Measuring porosity inside a cross-section of a tube

Erwan Plougonven, February 2018
eplougonven@uliege.be

Abstract

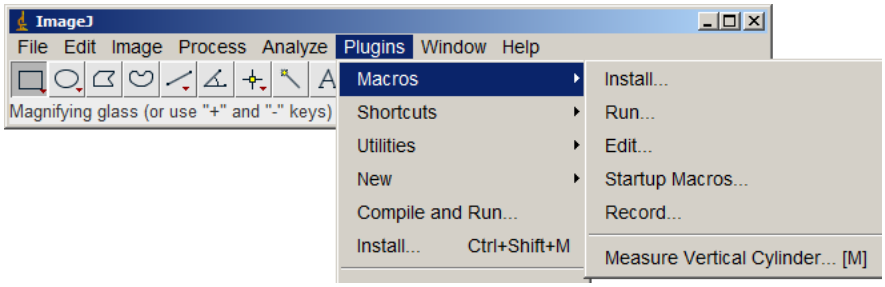
This macro (source code at the end of the document) was written to measure the porosity inside an axial cross-section in a tube. The macro makes four assumptions:

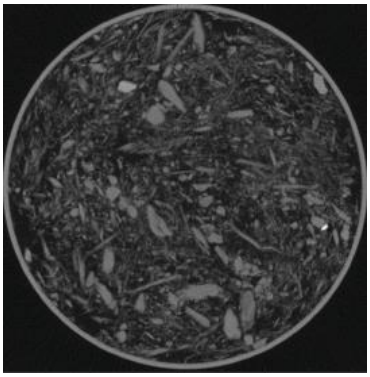
1. The cross-section is perpendicular to the tube axis
2. The tube thickness is uniform
3. No artefacts except noise are present in the images (e.g. beam hardening)
4. The solid appears in light gray over a dark background

From the set of cross-sections to analyze, the macro first thresholds the images, find the region inside the tube and computes the porosity in that region.

Installation

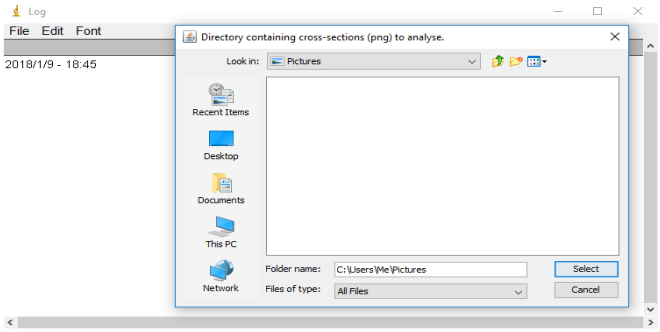
The image processing is implemented in an **ImageJ** macro named Cynthia-PoroMeasure.ijm. ImageJ is open source and available on the internet. The macro is a readable text file requiring at least version 1.49t of imageJ (the sub-menu Help→Update ImageJ... can perform the update automatically). To run the macro, simply go to Plugins→Macros→Run and select the macro file.



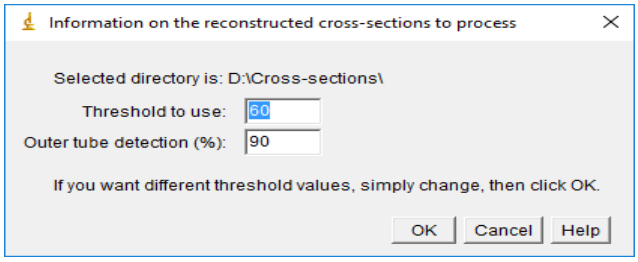


The script was written for cross-sections that resemble the one on the left, in which the porosity is in black. There **must not** be any kind of white border, title, or scale bars, just the object. The cross-sections must also be saved in the PNG format for the script will look for these types of files.

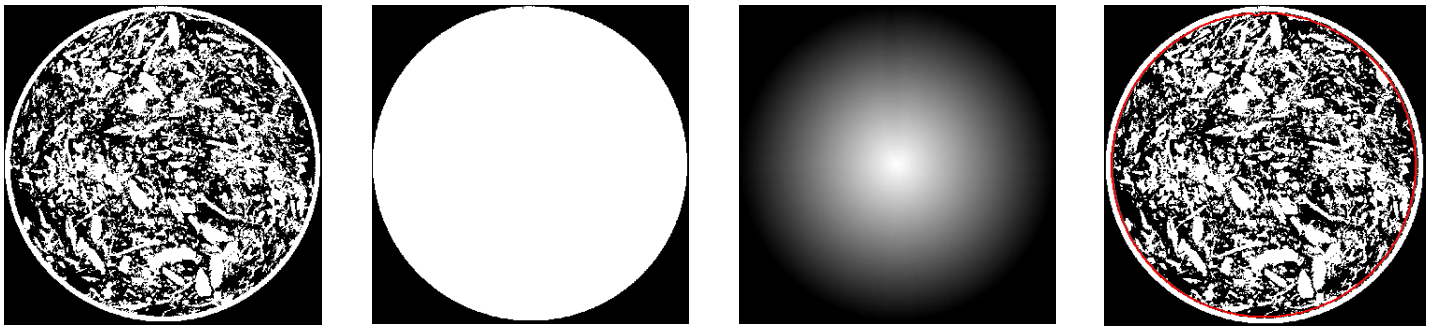
Usage



The process is fairly straightforward. There is no need to open any image before launching the macro from the Plugins menu. It will open a Log window in which all the information on the computation will be displayed, and a dialog window asking where to find all the cross-sections to analyze.



After selecting the directory, it looks for all PNG files in that directory. For each image, it will open it and determine what the threshold value would be if the automatic Isodata method were used. It will then find the average of all these thresholds and suggest that value in "*Threshold to use*" to binarise all the images. For the second parameter, "*Outer tube detection (%)*", the macro will, after binarization (figure 1a), fill the interior of the tube (figure 1b), compute for each pixel in the resulting mask the distance to the exterior (1c), and look at the pixels in the binary image at a distance r, starting at 1, and compute the percentage of solid. If this percentage is above the given value (default: 90%), it will increment r and repeat. Figure 1d shows one such layer of pixels on which the percentage of solid is computed. Once the solid percentage drops below the given value, all pixels in that layer and further in are selected as the region in which to compute the porosity.



a) b) c) d)

Figure 1. Process for finding the region inside the tube.

Results

A **Results.log** file will be created in that directory, along with binary image files showing the segmented cross-section without the tube. These image files are saved as TIFFS so that if the macro is run again these files will not be selected as inputs.

```
2018/1/9 - 19:25
Selected directory: D:\Cross-sections\.
Number of files to process: 3.
Threshold to use: 60.
sec0-420kv-low-5 -- Area: 83148 pixels. Average greylevel: 151.7923. Porosity: 40.4736%.
sec0-420kv-low-7 -- Area: 84271 pixels. Average greylevel: 103.4997. Porosity: 59.4119%.
sec0-420kv-low-9 -- Area: 84295 pixels. Average greylevel: 109.0333. Porosity: 57.2418%.
```

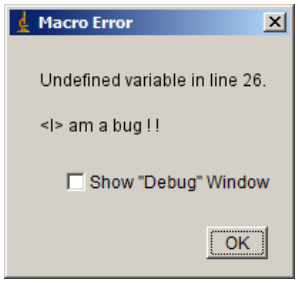
The log file contains the calculated porosity in the interior region, along with the area in pixels and average greylevel (between 0 and 255).

The TIFF files show the binary image of the interior region, all other pixels set to 0, i.e. black.

Notes

The two parameters that control the method (Threshold and Outer tube detection) are merely suggestions, if you find a different threshold value or detection percentage to use, simply change the values before clicking OK.

If a bug is found, it should open a window like the one shown here. If you send me a printscreen of the window and a copy of the Log contents, I'll try a fix it as soon as possible.



The script has been written in such a way as to be a basis for any processing on such cross-sections. If you wish to analyze your tomographic data using imageJ with such a script, either modify a copy of the ijm file that contains this macro (please don't forget cite me as a reference), or ask me.

If you use this macro in your research work, I would appreciate participating as a co-author in any relevant publication.

```
// Macro ImageJ pour mesurer la porosité dans une coupe axiale d'un échantillon cylindrinque
// PEPs - ULg
// Février 2018 - Erwan Plougonven
//
-------------------------------------------------------------------------------------------------
-----------------------------------
requires("1.49t");
getDateAndTime(year, month, week, day, hour, min, sec, msec);

setBatchMode(true);
print("\\Clear");
run("Close All");
//setBatchMode(true);
print(year+"/"+month+"/"+day+" - "+hour+":"+min);
// Define directory
dir = getDirectory("Directory containing cross-sections (png) to analyse.");
print("Selected directory: " + dir);
list = getFileList(dir);
logFN = dir+"Results.log";
if (File.exists(logFN))
    File.delete(logFN);
File.append(year+"/"+month+"/"+day+" - "+hour+":"+min, logFN);

count = 0;
meanTh = 0;
maxGreylevel = 255;
for (i=0; i<list.length; i++)
{
    curFile = toLowerCase(list[i]);
    if( endsWith(curFile,".png") )
    {
        open(dir + curFile);
        run("8-bit");
        setAutoThreshold("IsoData dark");
        getThreshold(lower,maxGreylevel);
        count = count + 1;
        delta = lower - meanTh;
        meanTh = meanTh + (delta/count);
        close();
    }
}

print("IsoData segmentation finds an average threshold value of "+meanTh+" for "+count+"
images found.");

tubeSolPerc = 90;

html = "<html>"
    +"<H3>ImageJ macro for analysis of cylindrical sample cross-sections</H3>"
    +"<H4>Coded in February 2018 by Erwan Plougonven</H4>"
    +"Supposes the cross-sections are png files, with a dark background.<BR>"
    +"Filled-in threshold value is an average of isodata thresholding on all "+count+"
    images of the directory.<BR>"
    +"Tube detection percentage is based on the porosity increase when moving progressively
    inside the object:<BR>"
    +"- Look at outer pixels of mask, count those marked as solid.<BR>"
    +"- If more than "+tubeSolPerc+"%found as solid, then erode and start again. Otherwise
    stop.";

Dialog.create("Information on the reconstructed cross-sections to process");
Dialog.addMessage("Selected directory is: "+dir);
//Dialog.setInsets(0, 20, 20);
Dialog.addNumber("Threshold to use:", meanTh);
//Dialog.setInsets(0, 20, 20);
Dialog.addNumber("Outer tube detection (%):", tubeSolPerc);
//Dialog.setInsets(20, 20, 20);
Dialog.addMessage("If you want different threshold values, simply change, then click OK.");
```

```
Dialog.addHelp(html);
Dialog.show();

th = Dialog.getNumber();
print("Chosen threshold is "+th+".");

File.append("Selected directory: "+dir +".", logFN);
File.append("Number of files to process: "+count +".", logFN);
File.append("Threshold to use: "+th +".", logFN);

scriptStartTime = getTime();

for (i=0; i<list.length; i++)
{
    curFile = toLowerCase(list[i]);
    if( endsWith(curFile,".png") )
    {
        print("\\Update: ("+i+"/"+count+") Analysing file "+list[i]+".");
        open(dir + curFile);
        prefix = File.nameWithoutExtension;
        rename("Img");
        run("8-bit");
        w = getWidth();
        h = getHeight();
        setThreshold(th, 255);
        setOption("BlackBackground", true);
        run("Convert to Mask");
        makeRectangle(0, 0, w, h);
        run("Select All");
        run("Copy");
        newImage("I", "8-bit black", w+2, h+2, 1);
        run("Paste");
        makeRectangle(1, 1, w, h);
        run("Select None");
        selectWindow("Img");
        close();
        selectWindow("I");
        run("Duplicate...", "title=Msk");
        run("Fill Holes");
        run("Options...", "iterations=1 count=1 black edm=Overwrite do=Nothing");
        run("Distance Map");
        getMinAndMax(min, max);
        print("Min and max are : "+min + " and "+max+".");
        r = 0;
        sp = 100;
        max = 10;

        while ( sp > tubeSolPerc && r < max)
        {
            roiManager("reset");
            r = r+1;
            selectWindow("Msk");
            setThreshold(r, r+1);
            run("Create Selection");
            roiManager("Add");
            selectWindow("I");
            roiManager("Select", 0);
            getStatistics(area, mean);
            print("Layer "+r+": average greylevel:"+mean);
            run("Select None");
            sp = mean/2.55;
        }

        if( r == max )
        {
            print("Error: never manager to outline outer tube. Exiting.");
        }
```

```
        else
        {
            print("Found layer of size "+r+" for just inside outer tube (solid percentage of
            "+sp+").");
            roiManager("reset");
            selectWindow("Msk");
            run("Select None");
            setThreshold(0, r);
            run("Create Selection");
            setForegroundColor(0,0,0);
            run("Fill");
            run("Make Inverse");
            setForegroundColor(255,255,255);
            run("Fill");
            roiManager("Add");
            selectWindow("I");
            roiManager("Select", 0);
            getStatistics(area, mean);
            //File.append(prefix+" -- Area: "+area+" pixels. Average greylevel: "+mean+".
            Porosity: "+100-(mean/2.55)+"%.", logFN);
            File.append(prefix+"\n -- Area: "+area+" pixels. Average greylevel: "+mean+".
            Porosity: "+100-(mean/2.55)+"%.", logFN);
        }
        run("Make Inverse");
        setForegroundColor(0,0,0);
        run("Fill");
        run("Select None");
        makeRectangle(1, 1, w, h);
        run("Crop");
        saveAs("TIFF", dir+prefix+"_bin.tif");
        run("Close All");
    }
}
scriptEndTime = getTime();
print("Script execution time : "+(scriptEndTime-scriptStartTime)/1000+" seconds.");
```