input = "D:/chupikachu/NEW\_IMAGES\_processing/toxicity\_process/crop/";

output = "D:/chupikachu/NEW\_IMAGES\_processing/2D\_Toxicity\_Intensity/histogram\_list/"; //used for get\_auto\_threshold\_value

output2 = "D:/chupikachu/NEW\_IMAGES\_processing/2D\_Toxicity\_Intensity/histogram\_list/"; // used for get\_Histogram\_list

setBatchMode(true);

list = getFileList(input);

list2 = newArray(list.length);

// delete the ".tif" from the name and save them in the list2 array

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

list2[i] = substring(list[i], 0, lengthOf(list[i]) - 4);

}

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

// get\_Histogram\_list\_for\_stack(input, output, list[i], list2[i]);

//}

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

// get\_Histogram\_list\_for\_stack\_sum\_all\_slice\_together(input, output, list[i], list2[i]);

//}

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

// //get\_Statistics(input, output, list[i], list2[i]);

// get\_Histogram\_list(input, output2, list[i], list2[i]);

//}

//// start get\_auto\_threshold\_value

autoThresholdValue = newArray(list.length);

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

open(input + list[i]);

run("8-bit");

setAutoThreshold("Default dark");

getThreshold(lower,upper);

autoThresholdValue[i] = lower;

}

Array.show(autoThresholdValue);

saveAs("results", output + "autoThresholdValue" + ".csv");

//// end get\_auto\_threshold\_value

setBatchMode(false);

function get\_Statistics(input, output, filename, filename\_without\_extension) {

open(input + filename);

getStatistics(area, mean, min, max, std, histogram);

run("Clear Results");

row = nResults;

setResult("Area",row, area);

setResult("Mean",row, mean);

setResult("Min",row, min);

setResult("Max",row, max);

setResult("Std",row, std);

updateResults();

selectWindow("Results");

saveAs("results", output + filename\_without\_extension + "\_Results" + ".csv");

run("Close");

selectWindow(filename);

close();

}

function get\_Histogram\_list(input, output2, filename, filename\_without\_extension) {

nBins = 256;

open(input + filename);

run("8-bit");

run("Clear Results");

row = 0;

getHistogram(values, counts, nBins);

for (i=0; i<nBins; i++) {

setResult("Value", row, values[i]);

setResult("Count", row, counts[i]);

row++;

}

updateResults();

selectWindow("Results");

saveAs("results", output2 + filename\_without\_extension + "\_list" + ".csv");

run("Close");

selectWindow(filename);

close();

}

function get\_Histogram\_list\_for\_stack(input, output, filename, filename\_without\_extension){

nBins = 256;

open(input + filename);

run("Clear Results");

// setOption("ShowRowNumbers", false);

for (slice=1; slice<=nSlices; slice++) {

setSlice(slice);

row = 0;

getHistogram(values, counts, nBins);

for (i=0; i<nBins; i++) {

setResult("Value", row, values[i]);

setResult("Count", row, counts[i]);

row++;

}

updateResults();

selectWindow("Results");

saveAs("results", output + filename\_without\_extension + "\_T" +pad(slice, 2) + ".csv");

}

}

function get\_Histogram\_list\_for\_stack\_sum\_all\_slice\_together(input, output, filename, filename\_without\_extension){

nBins = 256;

open(input + filename);

run("8-bit");

run("Clear Results");

// setOption("ShowRowNumbers", false);

for (slice=1; slice<=nSlices; slice++) {

setSlice(slice);

row = 0;

getHistogram(values, counts, nBins);

for (i=0; i<nBins; i++) {

if (slice == 1){

setResult("Value", row, values[i]);

setResult("Count", row, counts[i]);

row++;

}

if (slice != 1){

setResult("Count", row, getResult("Count", row) + counts[i]);

row++;

}

}

updateResults();

}

selectWindow("Results");

saveAs("results", output + filename\_without\_extension + "\_list" + ".csv");

}

function pad (a, left) {

while (lengthOf(""+a)<left) a="0"+a;

// separator=".";

// while (lengthOf(""+separator)<=right) separator=separator+"0";

// return ""+a+separator;

return ""+a

// reference

// http://imagej.1557.x6.nabble.com/Generate-001-002-name-increments-td3698075.html

}