**Supplemental File 1. Conversion of flow cytometric scatter profiles to cellular volumes (Image J implementation)**

Image stacks of GFP-expressing *Leishmania* parasites were obtained by spinning-disk confocal microscopy (see main methods for more detail). We implemented a ImageJ script to automatically identify parasites and measure their volumes {Schindelin, 2012 #40}. This script takes the 3D confocal microscopy stack and re-slices it along the x- and y-axes to create two new images. Automatic local thresholds are then calculated for each slice of the x-, y-, and z-stacks using Bernsen’s method {Sezgin, 2004 #83}. Each thresholded stack is then re-sliced to the original orientation and all three stacks are added together. This yields an image stack where the value of each voxel corresponds to the number of dimensions in which it was deemed to be part of a cell. Cell volumes were reliably defined by those voxels identified in 2 or 3 dimensions. Volumes were estimated using the ImageJ 3D Objects Counter with the threshold set to 2 {Bolte, 2006 #84}. Cells in direct contact were occasionally counted as one object. These were manually identified and corrected by dividing the volume of the combined object by the number of cells within it.

## 3\_Way\_Local\_Threshold.ijm

// 3\_Way\_Local\_Threshold

// John I Robinson

method = "Bernsen";

radius = "25";

// create working copy of current image

tempTitle = getTitle() + "-thresholded";

run("Duplicate...", "title=" + tempTitle + " duplicate");

setBatchMode(true);

rename(tempTitle);

run("8-bit");

// reslice stack from top and left

run("Reslice [/]...", "output=0.270 start=Top avoid");

rename("top");

selectWindow(tempTitle);

run("Reslice [/]...", "output=0.270 start=Left avoid");

rename("left");

// calculate local threshold on each view

selectWindow(tempTitle);

run("Auto Local Threshold", "method=" + method + " radius=" + radius + " parameter\_1=0 parameter\_2=0 white stack");

selectWindow("top");

run("Auto Local Threshold", "method=" + method + " radius=" + radius + " parameter\_1=0 parameter\_2=0 white stack");

selectWindow("left");

run("Auto Local Threshold", "method=" + method + " radius=" + radius + " parameter\_1=0 parameter\_2=0 white stack");

// convert back to standard view

selectWindow("top");

run("Reslice [/]...", "output=0.212 start=Top avoid");

selectWindow("left");

run("Reslice [/]...", "output=0.212 start=Top rotate avoid");

// merge results

selectWindow(tempTitle);

run("Subtract...", "value=254 stack");

selectWindow("Reslice of top");

run("Subtract...", "value=254 stack");

selectWindow("Reslice of left");

run("Subtract...", "value=254 stack");

imageCalculator("Add stack", tempTitle,"Reslice of top");

imageCalculator("Add stack", tempTitle,"Reslice of left");

// apply a useful LUT so we can see the results

reds = newArray(256);

reds[3] = 255;

greens = newArray(256);

greens[1] = 121;

blues = newArray(256);

blues[2] = 255;

setLut(reds, greens, blues);

// clean up

selectWindow("top");

close();

selectWindow("left");

close();

selectWindow("Reslice of top");

close();

selectWindow("Reslice of left");

close();

setBatchMode(false);

print("Done processing " + tempTitle);

selectWindow(tempTitle);