Jun 12, 2020

Wound healing migration/invasion assay in 96-well format

Nature Communications

DOI
dx.doi.org/10.17504/protocols.io.bgk4juyw

Douglas Adamoski[1], Sandra Martha Gomes Dias[1]

[1]Brazilian Biosciences National Laboratory (LNBio), Brazilian Center for Research in Energy and Materials (CNPEM), Zip Code 13083-970, Campinas, Sao Paulo, Brazil.

Douglas Adamoski

OPEN ACCESS

DOI: https://dx.doi.org/10.17504/protocols.io.bgk4juyw

External link: https://doi.org/10.1038/s41467-024-49874-x

Protocol Citation: Douglas Adamoski, Sandra Martha Gomes Dias 2020. Wound healing migration/invasion assay in 96-well format. **protocols.io https://dx.doi.org/10.17504/protocols.io.bgk4juyw**

Manuscript citation:
Adamoski, D., M. dos Reis, L., Mafra, A.C.P. *et al.* HuR controls glutaminase RNA metabolism. *Nat Commun* **15**, 5620 (2024). https://doi.org/10.1038/s41467-024-49874-x

Protocol status: Working

Created: May 19, 2020

Last Modified: June 12, 2020

Protocol Integer ID: 37244

Keywords: wound healing migration, assay, invasion, well format

Protocol materials

Collagen Type I solution from rat tail **Merck MilliporeSigma (Sigma-Aldrich) Catalog #**C3867

Acetone

2-Propanol **Merck MilliporeSigma (Sigma-Aldrich) Catalog #**190764

Chloroform

Troubleshooting

Collagen from rat tail tendon purification

2w

1 This first part of the protocol is aimed to prepare the Collagen from rat tail tendon. If your lab already has collagen suitable for 3D matrix polymerization, you may skip this step.

Collagen Type I solution from rat tail **Merck MilliporeSigma (Sigma-Aldrich) Catalog #**C3867

Note

Please cite: Rajan, N., Habermehl, J., Coté, M.-F., Doillon, C. J. & Mantovani, D. Preparation of ready-to-use, storable and reconstituted type I collagen from rat tail tendon for tissue engineering applications. *Nat. Protoc.***1,** 2753–2758 (2006).

2 Remove 10-15 rat tails from -80 °C and left at 4 °C for thawing. 24:00:00

3 Move tails from 4 °C to Room temperature . 00:30:00

4 Wash tails in ultrapure water (in a large beaker) and transfer to a new beaker with ultrapure water.

5 Remove one tail and dry it with a paper towel.

6 Hold tail 5 mm before its thinner end using tweezers. Hold it and, using hands, twist the tail around the tweezer until skin break. Pull the tweezer with the collagen fibers.

7 Cut the fibers and immerse them in PBS 1X (room temperature). Repeat several times until the tail completely processed (or impossible to handle). Transfer to new PBS for 00:05:00 .

8 After collecting all fibers, transfer them to a new beaker with 500 mL of PBS and wait for 00:05:00

9 Transfer again to 500 mL of pure acetone (inside chemical hood) and left stay for 00:00:00 . The fibers will lose their flexibility.

Acetone

10 Transfer to 500 mL [M] 70 % volume Isopropanol (350 mL of isopropanol and 150mL of ultrapure water) and left stay for 00:05:00

2-Propanol **Merck MilliporeSigma (Sigma-Aldrich) Catalog #**190764

11 Collect fibers in a large beaker with a magnetic bar and add 500 mL of acetic acid [M] 20 millimolar (mM) (574 µL glacial acid acetic in 500 mL of ultrapure water).

12 Stir for at least 48:00:00 at 4 °C . If the solution becomes too dense during this period, add the needed amount of [M] 20 millimolar (mM) acetic acid solution.

13 During the incubation, prepare small ice pieces with ultrapure water and store a household blender and centrifuge flasks at 4 °C

14 The solution should be viscous. Transfer the ultrapure ice to cup together with the collagen solution to the blender and pulse it multiple times (to avoid overheating). Just add a small amount of ice, enough to blend and keep the solution cold during the process. The solution will acquire a pale white color. Perform the steps at 4 °C

15 Transfer to centrifuge flasks and balance them. Immediately wash the blender to avoid protein binding to the plastic.

16 Centrifuge for 10000 x g, 4°C, 01:00:00

17 Transfer (without disturbing the pellet) to plastic or metallic containers to form ice. The final ice layer should be thinner than 5 mm.

18 Store at -20 °C until completely frozen 48:00:00 .

19 Remove the frozen collagen, crush it to smaller pieces and transfer to the freeze-dryer flask. Perform the lyophilization 96:00:00 . Thaw the condenser coil during the process, if needed.

20 Prepare 5 L [M] 20 millimolar (mM) of acetic acid and store at 4°C, together with a simple household hand mixer.

21 After finishing the lyophilization, remove collagen sponge, transfer to ziplock bags, and store at -80 °C .

22 In order to dissolve it, weight the desired amount (suggestion: [M] 4 mg/mL) and transfer this amount of collagen sponge to a beaker flask. Add enough acetic acid [M] 20 millimolar (mM) to achieve the desired concentration.

23 Dissolve the collagen using the hand blender. Use the following video as reference.

https://www.youtube.com/embed/RPPpgGe_AMo

24 Assembly dialysis with [M] 20 millimolar (mM) acetic acid (1:10, 4 °C) for 02:00:00 .

25 Change the dialysis buffer and left it Overnight at 4 °C with the magnetic bar.

26 Transfer the solubilized and dialyzed collagen to a sterile borosilicate glass flask with lid. Carefully add chloroform (10% of the total volume of collagen) at the bottom of the flask. Avoid mixing the chloroform with the collagen. Incubate for 24:00:00 at 4 °C Chloroform

27 Carefully remove the chloroform inside biological safety cabinet using aseptic technique and sterile tools. Left it open inside the biological safety cabinet on On ice , for

01:00:00 in order to remove chlorine gas.

28 Test for sterility and store at 4 °C until further use

Plate coating with Collagen I

29 Using pre-cooled tips (4 °C) and in laminar flow hood, prepare Collagen COATING solution (see table below, solution enough for one 96-well plate). First, add the Acetic Acid [M] 20 Mass Percent , then – while the stock solution is On ice – collect Collagen I with a positive-displacement pipette.

	Collagen COATING solution						
	Reagent	**Stock**		**Use**		**Volume**	
	Collagen I, from rat tail	4	mg/mL	0,3	mg/mL	450,0	µL
	Acid Acetic	20	mmol/L			5,6	mL
					Total	6	mL

30 Briefly vortex and spin the conical tube. Dispense 50 µL of Collagen COATING solution on each well from 96-well plate. Gently rock the plate to even coating of each well.

31 Place the plate in a 37 °C incubator, [M] 5 % volume CO2 , 01:00:00

32 Remove theplate from the incubator and aspirate the solution.

33 Wash twice with phosphate-buffered saline.

34 Dry out the plate in the hood, seal, and store at 4 °C 4 or immediately use.

Wound healing assay

35 Tripsinize cell lines intended to use in the assay. Count using the appropriate device (eg. Neubauer chamber). Prepare a final solution with an adequate number of cells to coated plate in 100 µL . Check the suggested amount in the table below.

	MDA-MB-231	40.000 cells
	BT549	30.000 cells
	Hs578t	30.000 cells

Suggested amount of common breast cancer cells lines

36 Allow the cell plate to sit at room temperature for 00:05:00 to allow cells to evenly disperse across the bottom of the plate. This may require more or less time for each cell type.

37 Place the plate in a 37 °C incubator, 5% CO_2, 12:00:00 - 24:00:00

38 Remove medium from cells and wash them with serum-free medium.

39 Add 150 µL of Serum-free medium supplemented with 0,1% HI-BSA

	Reagent	Use	Volume
	Bovide Serum Albumin	25 mg/mL	2.5 g
	PBS (1X)		100 mL
	Incubate at 56°C for 30 minutes (after 56°C temperature reached). Filter-sterilize with 0.22µm. Store at 4°C.		

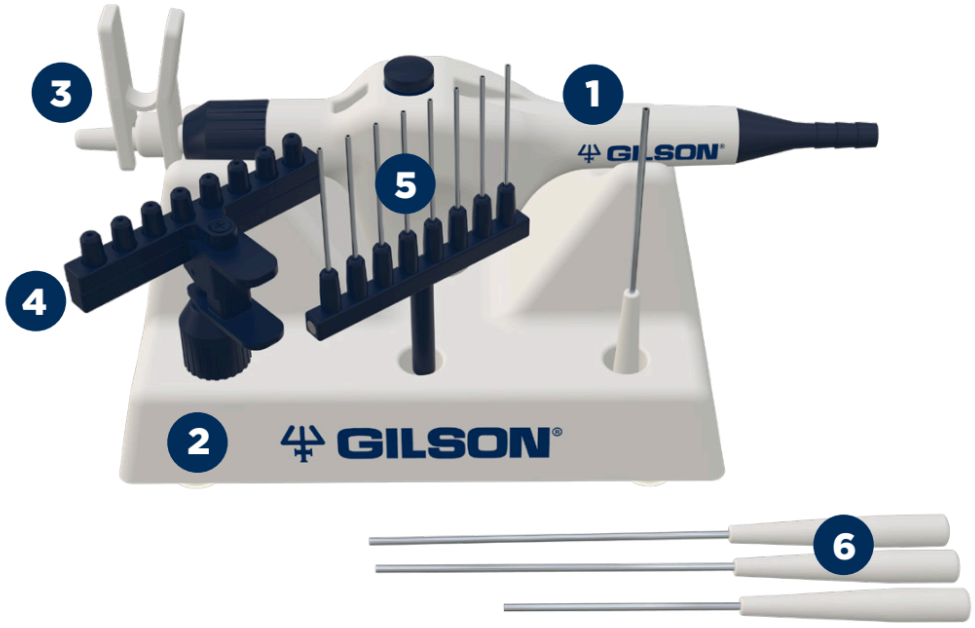
Heat-Inactivated Bovine Serum Albumin 2.5% (HI BSA) stock solution

40 Place the plate in a 37 °C incubator, 5% CO_2, 12:00:00 - 24:00:00

41 Remove the plate from the incubator and perform scratch in all wells.

Suggestion: Accessory 4 or 5 (both with attached 200 µL pipette tips) from Gilson's Aspiration Station Kit could be used to perform even and reproducible scratches.

Check on the microscope if you were successful in the process.



42 Remove the medium with aid of a multichannel pipette. Look to check if all wells have an integral scratch

43 Wash the plate once with serum-free medium.

STEP CASE

Collagen-overlaid invasion assay 14 steps

44 Left the wells with 100 µL of serum-free medium.

45 Put the plate over ice-covered with aluminum foil for 5 minutes.

46 Remove medium from wells.

47 Overlay each well with 50 µL of Collagen INVADING solution. Gently rock the plate to even coating of each well.

		Stock	Use	Volume
	Collagen I, from rat tail	4 mg/mL	2 mg/mL	3500 µL
	RPMI 1640 (5X concentrated)	5 X	1 X	1400 µL
	FBS	100 %	20 %	1400 µL
	Ultrapure sterile water			630 µL
	NaOH	2000 mM	20 mM	70 µL
			Final volume	7000 µL

*Note: NaOH should be added until pH reaches 7.4. This could be checked using pH test strips.

Collagen INVADING solution

Note

RPMI 1640 (5X concentrated) could be replaced with 25 mM HEPES-Buffered media if your microscope stage does not have CO2 injection

48 Return cells to the incubator for 00:30:00

49 Overlay the 3D matrices with 100 µL of medium with 10% FBS. The medium should be pre-warmed and pre-equilibrated in the incubator.

Note

Regular media could be replaced with 25 mM HEPES-Buffered media if your microscope stage does not have CO2 injection

50 Immediately image plate 0h. In Operetta system (Perkin-Elmer), use 10X Long WD objective, Brightfield mode, and set 15% overlap in “Well” options. Select 9 fields for each well in square-shape. Incubate with 5% CO2 at 37°C during the imaging.

Equipment

Operetta High-Content Analysis System — NAME

Perkin Elmer — BRAND

HH16000000 — SKU

51 Perform readings at every hour for 24:00:00 .

52 After transfer to Columbus, export images from Columbus (Using “Single Plane Tiff” option) to your computer.

53 Remove all accessory files and left each folder just with images.

54 Run ImageJ script “Scratch_and_Wound_1-Sorter_n_Tiler_v0.3_overlap_mult.ijm”

Command

Scratch_and_Wound_1-Sorter_n_Tiler_v0.3_overlap_mult.ijm (Fiji/ImageJ Macro)

Scratch_and_Wound_1-Sorter_n_Tiler_v0.3_overlap_mult.ijm

```
/*
Scratch and Wound analysis

two scripts:
Scratch_and_Wound_1-Sorter_n_Tiler_v0.3_overlap_mult.ijm
Scratch_and_Wound_v0.1.ijm

Douglas Adamoski
douglas.adamoski@gmail.com

Input:
Full folder with Columbus Export

ALERT 1:
Remove "ImageIndex.ColumbusIDX.csv" and "ImageIndex.ColumbusIDX.xml"
files from folder!

ALERT 2:
The experiment should be performed on Operetta equipement using 10X
magnification objetive
and 9 fields in square shape (3 lines x 3 columns)

ALERT 3:
Read ALERT 1! NO other files/folders than images should be in same
folder.

ALERT 4:
Remove []'s from path!

CHANGELOG
v0.3 - Second version, 11th july 2017
                Adapted for multiple timepoint input from TCO
controlled operetta
v0.2
v0.1 - First version, 29th may 2017
                Accept on-screen parameter changes
                Split in two scripts in order to easy the restart
```

```
                Split in two scripts in order to easy the restart
without losing the process
*/

// Create funcion ArrayUnique
// objective of this function is clean one vector in order to remove
duplicates
// gently provided by Richard Wheeler:
// http://www.richardwheeler.net/contentpages/textgallery.php?
gallery=ImageJ_Macros
function ArrayUnique(array) {
        array   = Array.sort(array);
        array   = Array.concat(array, 999999);
        uniqueA = newArray();
        i = 0;
        while (i<(array.length)-1) {
                if (array[i] == array[(i)+1]) {
                        //print("found: "+array[i]);

                } else {
                        uniqueA = Array.concat(uniqueA, array[i]);
                }
                i++;
        }
        return uniqueA;
}

// Asks for main dir
  dir = getDirectory("Choose a Directory ");
// Request variables from user
// changed from original example:
https://imagej.nih.gov/ij/macros/DialogDemo.txt
// Pixel Width scale (default operetta 10x: 0.64504)
Valor5=1.29;

// Define Tile overlap
Valor2=15;

// Create dialog window
  Dialog.create("Analysis parameters");
  Dialog.addMessage("Define the lateral pixel size in micrometers");
  Dialog.addNumber("Pixel Width scale (um):", Valor5);
  Dialog.addMessage("Define tile overlap in percentage");
  Dialog.addNumber("Tile overlap (%):", Valor2);
  Dialog.show();
  Valor5 = Dialog.getNumber();
  Valor2 = Dialog.getNumber();
```

```
// Prints ImageJ Version
```

55 Rename the “Tiles” folder to “TimepointBegin” (if the folder is relative to 0H) or “TimepointEnd” (if the folder is from any other timepoint).

56 Create an folder for each timepoint and put inside just “TimepointEnd” (the actual timepoin) and “TimepointBegin” (0H timepoint) folders. “TimepointBegin” will be the same for all “TimepointEnd”.

57 Run ImageJ script “Scratch_and_Wound_2-Measure_v0.2.1.ijm”.

Command

Scratch_and_Wound_2-Measure_v0.4.ijm (Fiji/ImageJ Script)

Scratch_and_Wound_2-Measure_v0.4.ijm

```
/*
Scratch and Wound analysis

two scripts:
Scratch_and_Wound_1-Sorter_n_Tiler_v0.3_overlap_mult.ijm
Scratch_and_Wound_2-Measure_v0.4

USes the old overlapper!!!

Douglas Adamoski
douglas.adamoski@gmail.com

Input:

Remove "log" from first script (left just folders)

CHANGE HERE:

ALERT 1:
Use the same folder names! Please!

ALERT 2:
Check with ALL files for EACH well are inside EACH folder in the
right timepoint.

ALERT 3:
Read and follow alerts 1 and 2.

CHANGELOG
v0.4 - August 15th 2017
                Ideas collected from:
                        ImageJ macros for the user-friendly analysis
of soft-agar and wound-healing assays
                                João Paulo Silva Nunes and Adriana
Abalen Martins Dias
                                BioTechniques, Vol. 62, No. 4, April
2017, pp. 175–179
```

```
v0.3 - Forth version, july 11th 2017
                Multiple timepoints input
                Animated GIF export
v0.2.2 - Third version, june 15th 2017
                Bug correction
v0.2.1 - Third version, june 15th 2017
                Bug correction
v0.2 - Second version, june 14th 2017
                Joins multiple ROIs
v0.1 - First version, may 29th 2017
                Accept on-screen parameter changes
                Split in two scripts in order to easy the restart
without losing the process

*/

// Create funcion ArrayUnique
// objective of this function is clean one vector in order to remove
duplicates
// gently provided by Richard Wheeler:
// http://www.richardwheeler.net/contentpages/textgallery.php?
gallery=ImageJ_Macros
function ArrayUnique(array) {
        array   = Array.sort(array);
        array   = Array.concat(array, 999999);
        uniqueA = newArray();
        i = 0;
        while (i<(array.length)-1) {
                if (array[i] == array[(i)+1]) {
                        //print("found: "+array[i]);

                } else {
                        uniqueA = Array.concat(uniqueA, array[i]);
                }
                i++;
        }
        return uniqueA;
}


// Asks for main dir
  dir = getDirectory("Choose a Directory ");
// Request variables from user
// changed from original example:
https://imagej.nih.gov/ij/macros/DialogDemo.txt
```

```
// Maximum scracth find
Valor1=90;
// Minimum cell threshold
Valor2=30;
// Minimum size for each cell in particle analysis (may count
projections)
Valor3=50;
// First timepoint folder
Valor4="001001001";
// GIF ms delay
Valor5=600;
// Numer of timepoints
Valor6=4;

// Create dialog window
  Dialog.create("Analysis parameters (Pre-entered suggestions)");
  Dialog.addMessage("Maximum Threshold for scratch");
  Dialog.addNumber("Value:", Valor1);
  Dialog.addMessage("Minimum Threshold for cell");
  Dialog.addNumber("Value:", Valor2)
  Dialog.addMessage("Minimum size particle for cell");
  Dialog.addNumber("Value:", Valor3)
  Dialog.addMessage("First timepoint folder name");
  Dialog.addString("Value:", Valor4)
  Dialog.addMessage("GIF delay (in ms)");
  Dialog.addNumber("Value:", Valor5)
  Dialog.addMessage("Timepoint number for GIF purposes");
  Dialog.addNumber("Value:", Valor6)
  Dialog.addCheckbox("Check this box to run in batch mode", false);
  Dialog.show();

  Valor1 = Dialog.getNumber();
  Valor2 = Dialog.getNumber();
  Valor3 = Dialog.getNumber();
  Valor4 = Dialog.getString();
  Valor5 = Dialog.getNumber();
  Valor6 = Dialog.getNumber();
  BatchStatus = Dialog.getCheckbox();

// Prints ImageJ Version
print("Fiji/ImageJ Version:", getVersion());
print("Scratch_and_Wound_v0.3.ijm");

// Report the values
  print("Main dir:", dir);
  print("Maximum Threshold for scratch:", Valor1);
  print("Minimum Threshold for cell", Valor2);
```

```
  print("Minimum threshold for cell", Valor2);
  print("Minimum size particle for cell", Valor3);
  print("Starting timepoint", Valor4);

print("Script started!");

// Prints the starttime
print("Script started at:");
     MonthNames =
newArray("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","
Nov","Dec");
     DayNames = newArray("Sun", "Mon","Tue","Wed","Thu","Fri","Sat");
     getDateAndTime(year, month, dayOfWeek, dayOfMonth, hour, minute,
second, msec);
     TimeString ="Date: "+DayNames[dayOfWeek]+" ";
     if (dayOfMonth<10) {TimeString = TimeString+"0";}
     TimeString = TimeString+dayOfMonth+"-"+MonthNames[month]+"-
"+year+"\nTime: ";
     if (hour<10) {TimeString = TimeString+"0";}
     TimeString = TimeString+hour+":";
     if (minute<10) {TimeString = TimeString+"0";}
     TimeString = TimeString+minute+":";
     if (second<10) {TimeString = TimeString+"0";}
     print(TimeString+second);

// Creates an array with all folders in dir
TimepointsToProcess = getFileList(dir);

// Creates an array with all images inside Timepoint start
DirTileList = getFileList(dir + Valor4 + File.separator);

// creates an empty array
TileList = newArray();

// Loop to remove folders
for (Cenoura=0; Cenoura<DirTileList.length; Cenoura++){
        // Check if is an folder (slash on end)
        if (endsWith(DirTileList[Cenoura], "/") == 1) {
                // nothing
        } else {
                // Concatenate array with image name
                TileList = Array.concat(TileList,
DirTileList[Cenoura]);
        }
}

        //
```

```
        print("Available images:");
        // Print tile list
        Array.print(TileList);

        //
        print("Available timepoints:");
        // Print timepoint list
        Array.print(TimepointsToProcess);

// Create folder for results
File.makeDirectory(dir + "Results" + File.separator);
File.makeDirectory(dir + "Results" + File.separator + "Images" +
File.separator);
File.makeDirectory(dir + "Results" + File.separator + "Tables" +
File.separator);

// Enters BatchMode
  if (BatchStatus==true) setBatchMode(true);

// Loop for tiles
for (Cenoura=0; Cenoura<TileList.length; Cenoura++){

        //
        print("Processing " + TileList[Cenoura]);

        //
        // MASK DEFINITION OVER TIMEPOINT BEGIN - START
        //

        // Open the image from timepoint zero
        open(dir + Valor4 + File.separator + TileList[Cenoura]);
        // Crop the image to avoid unaligned borders
        makeRectangle(120, 120, 3840, 2832);
        run("Crop");
        // Enhance Contrast to saturate the same amount of pixels in
all images
        run("Enhance Contrast...", "saturated=0.3 normalize");
        // runs a Sobel edge detector to highlight sharp changes in
intensity of image
        run("Find Edges");
        // Increases contrast and accentuates detail in the image or
selection, but may also accentuate noise.
        run("Sharpen");
        // Run an Threshold limit
        setThreshold(0, Valor1);
        // Convert to mask
```

```
        run("Convert to Mask");
        // Erode and not dilate as the image is inverted
        run("Erode");
        //
        run("Analyze Particles...", "size=200000-50000000
show=Outlines display clear summarize add");
        // Fecha a janela de resultados
    selectWindow("Results");
    run("Close");
        // Close open images
              while (nImages>0) {
         selectImage(nImages);
          close();
                }

        //
        // MASK DEFINITION OVER TIMEPOINT BEGIN - END
        //



        //
        // STACK CREATION
        //

        // Loop for timepoints
        for (Kiwi=0; Kiwi<TimepointsToProcess.length; Kiwi++){

        // Open the image from timepoint zero
        // open(dir + TimepointsToProcess[Kiwi] + File.separator +
TileList[Cenoura]);
        open(dir + TimepointsToProcess[Kiwi] + TileList[Cenoura]);
        // Timepoint loop close
        }

        // Create stack
        run("Images to Stack", "name=Stack title=[] use");

                // Crop the image to avoid unaligned borders
                makeRectangle(120, 120, 3840, 2832);
                run("Crop");

                // Enhance Contrast...
                // process_all uses the whole stack
                // use define stack histogram
                run("Enhance Contrast...", "saturated=0.3 normalize
process_all use");
```

```
                // Save the contrast enhanced stack
                saveAs("Tiff", dir + "Results" + File.separator +
"Images" + File.separator + TileList[Cenoura] + "_01_Contrast");

                //
                run("Sharpen", "stack");
                //
                run("Find Edges", "stack");
                //
                run("Invert LUT", "stack");
                //
                setOption("BlackBackground", false);
                //
                setThreshold(Valor2, 255);
                //
                run("Convert to Mask", "method=Default
background=Light calculate");
                //
                //run("Close-", "stack");
                run("Dilate", "stack");
                //
                run("Select None");

                // Select all ROIs
                // gently from
                // http://imagej.1557.x6.nabble.com/Macro-Language-
select-all-masks-from-ROI-Manager-td5004323.html
                count=roiManager("count");
                array=newArray(count);
                for(i=0; i<count;i++) {
                    array[i] = i;
                }
                roiManager("Select", array);

                // if statement to check if there are more than one
ROI
                   if (array.length>1) {
                        // Combine selected arrays (from last command
                        roiManager("Combine");
                        // Add this new selection as an roi
                        roiManager("Add");
                        // Select the previous array
                        roiManager("Select", array);
                        // Delete
                        roiManager("Delete");
                }
```

```
                        // Select the new (or previous) merged ROI
                        roiManager("Select", 0);

                        //
                        run("Analyze Particles...", "size=" + Valor3
+ "-Infinity show=Outlines clear summarize stack");

                // downscale and save the overlay of selected cells
        selectWindow("Drawing of " + TileList[Cenoura] +
"_01_Contrast.tif");
                run("Scale...", "x=0.2 y=0.2 z=1.0 width=768
height=566 depth=" + Valor6 + " interpolation=Bilinear average
process create");
                saveAs("Tiff", dir + "Results" + File.separator +
"Images" + File.separator + TileList[Cenoura] + "_02_Selected");

                // Save the summary
                selectWindow("Summary of " + TileList[Cenoura] +
"_01_Contrast.tif");
                saveAs("Results", dir + "Results" + File.separator +
"Tables" + File.separator + TileList[Cenoura] + ".csv");
                // Fecha a janela de resultados
           run("Close");
           selectWindow("Summary");
           run("Close");
                // Fecha as imagens abertas
                      while (nImages>0) {
                selectImage(nImages);
                close();
                        }
        //
        // Open contrast enhanced to save with overlay
        open(dir + "Results" + File.separator + "Images" +
File.separator + TileList[Cenoura] + "_01_Contrast.tif");

        // Define overlay
        run("From ROI Manager");
        run("Overlay Options...", "stroke=red width=10 fill=none set
apply");
        saveAs("Tiff", dir + "Results" + File.separator + "Images" +
File.separator + TileList[Cenoura] + "_03_ContrastOverlay");

        // Scale down for gif creation
        run("Scale...", "x=0.2 y=0.2 z=1.0 width=768 height=566
depth=" + Valor6 + " interpolation=Bilinear average process create");
```

```
        // Lowres gif creation
        run("Animated Gif ... ", "name=" + TileList[Cenoura] +
"_03_ContrastOverlay" + " set_global_lookup_table_options=[Do not
use] optional=[] image=[No Disposal] set=" + Valor5 + " number=0
transparency=[No Transparency] red=0 green=0 blue=0 index=0
filename=" + dir + "Results" + File.separator + "Images" +
File.separator + TileList[Cenoura] + "_04_Animated_lowRes.gif");

        // Delete image without overlay
        File.delete(dir + "Results" + File.separator + "Images" +
File.separator + TileList[Cenoura] + "_01_Contrast.tif");
                //
                // Clean roiManager
                roiManager("Delete");

         // Try to release memory to the system
                run("Collect Garbage");
                 run("Close All");
                 call("java.lang.System.gc");
                    // print("Finished
field",TilesASeremProcessadas[Cenoura]);
}

//
print("Results Analyzed!");

// Exits BatchMode
setBatchMode(false);

// Prints the endtime
print("Script finished at:");
     getDateAndTime(year, month, dayOfWeek, dayOfMonth, hour, minute,
second, msec);
     TimeString ="Date: "+DayNames[dayOfWeek]+" ";
     if (dayOfMonth<10) {TimeString = TimeString+"0";}
     TimeString = TimeString+dayOfMonth+"-"+MonthNames[month]+"-
"+year+"\nTime: ";
     if (hour<10) {TimeString = TimeString+"0";}
     TimeString = TimeString+hour+":";
     if (minute<10) {TimeString = TimeString+"0";}
     TimeString = TimeString+minute+":";
     if (second<10) {TimeString = TimeString+"0";}
     print(TimeString+second);

// Saves the log
LogName = dir + "LogFile_Scratch_FindThr" + Valor1 + "_CountThr" +
Valor2 + "_SizeMin" + Valor3 + ".txt";
```

```
selectWindow("Log");  //select Log-window
saveAs("Text", LogName);

// Open an dialog screen on script finish asking about to quit imagej
  Dialog.create("Script finished");
  Dialog.addMessage("Good luck!")
  Dialog.addCheckbox("Check this box to quit ImageJ", false);
  Dialog.show();
  CloseStatus = Dialog.getCheckbox();
  if (CloseStatus==true) run("Quit");


////////////////////////////////////////////
```