocyte
//never enters any of the squares and do not use any
//squares it enters as controls

//Check the phase contrast images to make sure the
//edge of the drop is not in any of the squares and
//do not use any squares where this is the case as controls

//The order of the squares in the results table is as follows:
//      1          2
//
//
//      3          4
//END OF MACRO
```

Step 3: Measuring surrounding media fluorescence

This macro measures the mean fluorescence in the corners of each frame of an image stack. Any areas identified in the previous step were excluded from the overall average for each cumulus-oocyte-complex.

```
//This macro will measure the mean fluorescence
//of the surrounding media in each corner of an
//image stack over time.
//
//The order of the squares in the results table is as follows:
```

```
//      1          2
//
//
//      3          4

//Global variables
var dir="Temp";
var saveDir="Temp";
var fs="Temp";
var list = 0;
var w="Temp";
var h="Temp";
var w1="Temp";
var h1="Temp";
var aROI="Temp";
var bROI="Temp";
var nameStore="Temp";

fileFunction();

//loop the measuring code for all formatted .tiff stacks
//in the directory
for(i=0; i<list.length; i++) {
    filename = dir + list[i];
    //Clear out ROIs from previous image (ignored if first
image)
    if (i>0) {
        countAROI=roiManager("Count");
        countBROI=countAROI-1;
        for (n=countBROI; n>-1; n--) {
            roiManager("Select", n);
```

```
        roiManager("Delete");
    }
}
if (endsWith(filename, ".tif")) {
    open(filename);

    controlFunction();

    run("Close All");
    close("Results");
    run("Collect Garbage");
}
}

//END OF MACRO

//Functions
function fileFunction() {
    //Ask the user to select the source directory
    dir = getDirectory("Select the source directory");

    //Define the output directory and create it
    saveDir = dir+"Output";
    File.makeDirectory(saveDir);
    fs = File.separator;

    //Generate a list of the files in the selected directory
    list = getFileList(dir);
    Array.sort(list);
}
```

```
function controlFunction() {  
    //Get the image size  
    w = getWidth();  
    h = getHeight();  
    w1 = w -100;  
    h1 = h -100;  
  
    //Get the stack name  
    nameStore = getTitle();  
  
    //Draw the control squares as store them as ROIs  
    run("ROI Manager...");  
  
    //Top Left Square  
    makeRectangle(0, 0, 100, 100);  
    roiManager("Add");  
  
    //Top Right Square  
    makeRectangle(w1, 0, w, 100);  
    roiManager("Add");  
  
    //Bottom Left Square  
    makeRectangle(0, h1, 100, h);  
    roiManager("Add");  
  
    //Bottom Right Square  
    makeRectangle(w1, h1, w, h);  
    roiManager("Add");  
  
    //Set the measurements
```

```
run("Set Measurements...", "area mean standard min
display redirect=None decimal=3");

//Measure the four control boxes
selectWindow(nameStore);
roiManager("Multi Measure");

//Save the Results as "Control"
selectWindow("Results");

//Save the results table as an excel file
saveAs("Results", saveDir+fs+nameStore+" - Control.xls");

//Close the Results
close("Results");

//Clear the ROIs
aROI=roiManager("Count");
bROI=aROI-1;
for (n=bROI; n>-1; n--) {
    roiManager("Select", n);
    roiManager("Delete");
}

//Reset the selection by drawing a pointless line
selectWindow(nameStore);
makeLine(355, 172, 355, 172);

//Close the ROI Manager
close("ROI Manager");
}
```

Step 4: Defining the cumulus and oocyte regions of interest (ROI)

This macro prompts the user to select the oocyte with a circle ROI and draw around the cumulus vestment. It then stores the oocyte selection in the ROI manager and creates a donut shaped ROI by subtracting the oocyte area from the whole COC area. These ROIs are then used in the next step to measure oocyte and cumulus vestment fluorescence individually.

```
//This macro will perform image stack alignment
// (unless turned off with '//') and then prompt
// the user to select around the cumulus and then
// oocyte and subtract the oocyte from the cumulus
// ROI to create a donut shape.
```

```
//NOTE: this macro works individually and has no
// batch mechanics.
```

```
//Global Variables
var ROICount="Temp";
var ROINumber="Temp";
```

```
//Step 1: Have a converted image stack open (the
// channel you wish to measure) and clear out any
// previous ROIs
```

```
ROICount=roiManager("Count");
if (ROICount>0) {
    ROINumber=ROICount-=1;
    for (n=ROINumber; n>-1; n--) {
        roiManager("Select", n);
        roiManager("Delete");
    }
}
```

```
//Step 2: Align the Image Stack (for moving/spinning COCs)
alignFunction();

//Step 3: Open the ROI manager
run("ROI Manager...");

//Step 4: Prompt user to select around the cumulus cells
waitForUser("Draw around the most expanded cumulus");
roiManager("Add");

//Step 5: Prompt user to select around the oocyte
waitForUser("Draw a circle around the oocyte only");
roiManager("Add");

//Step 6: Make the cumulus ROI ring shaped (exclude the oocyte)
roiManager("Select", 0);
setKeyDown("alt");
roiManager("Select", 1);
setKeyDown("none");
roiManager("Add");
roiManager("Select", 0);
roiManager("Delete");

//Step 7: Rename the ROIs
roiManager("Select", 0);
roiManager("Rename", "Oocyte");
roiManager("Select", 1);
roiManager("Rename", "Cumulus");

//Step 8: Clean Up
```

```
run("Collect Garbage");

//END OF MACRO

//Functions
function alignFunction() {
    //Align the stack to counteract COC movement
    run("StackReg ", "transformation=[Rigid Body]");

    //Clear the running memory
    run("Collect Garbage");
}
```

Step 5: Measuring cumulus and oocyte fluorescence

This macro takes the ROIs defined in the previous step and measures the mean fluorescence intensity in the cumulus and the oocyte. It is important to note that the same ROIs for a single COC were used across different fluorescent channels to keep the area measured consistent.

```
//This macro will select the ROIs of the cumulus and
// oocyte (previously defined in the Step 4 Macro)
// and perform and save both basic measurements and
// GLCM analysis.

//NOTE: this macro works individually and has no
// batch mechanics.

//Global Variables
var dir="Temp";
var fs="Temp";
var nameStore="Temp";
```

```
//Step 1: Set the directory the data outputs will be saved in
fileFunction();

//Step 2: Get the stack image name and store it in a variable
nameStore = getTitle();

//Step 3: Set the measurements that will be collected
run("Set Measurements...", "area mean standard min display
redirect=None decimal=3");

//Step 4: Perform the basic measurements of the cumulus and oocyte
regions
roiManager("Select", newArray(0,1));
roiManager("Multi Measure");

//Step 5: Save the basic measurements output
saveOutputB();

//Step 6: Clean Up
run("Collect Garbage");

//END OF MACRO

//Functions
function fileFunction() {
    //Ask the user to select the directory the data will be
    saved in
    dir = getDirectory("Select the data save directory");
    fs = File.separator;
}

function saveOutputB() {
```

```
//Select the results table
selectWindow("Results");

//Save the results table as an excel file
saveAs("Results", dir+fs+nameStore+" - Basic
Results.xls");
}
```