**Dalmatian plot procedure**

**Protocol 1**: Dalmatian plot procedure.

In the following instruction the Dalmatian plot procedure is described step by step. The instruction is based on a basic description provided by Ingo Fetzer (ingo.fetzer@stockholmresilience.su.se) who also wrote the macro files.

Before you start, have software ImageJ (<http://rsbweb.nih.gov/ij/>) and R (http://r-project.org) with additional package 'vegan' (<http://r-project.org/web/packages/vegan/index.html>) installed.

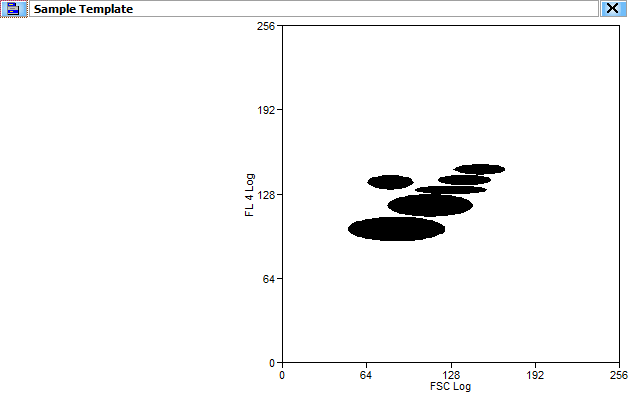
**Part I** Creation of gate images from flow cytometric raw data.

* The software Summit Version 4.3 is used for data analysis. The use of alternative software is possible. The individual steps described in the following might then differ.
* The fcs-files of interest are opened.
* Mark all subcommunities with a gate.
* Optional: Copy statistics as text and paste into Microsoft Excel document.
* Delete the fcs-file so that only the sample template of the gates is visible.
* Save the sample template as graphic in .bmp format.
* Depending on your data analysis software also other image formats like .emf are possible. Do not use .jpg or other compressed formats.
* Perform in this way for all samples.
* Finally, all images should be stored in one folder (which should only include these images without any other data files).

**Part II** Colouring of gates.

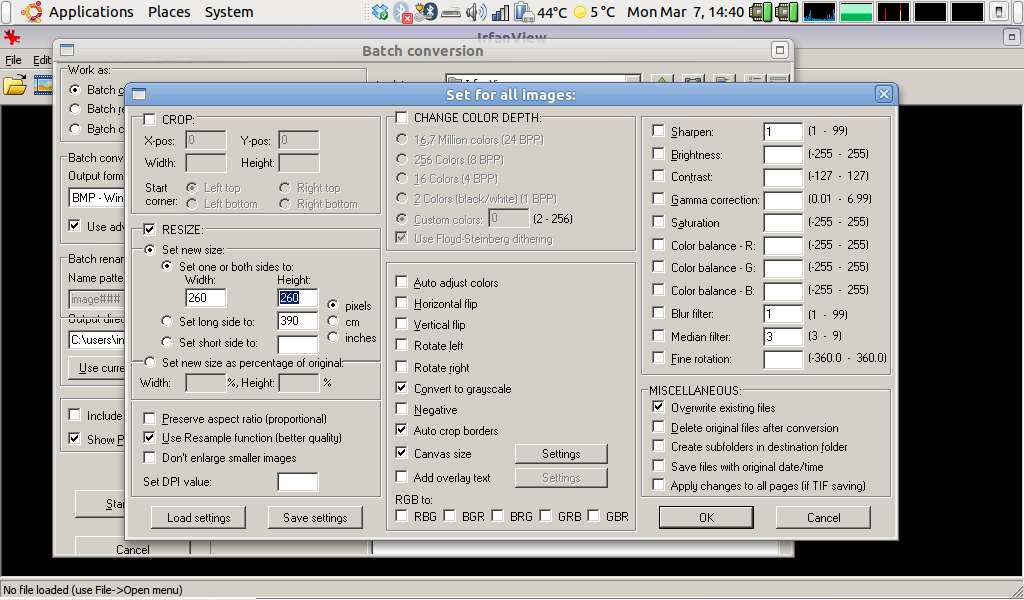
* Open gate images in Microsoft Paint or any alternative software.
* Colour all gates black and remove gate labels.
* Save again as .bmp or .emf file.
* Optional: If abundance information is to be included in the analysis, the gates have to be coloured in grey values according to their abundance. This step can be performed with, e.g., ImageJ.



**Part III** Optional: Batch conversion of pictures from any format into uncompressed pixel format.

This step is only required if your image files are stored in vector image formats (e.g. .wmf, .emf, etc) and you need to convert them into pixel image formats (.gif, .tif, .bmp). Otherwise proceed with Part IV.

* Open software IrfanView (free download: http://www.irfanview.com) /software/IrfanView/i\_view32.exe.
* Go to Files -> Batch Conversion/Rename.
* A new window opens.
* Here tick in 'Work as:' Batch conversion.
* 'Look in:' Choose your folder where your images to convert are stored in.
* Press on the button 'Add all'. Now all files that you want to convert are shown in the list below.
* Go to 'Batch conversion settings:' As 'Output format' choose e.g. BMP and tick 'Use advanced options'.
* Click on button 'Advanced'. A new window will open to set additional necessary Conversion parameters. Tick 'Convert to grayscale', 'Autocrop borders', and 'Canvas size' and set all values in 'Canvas size settings' to -10. Also tick 'Resize' with 'Set new size' to e.g. 260 for both width and height. See that 'Create subfolders in destination folder' under 'Miscellaneous' is not ticked. Finally click on 'Ok'.
* In 'Output directory for files' click on 'Use current directory' and add in the path window above '\convert\' to the path.
* Click on 'Start batch'.



* A new window should open to display the progress of each picture conversion. At the end everything should be 'Ok' and no errors should have occurred.
* Close all windows and quit IrfanView.

**Part IV** Create overlaps from the gate images using ImageJ.

After starting the program you will first need to install a macro. A macro is a simple text file that automates a series of ImageJ commands which can be executed by pressing a key. This macro enables ImageJ to

* read all images from a folder and create overlaps of all combinations of every image that is in there. Remember that if you have N images then 2N additional pictures will be created which, depending on the amount of your images can be many!! So depending on your computer it may take a while.
* count the amount of black pixels in every picture and store them in a text file called 'TotalArea.txt'. This file is the basis one needs for the final computation of the MDS plot and a cluster analysis.

The macro is provided as Supplementary Protocol 2. Copy and save the complete macro as text file, e.g., under the name “overlap\_measure\_binary.txt”.

* Start ImageJ going to /Software/ImageJ/ImagJ.exe.
* To install the macro press 'Ctrl+Shift+M' and select /macros/Overlap\_measure\_binary.txt.
* Now ImageJ should tell you '2 Macros installed'.
* Press button 'F3' and a window should pop up where you select the directory with all your image files. Wait until ImageJ finishes and displays a new window saying 'Creating overlaps finished'. Click on 'Ok'. Now a new folder should have been created called 'overlaps' within your image folder containing your pictures plus all the overlaps as displayed by the file names.



* Press the button 'F4' and go to the newly created folder 'overlaps'. Two new windows will pop up containing all counts of the black pixels from all images. Wait until ImageJ finishes with 'Measurements finished'. Click 'Ok' and also manually close both measurement windows.
* The 'overlap' folder should now contain two new files named
  + 'TotalAreas.txt' containing 'Slice' (=Image file name), 'Count' (=Number of identified objects, ideally this should be equal to number of gates), 'Total Area' (=Total number of black pixels), 'Average Size' (= Average size of each object), and 'Area Fraction' (=relative fraction of how much the black-pixel account for the whole picture).
  + 'SingleObjectMeasurements.txt' containing the label of the image file, the blackpixel-counts of every single gate, and its x and y position on the image.
* You may open both files by drag & dropping them on an open Excel sheet.

**Part V** Calculating similarity matrix, NMDS plot and cluster analysis with R.

For this part you have to use a script for R called 'Calulate\_matrix-general.r' which is provided as Supplementary Protocol 3. The complete procedure can be copied into a text document. The name has to be adapted to the ending .r (e.g. calculate\_matrix-general.r).

* Start R with /Software/R/R-[version]/bin/Rgui.exe. The R-Console window should appear.
* Use the file “Calculate\_matrix-gerneral.r” and drag and drop it on the open R-Console window.
* A new window should open. Navigate to the file 'TotalArea.txt' created previously.
* A plot window with both the NMDS and the cluster analysis will appear. The labeling and the output are very general with labels used from the sample labels directly without symbols and colors. This is done on purpose to provide a general macro which can be used for all purposes and sample sizes. In order to customize groups, symbols and colors according to treatment, environmental conditions one need to add script lines to the R script or do it manually by degrouping the plots in other image manipulation programs (e.g. Microsoft PowerPoint).
* You may save the plot by either right click on the plot and choose 'Copy as metafile' to paste it into any image manipulation program of your choice (e.g. Microsoft PowerPoint, CorelDRAW) or use 'Save as metafile' to store plot. More formats (e.g. .pdf, .tif, .gif) are available choosing in the upper left corner of the R-Console File ‘Save as’.
* In case you need the output of the similarity matrix type in the R console 'mat' and copy and paste the matrix to Microsoft Excel or OpenOffice Calc.

**Protocol 2**: Dalmatian plot macro for ImageJ.

The complete macro can be copied into a text file (e.g. overlap\_measure\_binary.txt) and used for ImageJ according to the procedure described above.

//This macro collection is written for creation of overlap images from gate images of flow cytometric scatter plots

//and final pixel estimation of all images within a folder. Measurement outputs are finally stored in two text files

//This script conducts Step 2 of 3 as described in the manual.

//The script was written by Ingo Fetzer (ingo.fetzer@ufz.de), Dep. Environmental Microbiology, Helmholz-Centre for

//Envrionmental Research - UFZ.

//Current Version 20110307.

macro "Create overlaps [F3]"

{

setBatchMode(true);

var dir = getDirectory("Choose a Directory ");

var list =

getFileList(dir);

overlapfolder = dir+"overlap"+File.separator;

File.makeDirectory(overlapfolder);

for(k=0; k<list.length; k++)

{

var file1 = dir+list[k];

open(file1);

run("Invert");

run("8-bit");

run("Divide...", "value=2.000");

run("Invert");

var title1 = getTitle;

saveAs("bmp", overlapfolder+title1);

close();

}

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

{

for(j=i+1; j<list.length; j++)

{

var file1 = dir+list[i];

var file2 = dir+list[j];

open(file1);

run("8-bit");

run("Divide...", "value=2.000");

var title1 = getTitle;

open(file2);

run("8-bit");

run("Divide...", "value=2.000");

var title2 = getTitle;

imageCalculator("add create", title1, title2);

var index = lastIndexOf(title1, ".");

if (index!=-1) title1 = substring(title1, 0, index);

var index = lastIndexOf(title2, ".");

if (index!=-1) title2 = substring(title2, 0, index);

saveAs ("bmp", overlapfolder+"overlap\_"+title1+title2);

close();

}

}

Dialog.create("Message");

Dialog.addMessage("Creating overlaps finished !");

Dialog.show();

}

macro "Measure total areas [F4]"

{

setBatchMode(true);

run("Clear Results");

run("Set Measurements...", "area area\_fraction limit display redirect=None

decimal=3");

var dir = getDirectory("Choose a Directory ");

var list = getFileList(dir);

//getThreshold(lower, upper);

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

var file = dir+list[i];

open(file);

setAutoThreshold("Default");

//run("Threshold...");

setThreshold(0, 200);

run("Measure");

}

selectWindow("Results");

saveAs("txt", dir+"Results\_Greyscale");

Dialog.create("Message");

Dialog.addMessage("Measurements finished !");

Dialog.show();

}

**Protocol 3**: Dalmatian plot macro for R.

The complete procedure can be copied into a text document. The name has to be adapted to the ending .r (e.g. calculate\_matrix-general.r). Afterwards, it can be used as drag and drop file for R according to the description above.

#This script is written for calculation of similarity matrix and computing MDS plot from

#overlap image pixel counts from flow cytometric scatter plots.

#This script is Step 3 of 3 as described in the manual

#The script was written by Ingo Fetzer (ingo.fetzer@ufz.de), Dep. Environmental Microbiology, Helmholz-Centre for

#Envrionmental Research - UFZ

#Current Version 20100307

dat<-read.delim(file.choose(), row.names=2)

#dat<-read.delim('/home/ingo/Desktop/Test/overlap/Results\_Greyscale.txt', row.names=2)

new.names<-gsub(".bmp","",row.names(dat))

row.names(dat)<-new.names

dat.ol<-dat[grep("overlap",row.names(dat)),]

dat.si<-dat[-c(grep("overlap",row.names(dat))),]

out.temp<-data.frame(t(combn(row.names(dat.si), 2)), t(combn(dat.si[,2], 2)))

out<-data.frame(out.temp, dat.ol[,2])

colnames(out)<-c("Scatter.1","Scatter.2","Pixel.Scatter.1","Pixel.Scatter.2","Pixel.overlap")

rownames(out)<-rownames(dat.ol)

jacc.ind<- 2\*out[,5]/(out[,3]+ out[,4])-1 #jaccard index

jacc.dist<- (out[,5]-((out[,3]+out[,4])-out[,5]))/out[,5] #jaccard distance

out.fin<-data.frame(out,Jaccard=jacc.dist)

size<-nrow(dat.si)

mat<-matrix(nrow=size, ncol=size)

j<-size-1

temp<-0

for (i in 1:j) {

mat[,i]<-c(rep(NA,size-length(out.fin[(temp+1):(temp+j),6])), out.fin[(temp+1):(temp+j),6])

temp<-temp+j

j<-j-1

}

diag(mat)<-0

colnames(mat)<-rownames(mat)<-rownames(dat.si)

#mat[mat<0]<-0

mat.dist<-as.dist(mat)

library(vegan)

par(mfrow=c(1,2), pty="s")

mds<-metaMDS(mat.dist, autotransform=F)

plot(mds, type="t")

plot(hclust(mat.dist, "average"), hang=-1)