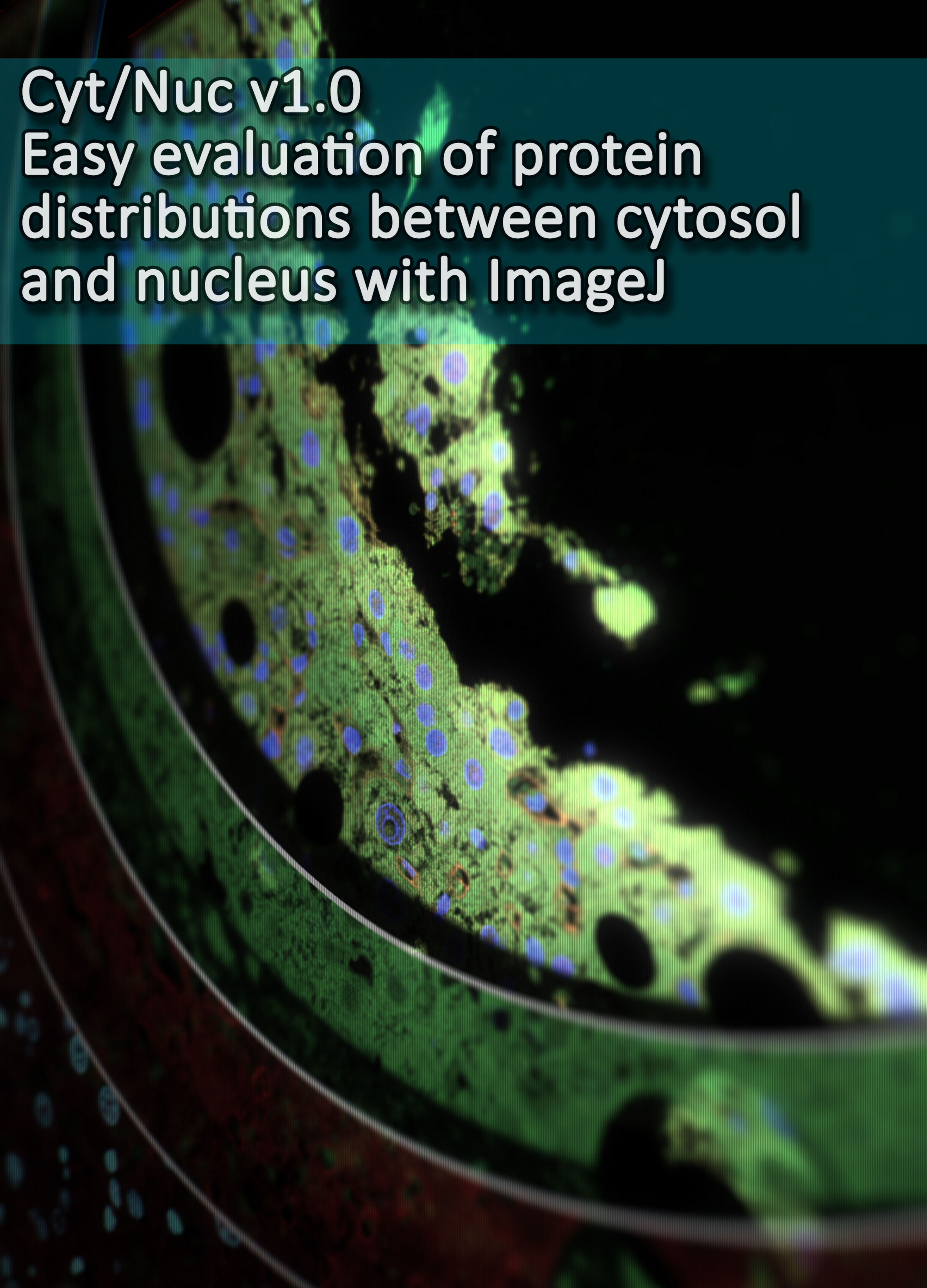
Cyt/Nuc v1.0
Easy evaluation of protein distributions between cytosol and nucleus with ImageJ



Cyt/Nuc, version 1.0 with comments:

```
Line 01   filetype=".bmp";
Line 02   Thresholdlowblue = 100;
Line 03   Thresholdlowgreen = 30;
Line 04   Nucsize=300;
Line 05   Thresholdhighblue = 255;
Line 06   Thresholdhighgreen = 255;
Line 07
Line 08   run("Colors...", "foreground=black background=black selection=yellow");
Line 09   run("Set Measurements...", "area mean standard modal min centroid center perimeter
          bounding fit shape feret's integrated median skewness kurtosis area_fraction stack limit
          display redirect=None decimal=3");
Line 10
Line 11   dir=getDirectory("Choose a Directory");
Line 12   list = getFileList(dir);
Line 13
Line 14   getDateAndTime(year, month, dayOfWeek, dayOfMonth, hour, minute, second, msec);
Line 15   MonthNames =
          newArray("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec");
Line 16   if (hour<10) {hours = "0"+hour;}
Line 17   else {hours=hour;}
Line 18   if (minute<10) {minutes = "0"+minute;}
Line 19   else {minutes=minute;}
Line 20   if (month<10) {months = "0"+(month+1);}
Line 21   else {months=month+1;}
Line 22   if (dayOfMonth<10) {dayOfMonths = "0"+dayOfMonth;}
Line 23   else {dayOfMonths=dayOfMonth;}
Line 24
Line 25   resultsDir = dir + "Results "+year+"-"+MonthNames[month]+"-"+dayOfMonths+"
          "+hours+"h"+minutes+ File.separator;
Line 26   File.makeDirectory(resultsDir);
Line 27
Line 28   for (i=0; i<list.length; i++) {
Line 29       if (endsWith(list[i], filetype)){
Line 30               print(i + ": " + dir+list[i]);
Line 31             open(dir+list[i]);
Line 32             imgName=getTitle();
Line 33
Line 34   run("Split Channels");
Line 35
Line 36   selectWindow(imgName + " (blue)");
Line 37   setAutoThreshold("Default dark");
Line 38   setThreshold(Thresholdlowblue, Thresholdhighblue);
Line 39   run("Analyze Particles...", "size=Nucsize-Infinity display clear include summarize add
          in_situ");
Line 40   close();
Line 41
Line 42   if (roiManager("count")>0) {
Line 43   selectWindow(imgName + " (green)");
Line 44   run("Duplicate...", " ");
Line 45   roiManager("Show All");
Line 46   a = newArray(roiManager("count"));
Line 47
Line 48   if (roiManager("count")>1) {
Line 49   for (j=0; j<roiManager("count"); j++) a[j]=j;
Line 50   roiManager("select", a);
Line 51   roiManager("Combine");
Line 52   roiManager("Update");
```

```
Line 53   }
Line 54
Line 55   if (roiManager("count")==1) {
Line 56   roiManager("Select", 0);
Line 57   }
Line 58
Line 59   setForegroundColor(0, 0, 0);
Line 60   run("Fill", "slice");
Line 61   saveAs(resultsDir+imgName+ "-cytosol"+filetype);
Line 62   close();
Line 63
Line 64   roiManager("Show All");
Line 65   a = newArray(roiManager("count"));
Line 66
Line 67   if (roiManager("count")>1) {
Line 68   for (j=0; j<roiManager("count"); j++) a[j]=j;
Line 69   roiManager("select", a);
Line 70   roiManager("Combine");
Line 71   roiManager("Update");
Line 72   }
Line 73
Line 74   if (roiManager("count")==1) {
Line 75   roiManager("Select", 0);
Line 76   }
Line 77
Line 78   run("Make Inverse");
Line 79   setForegroundColor(0, 0, 0);
Line 80   run("Fill", "slice");
Line 81   saveAs(resultsDir+imgName+ "-nucleus"+filetype);
Line 82   run("Close All");
Line 83
Line 84   open(dir+list[i]);
Line 85           imgName=getTitle();
Line 86   run("Restore Selection");
Line 87   run("Flatten");
Line 88   saveAs("Jpeg", resultsDir+imgName+ "-mask image.jpg");
Line 89   close();
Line 90
Line 91   open(dir+list[i]);
Line 92           imgName=getTitle();
Line 93
Line 94   run("Split Channels");
Line 95
Line 96   selectWindow(imgName + " (blue)");
Line 97   setAutoThreshold("Default dark");
Line 98   setThreshold(Thresholdlowblue, Thresholdhighblue);
Line 99   run("Restore Selection");
Line 100  run("Flatten");
Line 101  saveAs("Jpeg", resultsDir+imgName+ "-mask blue.jpg");
Line 102  selectWindow(imgName + " (green)");
Line 103  setAutoThreshold("Default dark");
Line 104  setThreshold(Thresholdlowgreen, Thresholdhighgreen);
Line 105  run("Restore Selection");
Line 106  setForegroundColor(0, 0, 0);
Line 107  run("Fill", "slice");
Line 108  run("Flatten");
Line 109  saveAs("Jpeg", resultsDir+imgName+ "-mask green.jpg");
Line 110  }
Line 111
Line 112  b = newArray(roiManager("count"));
```

```
Line 113  if (roiManager("count")>1) {
Line 114  for (k=0; k<roiManager("count"); k++) b[k]=k;
Line 115  roiManager("Select", b);
Line 116  roiManager("Delete");
Line 117  }
Line 118
Line 119  if (roiManager("count")==1) {
Line 120  roiManager("Select", 0);
Line 121  roiManager("Delete");
Line 122  }
Line 123
Line 124  run("Close All");
Line 125      }
Line 126  }
Line 127
Line 128  run("Clear Results");
Line 129  dir2=resultsDir;
Line 130  print(dir2);
Line 131  list = getFileList(dir2);
Line 132  for (p=0; p<list.length; p++) {
Line 133      if (endsWith(list[p], filetype)){
Line 134              print(p + ": " + dir2+list[p]);
Line 135            open(dir2+list[p]);
Line 136            imgName=getTitle();
Line 137        setAutoThreshold("Default dark");
Line 138        setThreshold(Thresholdlowgreen, Thresholdhighgreen);
Line 139        rename(imgName);
Line 140        run("Measure");
Line 141
Line 142        run("Close All");
Line 143      }
Line 144  }
Line 145
Line 146  dest=resultsDir+"results.txt";
Line 147  selectWindow("Results");
Line 148  print("[Results]","");
Line 149  saveAs("results", dest);
Line 150  File.append("*****  Settings  ***** " + "Bluelow: " + Thresholdlowblue + "; Greenlow: " +
          Thresholdlowgreen + "; Bluehigh: " + Thresholdhighblue + "; Greenhigh: " +
          Thresholdhighgreen + "; NucSize: " + Nucsize, dest);
Line 151  File.append(" *****  Evaluation date and time ***** "+year+"-"+MonthNames[month]+"-
          "+dayOfMonths+"-"+hours+"h"+minutes, dest);
Line 152
Line 153  selectWindow("Summary");
Line 154  saveAs("text", resultsDir+"summary.txt");
Line 155  File.append("*****  Settings  ***** " + "Bluelow: " + Thresholdlowblue + "; Greenlow: " +
          Thresholdlowgreen + "; Bluehigh: " + Thresholdhighblue + "; Greenhigh: " +
          Thresholdhighgreen + "; NucSize: " + Nucsize, resultsDir+"summary.txt");
Line 156  File.append(" *****  Evaluation date and time ***** "+year+"-"+MonthNames[month]+"-
          "+dayOfMonths+"-"+hours+"h"+minutes, resultsDir+"summary.txt");
Line 157
Line 158  selectWindow("Results");
Line 159  run("Close");
Line 160  selectWindow("Log");
Line 161  run("Close");
```

Composition and function of this script:

Line 01-06: This part is the only one that really needs users' input. The file type (".bmp" in this case) and the according thresholds used for structure recognition have to be specified. The range of those parameters can be easily estimated by loading a representative image from the stack that has to be evaluated. The individual channels are separated via choosing »Image > Color > Split Channels« in ImageJ. At next, the according images are selected (here, representing the channels green, and blue) and via »Image > Adjust > Threshold« the thresholds' lower and upper limits are defined according to the mask (in red) applied to the image, representing the structures that have to be recognized. The upper limit remains in most cases unchanged at 255 (8-bit images), or 4095 (12-bit), respectively. The parameter "Nucsize" defines the area (in pixels) of the smallest structure that is recognized as nucleus (in this example in the blue channel).

Line 08-09: Here, settings are applied that ensure proper function of this macro and define the data displayed in the results table (after evaluation). Feel free to change those settings just by remove of the according parameters. The original line can be kept by adding // in the beginning. Lines that begin with "//" are ignored. This is also useful for annotations. You may also remove the **Lines 158-161** in order to edit the displayed data: "Results"-Window>"Results">"Set Measurements…".

Line 11-12: After starting this macro, the user is asked to select the folder containing the images to process.

Line 14-26: In the selected folder a new subfolder named "results" (date and time of evaluation are included in the name) is created that will store the data produced by this macro during image analysis.

Line 28-32: This is the main loop, responsible for processing every single image in the specified folder. The loop opened in **Line 28** is closed in **Line 126**. Everything between the curly brackets is repeated i-times, in which i is the list-length of the filenames in the image-folder.

Line 34: The original image is split in its red, green, and blue channels.

Line 36: The DAPI channel (in this case blue) is selected. In more recent versions of ImageJ this command may be replaced by "*selectWindow("C3-"+imgName)*" due to a new window designation, as well as in all other lines including this command ("C1-" for the red, "C2-" for the green and "C3-" for the blue channel, respectively).

Line 38-39: The specified parameters (**Line 02-06**) are applied.

Line 42-57: This part prevents the script from stopping in cases like "no nucleus can be identified" or "only a single nucleus is identified". Images without any nuclei are ignored and skipped, listed only in the "Summary Table", but not in the final "Results Table", while the macro proceeds.

Line 59-110: This lines record what is recognized as nucleus and cytosol in form of additional images, enabling the user to reconstruct later what the macro was doing exactly. Those images may be helpful for customization of the parameters defined in the **Lines 02-06**, if necessary.

Line 112-122: This part prevents problems that may occur, if images without any recognized nuclei are processed after images with nuclei. Without erasing the content of the "ROI manager" (ROI stands for "region of interest", the manager enables selection and editing of recognized ROIs), the mask from the previous image (with nuclei) is applied to the current one (without nuclei) and falsifies the result.

Line 124: This command closes all open images.

Line 128: This removes unimportant data from the "Results Table".

Line 129-144: The actual evaluation process starts after preprocessing of the original images, by evaluating the files stored in the folder named "results". In this folder both BMP- and JPEG-images

are stored, but as specified in **Line 01**, only the BMP-files are processed, that contain the structures recognized as “cytosol” and “nuclei”.

Line 146-156: This saves the content of the “Results Table” in the file “results.txt” as well as the settings that were used for image analysis; the content of the “Summary”-window is saved as “summary.txt”. Both date and time of the according evaluation are also included.

Line 158-161: Closes both “Results Table” and “Log”-window after saving the important data.

A script splitting large images into tiles

```
Line 01  tilesrowcolumn=4;
Line 02  filetype=".bmp";
Line 03
Line 04  dir=getDirectory("Choose a Directory");
Line 05  splitDir=dir + "\Tiles\\";
Line 06  File.makeDirectory(splitDir);
Line 07  list = getFileList(dir);
Line 08
Line 09  for (i=0; i<list.length; i++) {
Line 10      if (endsWith(list[i], filetype)){
Line 11              print(i + ": " + dir+list[i]);
Line 12             open(dir+list[i]);
Line 13             imgName=getTitle();
Line 14  run("Montage to Stack...", "images_per_row=tilesrowcolumn
Line 15  images_per_column=tilesrowcolumn border=0");
Line 16
Line 17  for (j=1; j<= tilesrowcolumn*tilesrowcolumn; j++)
Line 18  {
Line 19         setSlice(j);
Line 20         saveAs(splitDir+imgName+"-0"+j+ filetype);
Line 21  }
Line 22         run("Close All");
Line 23      }
Line 24  }
```

Composition and function of this script:

Line 01: Here, the number of tiles per row and column is defined, in this case 4. Thus, the images processed by this macro are subdivided into 4x4=16 tiles.

Line 02: This line specifies the file type for both the original images that have to be processed and the resulting tiles (BMP in this case). Here, saving in a lossless image format is recommended to avoid color and structure artefacts.

Line 05: The name of the folder that has to be created to store the tiled images (“Tiles”, in this case).

Line 17-21: This loop saves the generated tiles and is consequently repeated for tilesrowcolumn^2 as defined in **Line 01**.

Dear user, this is a short manual explaining the use of the ImageJ-macro "CytNuc" v.1.0.

This package contains also the following:

- An large original image with two fluorescence-channels (blue for DAPI, green for immunoproteasome; red channel contains no information)
- A tiled version of the large original image
- The images produced by the macro during analysis (showing recognized structures like nuclei and cytosol)
- Files containing the results (results.txt) and the settings used (summary.txt) during evaluation.

Dear user, please read the following short "manual" in order to provide an adequate function of the "Cyt/Nuc"-macro.

Necessary ImageJ/Fiji-settings.
To ensure, that the macros will work properly and produce reliable results, please apply the following settings to your version of ImageJ/Fiji:

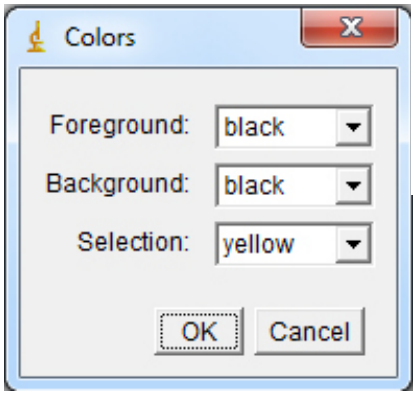


Fig. 1:
Color settings of ImageJ/Fiji

In the software select "Edit" > "Options" > "Colors":
These are the default program settings (**Fig. 1**), they should be left that way (both fore- and background set black) in order to get the macro work properly.

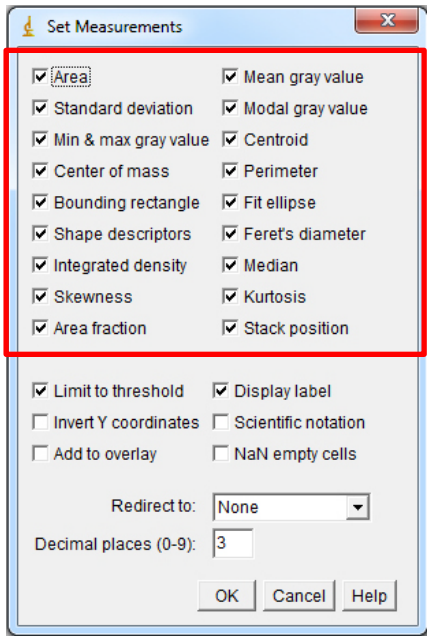


Fig. 2:
Measurement settings of ImageJ/Fiji
The red box indicates the data/parameters that will be extracted from your images.

Open a random image, "Analyze" > "Measure" > open "Results" in the Results-Window > "Set Measurements":

As shown in **Fig. 2**, especially "**Limit to threshold**" has be checked to ensure proper function of our macro that uses thresholds to identify structures. If you are not sure about the information you want to extract from your images (**Fig. 2, red box**), select them all – you can remove unnecessary data later.

The setting of the measurements that are displayed can also applied/changed by adding the following line at the beginning of (any) macro:

run("Set Measurements...", "area mean standard modal min centroid center perimeter bounding fit shape feret's integrated median skewness kurtosis area_fraction stack limit display redirect=None decimal=3");

Settings of the macro:

If you use the sample images (the tiled ones in order to process several images in a single folder), three parameters have to be defined:

- The lower threshold of blue channel ("Thresholdlowblue") in Line 02 (Table 1 of our publication).
- The lower threshold of green channel ("Thresholdlowgreen") in Line 03 (Table 1 of our publication).
- The average area of a single nucleus ("Nucsize") in Line 04 (Table 1 of our publication).

The settings used in the sample images (tiled ones) are
thresholdlowblue = 45
thresholdlowgreen = 10
Nucsize = 200

Those settings work fine with the tiled sample images.

For other images they have to be adapted using a representative one.

How to estimate/adapt the three necessary values in an unknown image?

- Start ImageJ
- Open a sample image (just by drawing it over the program window)
- Select “Image” > “Color” > “Split Channels”; the image is split up in its three channels red (empty in this case), green and blue
- Select the blue channel (visible in the name of the according window)
- “Image” > “Adjust” > “Threshold”
- Move the upper slider of the “Threshold”-window until the image looks like this: All nuclei are red. Pixels that are not red are excluded by the lower limit of the threshold. In this case “thresholdlowblue” is 35 (see **Fig. 3**).

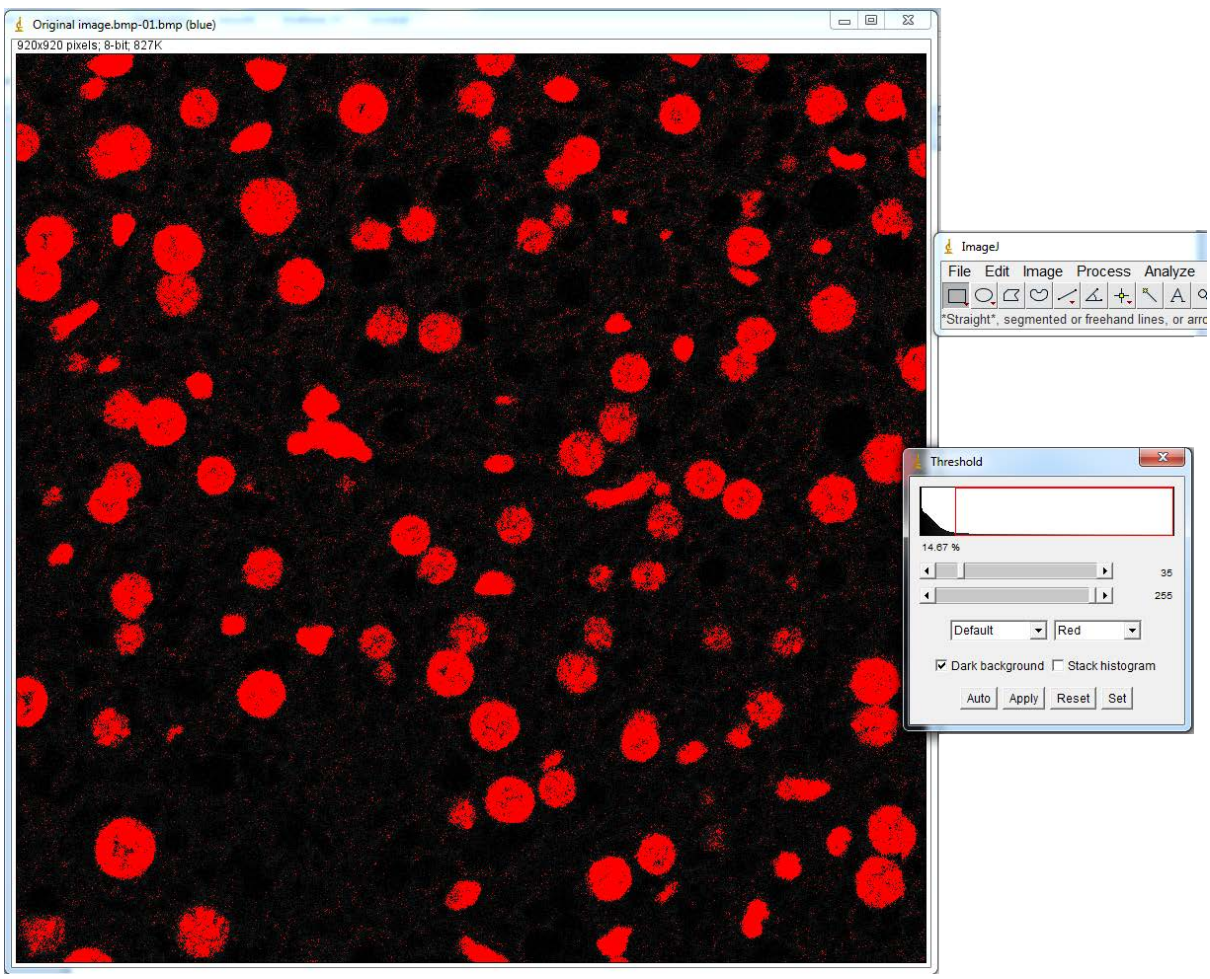


Fig. 3: Threshold settings for proper recognition of the nuclei.

- Now, select “Analyze” > “Analyze Particles” from the software.

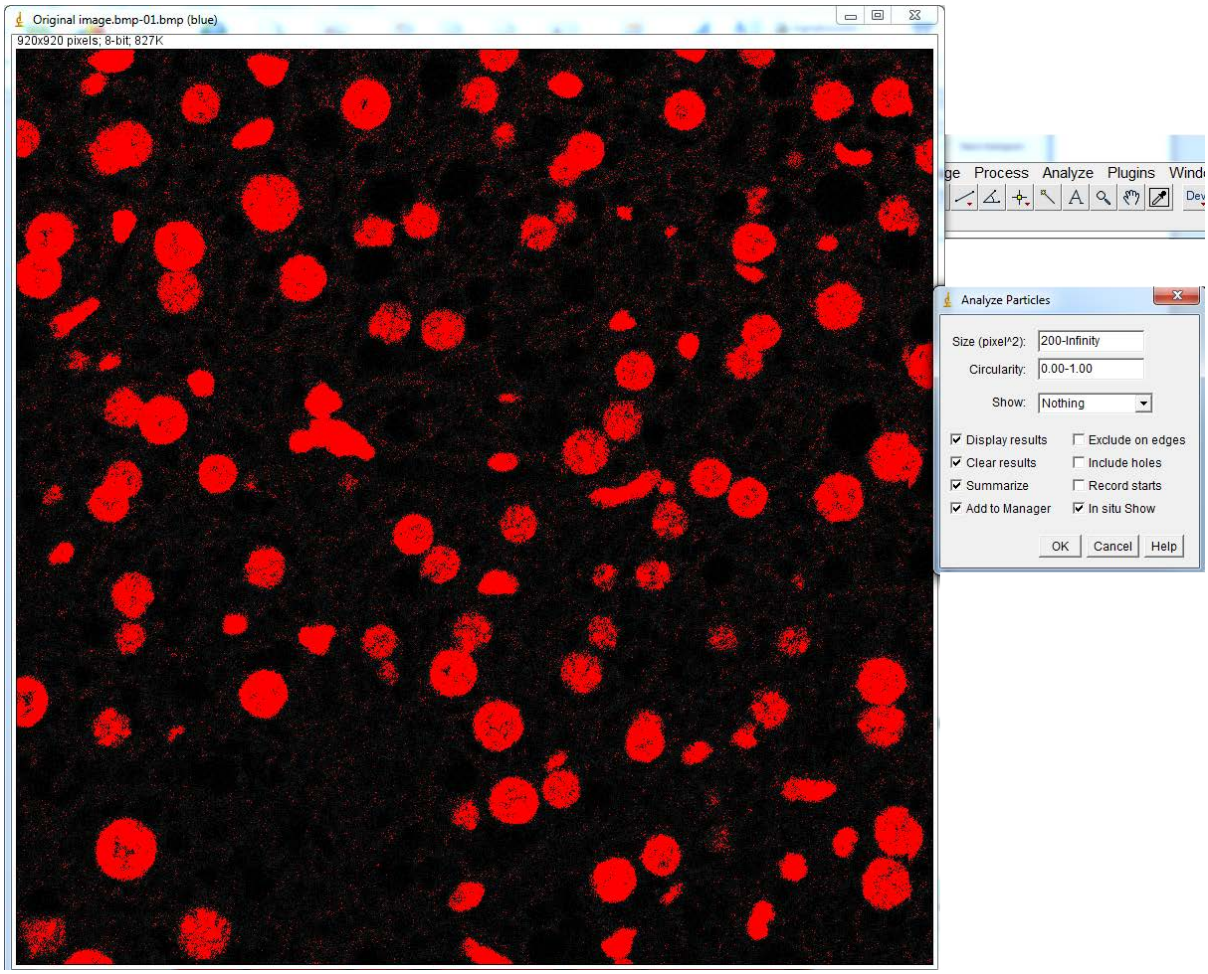


Fig. 3b: Settings for “Analyze Particles” after setting of the threshold for the nuclei.

- In this case, the area of the nuclei is in a range larger than 200 pixels (see **Fig. 3b**).

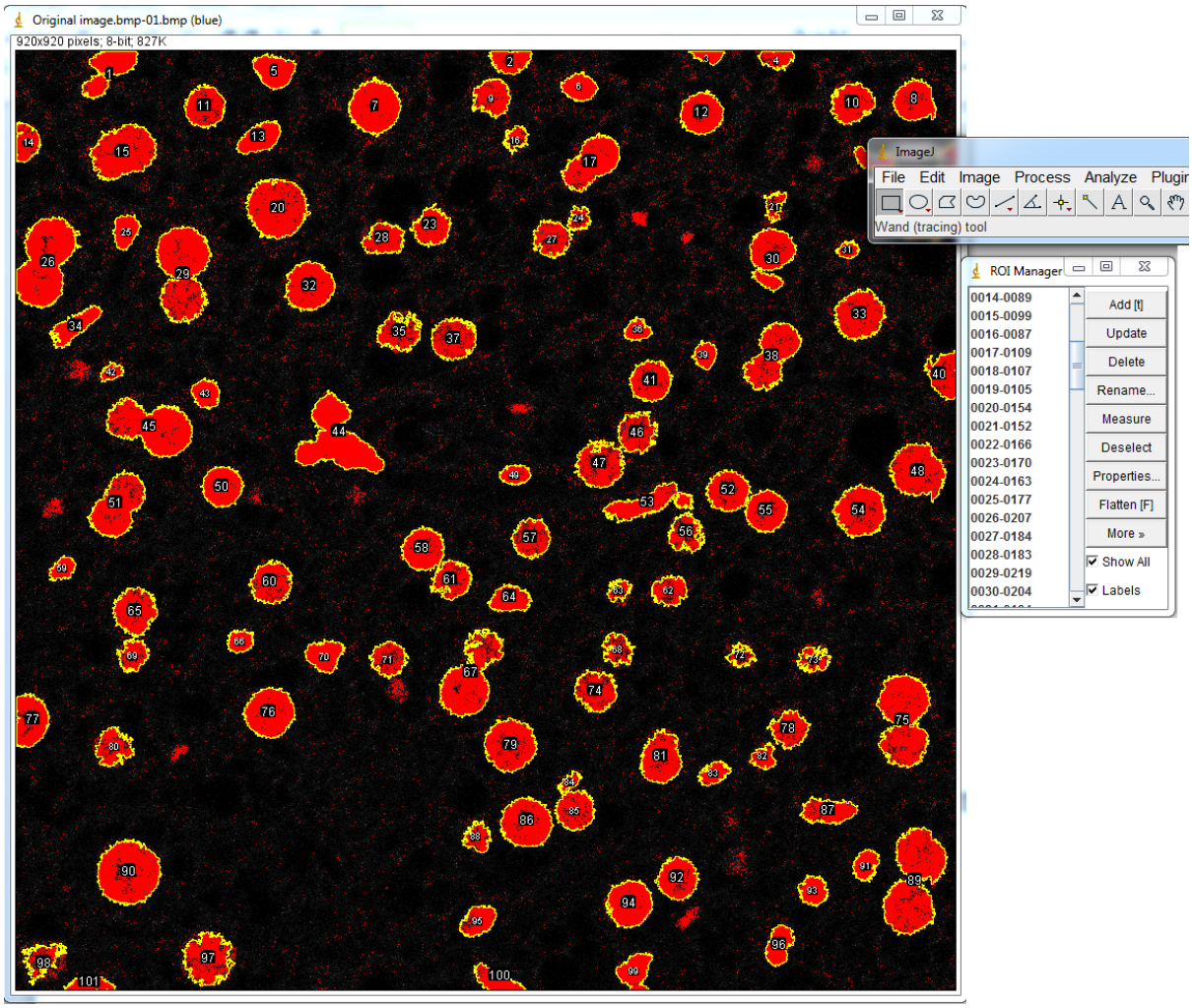


Fig. 4: Recognized nuclei using the "Analyze Particles" function.

- Now, the image should look like this (**Fig. 4**), the structures above "thresholdlowblue" are recognized as particles. => Here, the "Nucsize" = 200.

- Close the image containing the blue channel and select the green one. Repeat the procedure to determine the threshold.

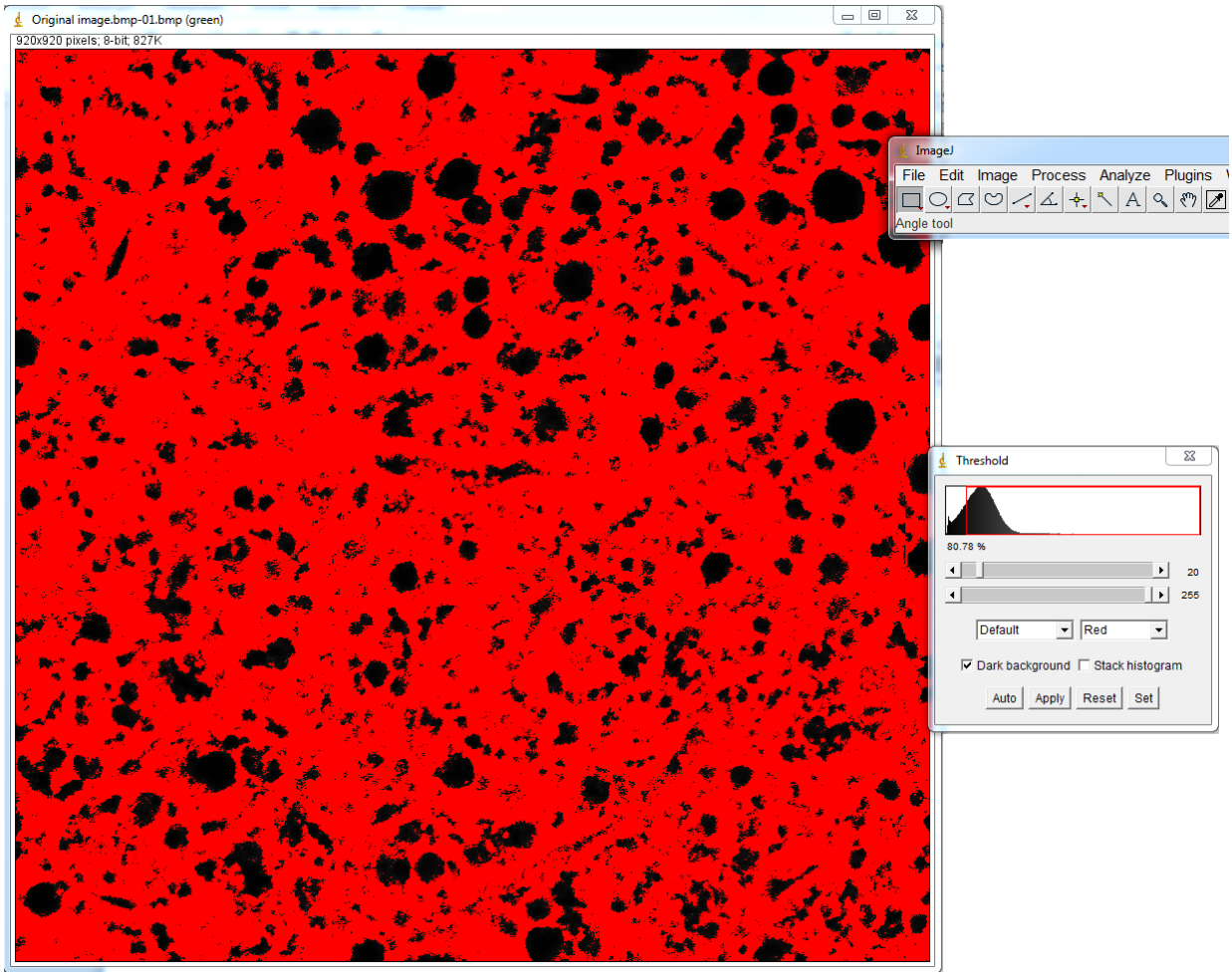


Fig. 5: Threshold for the channel containing the protein of interest.

=> In this case, the “thresholdlowgreen” is 20 (see **Fig. 5**).

Those are the three main values (have to be adjusted only once) that are applied to the whole bunch of images.

How to start a macro?

- Select “Plugins” > “Macros” > “Record”; the “Recorder”-window opens. This window records every single action done in the software now. Paste the macro-code from the publication here.
- Click on “Create” in the “Recorder”-window. Then close the “Recorder”-window.
- The “Macro.ijm”-Window opened. Here, you can still change the three basic setting; select “Macros” > “Run Macro”.
- Now, the macro asks for the folder containing the images to process. Just select the folder, do not open it (via doubleclick or something). Click “Select”.

=> The macro starts, all BMP-images in the folder will be processed now. All results are saved in a new subfolder.

A short discussion of the macro:

Images that contain a "hole" (red arrow) are not a problem.

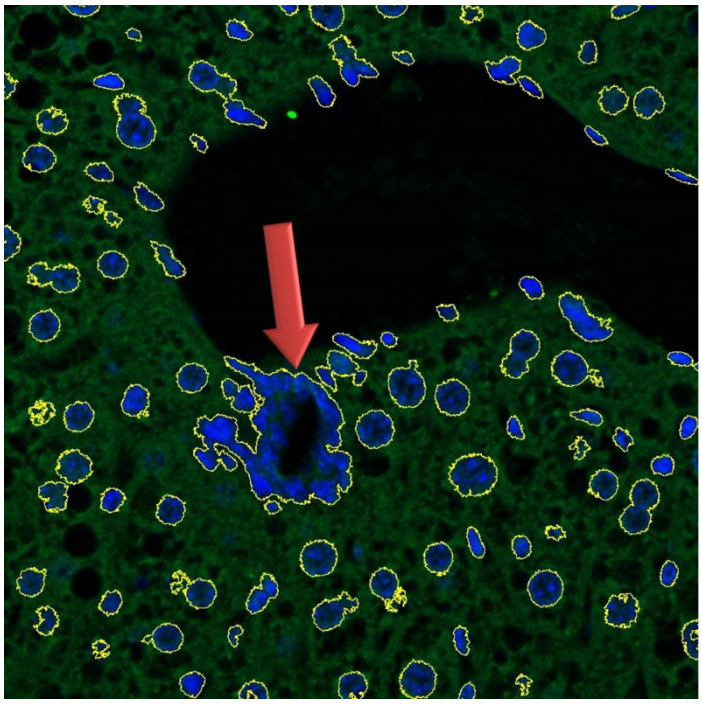


Fig. 6: A dark area that is included by a ring formed of nuclei.

The dark area inside of the ring formed by nuclei (**Fig. 6**) is completely ignored, since the threshold value set for the blue channel clips pixels beyond a certain intensity. Thus, the dark area does not falsify the measurement.

Why there is no automatized recognition of the threshold integrated?

Most images from an experimental series will not have significant differences in DAPI-staining. Though, if an automatized setting of the threshold value for the blue channel ("thresholdlowblue") is necessary, the following code may be helpful:

Line 01	setAutoThreshold("Mean dark");
Line 02	getThreshold(lower,upper);
Line 03	setThreshold(lower*0.65,threshmax);
Line 04	if (lower<=110){
Line 05	setThreshold(lower*0.9,threshmax);
Line 06	}
Line 07	if (lower<=60){
Line 08	setThreshold(lower,threshmax);
Line 09	}

Function of this macro:

Line 01: Settings of the threshold-window
Line 02: Reads out the automatic threshold suggested by ImageJ
Line 03: This values tend to be too high in bright images, thus they are "corrected" by the multiplicator 0.65.
Line 04-05: In images of average brightness, the suggested threshold is quite good, thus the correction factor is reduced to 0.9
Line 07-08: In darker images, the suggested threshold has not to be corrected.

What, if the blue channel has to be applied not to one, but to two different channels?

Our first attempt was to integrate both necessary codes ("blue channel mask applied to the green channel" and "blue channel mask applied to the red channel") to a single one. The final, very quick solution was basically to append the second code (with code referring to the green channel replaced by reference to the red channel) to the first one. Thus, both channels are processed subsequently.

Due to our experience, this is the fastest and most reliable method to solve this problem. Trying to "fuse" about 140 lines of macro-code for parallel processing of both channels is much more error-prone than just attaching it for subsequent processing. The same solution is recommended, if more than 2 channels have to be processed with the same mask.

What if the code does not work in newer versions of ImageJ?

The version we used was mainly "ImageJ 1.50b". In newer version (like the self-updating Fiji), the window-naming was/may be changed.

Thus, the “Cyt/Nuc”-code pointing to different channels may be changed as mentioned **Table 1**:

> **Line 22:** The DAPI channel (in this case blue) is selected. In more recent versions of ImageJ this command may be replaced by “*selectWindow("C3-"+imgName)*” due to a new window designation, as well as in all other lines including this command (“C1-“ for the red, “C2-“ for the green and “C3-“ for the blue channel, respectively).

Summarizing the idea of the macro:

This macro is a massive timesaver for an absolute common issue of fluorescence microscopy (quantifying/investigating one or two channels compared to a third one). The code is easy to understand and consequently also easy to customize to fit other problems.

Yours sincerely,

Tobias Jung, PhD

Homepage:

www.scyrus.de

E-mail:

tobias.jung@dife.de