**Supplementary Material 1: ImageJ macro code for automated thickness calculations**

The custom ImageJ macro code used to automate thickness calculations through the analysis of z-stacks obtained by a confocal laser scanning microscopy. Notes in bold represent explanatory comments of the code.

1. setBatchMode("hide"); **//runs faster**
2. label = getTitle();
3. getVoxelSize(width, height, depth, unit);
4. dir = getInfo("image.directory") **//directory of current image**
5. rename("orig"); **//helps keep track of original image**
6. getDimensions(width, height, channels, slices, frames);
7. **//calculate half max intensity of all z for each x,y**
8. run("Z Project...", "projection=[Max Intensity]");
9. run("Divide...", "value=2");
10. **//subtract from orig to keep all pixels above half max**
11. imageCalculator("Subtract create stack", "orig","MAX\_orig");
12. setThreshold(1, 255); **//all non-zero are above half max**
13. setOption("BlackBackground", false);
14. run("Convert to Mask", "method=Default background=Light");
15. binaryImage = getImageID(); **//will need binary image for m, Ra, Rq**
16. run("Duplicate...", "duplicate"); **//can work on this one for now**
17. run("32-bit"); **//will need floating point numbers**
18. **//divide by 255 and by step height(0.1) to get height in μm**
19. run("Divide...", "value="+(255/depth)+" stack");
20. **//add up all slices to get thickness**
21. run("Z Project...", "projection=[Sum Slices]");
22. thickness = getValue("Mean"); //mean thickness
23. **//save a histogram of thickness measurement**
24. run("Histogram", "bins=100 use x\_min=0.0 x\_max=10 y\_max=Auto");
25. saveAs("Tiff", dir + label + "Histogram" + "tif");
26. print(label + ", " + thickness);
27. run("Close All"); **//close all the images and histogram**
28. setBatchMode("exit and display");

**Supplementary Material 2: ImageJ macro code for automated thickness and surface roughness calculations**

The custom ImageJ macro code used to calculate thickness and also estimate surface roughness through the analysis of z-stacks obtained by a confocal laser scanning microscopy. Notes in bold represent explanatory comments of the code.

1. setBatchMode("hide"); **//runs faster**
2. label = getTitle();
3. getVoxelSize(width, height, depth, unit);
4. vals = newArray(5); **//array to hold measurements**
5. dir = getInfo("image.directory") **//directory of current image**
6. rename("orig"); **//helps keep track of original image**
7. getDimensions(width, height, channels, slices, frames);
8. run("Z Project...", "projection=[Max Intensity]"); **//max intensity of all z for each x,y**
9. run("Divide...", "value=2"); **//half max intensity of all z for each x,y**
10. imageCalculator("Subtract create stack", "orig","MAX\_orig"); **//keep all pixels above half max**
11. setThreshold(1, 255); **//all non-zero are above half max**
12. setOption("BlackBackground", false);
13. run("Convert to Mask", "method=Default background=Light");
14. binaryImage = getImageID(); **//will need binary image for m, Ra, Rq**
15. run("Duplicate...", "duplicate"); **//can work on this one for now**
16. run("32-bit"); **//will need floating point numbers**
17. run("Divide...", "value="+(255/depth)+" stack"); **//divide by 255 & step height to get height in μm**
18. run("Z Project...", "projection=[Sum Slices]"); **//add up all slices to get thickness**
19. vals[0] = getValue("Mean"); **//mean thickness**
20. run("Histogram", "bins=100 use x\_min=0.0 x\_max=10 y\_max=Auto"); **//make histogram of thickness**
21. saveAs("Tiff", dir + label + "Histogram" + "tif"); **//save thickness histogram**
22. close(); close(); close(); **//closes histogram, sum and 32 bit stack, binary stack now on top**
23. rarq("top", vals); **//do top Ra and Rq calculations using function**
24. rarq("bottom", vals); **//do bottom Ra and Rq calculations using function**
25. **//make a table to print results to**
26. title = "thicknessRoughness"; **//title of table**
27. f = "[" + title + "]"; **//used to print to table**
28. if (!isOpen(title)){ **//if the table doesn't exist, make it**
29. run("Table...", "name=" + title + " width=250 height=250");
30. print(f, "\\Headings:label\tThickness\tRaTop\tRqTop\tRaBottom\tRqBottom");
31. }
32. print(f, label + "\t" + vals[0] + "\t" + vals[1] + "\t" + vals[2] + "\t" + vals[3] + "\t" + vals[4]);
33. run("Close All"); **//close all the images and histogram**
34. setBatchMode("exit and display");
35. function rarq(arg1, arg2) {
36. setBatchMode("hide"); **//runs faster**
37. newImage("temp", "32-bit black", width, height, 1); **//make duplicate image of same width & height**
38. run("Select All"); run("Set...", "value=NaN"); **//NaN pixels (holes in scaffold) not measured**
39. if (arg1 == "top") { begin = slices; end = 1; step = -1; }
40. else { begin = 1; end = slices; step = 1; }
41. vals = arg2; **//result value array**
42. for (x=0; x<width; x++) { **//loop through each x, y starting at top z, march down till first 255**
43. for (y=0; y<height; y++) {
44. for (z=begin; z!=end; z+=step) {
45. selectWindow("Result of orig");
46. Stack.setSlice(z);
47. val = getPixel(x,y);
48. if (val==255){ **//found the surface**
49. selectWindow("temp");
50. setPixel(x, y, z); **//use this z for this x,y position**
51. break; **//exit z the loop**
52. }
53. }
54. }
55. }
56. updateDisplay(); **//updates values on temp image**
57. m = getValue("Mean"); **//m = average of top**
58. run("Duplicate...", " "); **//make new image temp – m**
59. run("Subtract...", "value="+m);
60. run("Divide...", "value="+(255/depth)+" stack");**//divide by 255 & step height to get height in μm**
61. run("Abs"); **//take the abs of this image**
62. sum = getValue("IntDen"); **//sum of this image**
63. Ra = (1 / (width\*height) ) \* sum;
64. run("Square"); **//square topPixelsMinusM**
65. sum = getValue("IntDen"); **//sum of this image**
66. Rq = sqrt( (1/(width\*height)) \* sum );
67. if (arg1 == "top") {vals[1] = Ra; vals[2] = Rq;}
68. else {vals[3] = Ra; vals[4] = Rq;}
69. close(); close(); **//close temp and dupllicate images**
70. }

**Supplementary Material 3: Raw data corresponding to the thickness averages of three different collagen scaffolds as estimated by TEM versus CLSM and depicted in Figure 2b**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Thickness (μm) | | | | | | |
|  | Collagen scaffold A | | Collagen scaffold B | | Collagen Scaffold C | |
| CLSM | TEM | CLSM | TEM | CLSM | TEM |
| 1.5 | 1.4 | 2.8 | 2.2 | 3.4 | 3.2 |
| 1.9 | 1.3 | 2.7 | 2.3 | 3.6 | 3.2 |
| 2.1 | 1.5 | 2.3 | 2.2 | 3.2 | 3.2 |
| 2.5 | 1.4 | 2.4 | 2.3 | 3.6 | 3.1 |
| 1.5 | 1.3 | 2.6 | 2.4 | 3.5 | 3.3 |
| 1.5 | 1.5 | 2.6 |  | 3.7 |  |
| 1.7 | 1.1 | 2.5 |  | 3.4 |  |
| 2.2 |  | 2.7 |  | 3.9 |  |
| 2.1 |  | 2.9 |  | 3.3 |  |
| 1.9 |  | 2.7 |  | 3.4 |  |
| 1.5 |  | 2.1 |  | 3.2 |  |
| 1.8 |  | 2.8 |  | 3.4 |  |
| 1.6 |  | 2.7 |  | 3.4 |  |
| 1.3 |  | 2.5 |  | 3.5 |  |
|  |  | 2.5 |  | 3.6 |  |
|  |  | 2.8 |  | 3.6 |  |
|  |  | 2.3 |  | 3.6 |  |
|  |  | 2.9 |  |  |  |
|  |  | 2.8 |  |  |  |
|  |  | 2.2 |  |  |  |
| Average | 1.8 | 1.4 | 2.6 | 2.3 | 3.5 | 3.2 |
| StDev | 0.3 | 0.14 | 0.12 | 0.1 | 0.1 | 0.1 |