Quick User Guide for GUV fusion analysis

Quantification of Giant Unilamellar Vesicle Fusion Products by High-Throughput Image Analysis

Adriano Caliari [1,2], Martin M Hanczyc [2], XXX XXX [3], Jian Xu * and Tetsuya Yomo *

1: Laboratory of Biology and Information Science, School of Life Sciences, East China Normal University,
Shanghai 200062, China; adriano.caliari@hotmail.com (A.C.)

2 Laboratory for Artificial Biology, Department of Cellular, Computational and Integrative Biology (CIBIO),
University of Trento, Polo Scientifico e Tecnologico Fabio Ferrari, Polo B, Via Sommarive 9, Povo, 38123,
Trentino Alto-Adige, Italy; martin.hanczyc@unitn.it (M.M.H)

* Correspondence: xujian@sei.ecnu.edu.cn (J.X.); tetsuyayomo@gmail.com (T.Y.);
Tel.: +86-(21)62233727 (J.X. & T.Y.)

Analysing multi-point multi-channel GUVs images

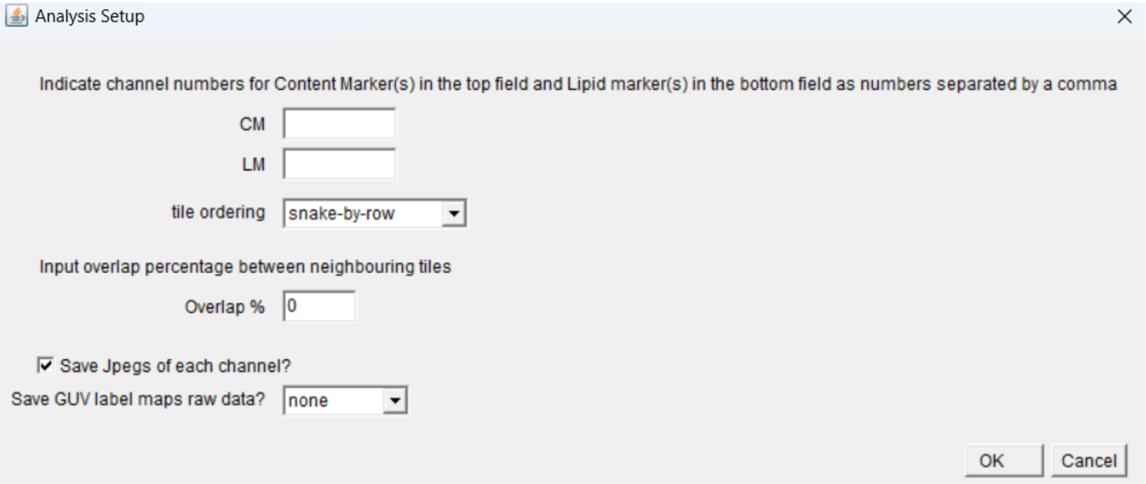
To analyse a set of GUV fusion experimental data, first the imageJ macro “GUVfusion_analysis.ijm” is used to segment and export single compartment information from the images. Then, R template scripts perform the rest of the analysis by specifying the various experimental parameters.

The ImageJ macro requires the MorphoLibJ plugin (https://imagej.net/plugins/morpholibj) and CLAHE plugin (https://imagej.net/plugins/clahe).

Files supported have the following requirements in terms of channel number or size:

- At least one channel identifying GUVs membranes is needed. The number of fluorescence channel has to be at least 1.
- Z-stacks and timelapse data are not currently supported and adaptation of imageJ and R scripts to these formats is needed to analyse them. Single slice multi point files are the assumed format.

To perform the analysis, download the files to be analysed to a folder (input folder). They must follow the naming convention DATE_SAMPLENAME_REPLICATE_ to be correctly analysed. Date format preferred is yyyymmdd. The first step of the analysis is opening the ImageJ macro (“GUVfusion_Analysis.ijm”) in the FIJI editor and running it. After selecting an input folder and an output folder, the prompt in fig. 1 should be visible.



Analysis Setup

Indicate channel numbers for Content Marker(s) in the top field and Lipid marker(s) in the bottom field as numbers separated by a comma

CM

LM

tile ordering: snake-by-row

Input overlap percentage between neighbouring tiles

Overlap %: 0

☑ Save Jpegs of each channel?

Save GUV label maps raw data?: none

Figure 1 prompt for image analysis setup

The analysis will start after providing the following parameters:

- Content markers channel numbers separated by comma.

- Lipid markers channel numbers separated by comma.
- Tile ordering to define the spatial relationship between the images organized in a 1D vector and their original position during acquisition. Four options can be selected: row by row (figure 2 top left), snake by row (figure 2 top right), column by column (figure 2 bottom left) or snake by column (figure 2 bottom right).
- If the points have no spatial relationship (random sampling or tiles of large images with no overlap) specify "0%" in the prompt upon startup of the ImageJ macro
- Flagging the "Save jpeg" box allows output of single channel compressed images for visualization purposes. If not selected, only merged images (used for GUV representation in R) and masks of detected GUVs (to inspect segmentation outcome) are saved.
- Label maps of GUVs can also be saved in uncompressed format to select specific GUV ROIs (membrane, lumen or entire GUV) in object-wise colocalization analysis in R.

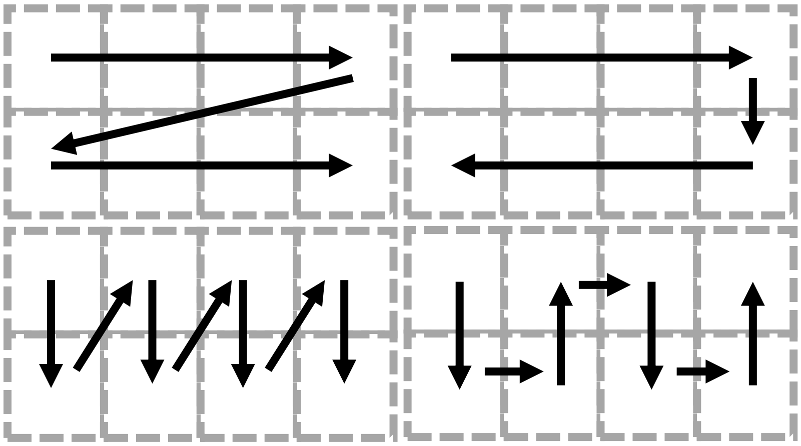


Figure 2 image reordering settings available in the macro

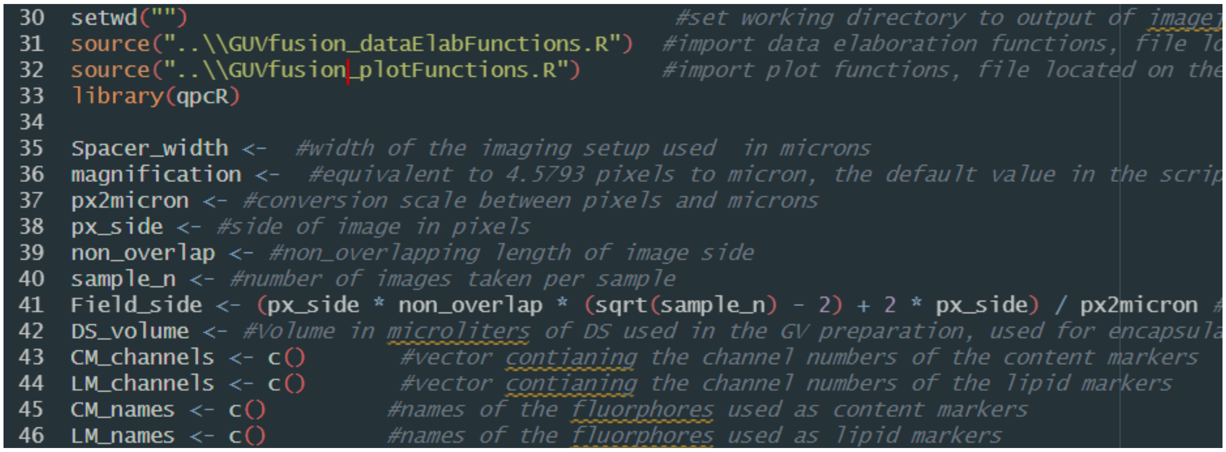
Once the appropriate settings are selected, the macro will sequentially open the files in the input folder and will create two subfolders in the output: ELAB_IMGs and RAW_Data. The former will contained all compressed and contrast adjusted images, the latter instead will contain csv and tiff data for every image in every file of the input.

For each image of the input files, one csv and tiff is saved per fluorescence channel with one csv specifying morphological properties of the GUVs. For details on the macro functions refer to section 2 and comments in the ijm file.

The raw data output is further elaborated by the R script "GUVfusion_AnalysisTemplate.R". Several preliminary operations are needed to properly set up the analysis file. Step by step instructions are available in section 3. Briefly, the template makes use of functions specified in two other R files, located by default one level above the working directory. "GUVfusion_dataElabFunctions.R" and "GUVfusion_plotFunctions.R" contain functions needed to import and format data for each image file analysed and to visualize the data in convenient plots.

The following libraries have to be installed beforehand:

qpcR; multimode; magick; imager; graphics; FField; RColorBrewer; sm; colorspace.



```
30  setwd("")                                     #set working directory to output of imagej
31  source("..\\GUVfusion_dataElabFunctions.R")  #import data elaboration functions, file lo
32  source("..\\GUVfusion_plotFunctions.R")      #import plot functions, file located on the
33  library(qpcR)
34
35  Spacer_width <-  #width of the imaging setup used  in microns
36  magnification <-  #equivalent to 4.5793 pixels to micron, the default value in the scrip
37  px2micron <- #conversion scale between pixels and microns
38  px_side <- #side of image in pixels
39  non_overlap <- #non_overlapping length of image side
40  sample_n <- #number of images taken per sample
41  Field_side <- (px_side * non_overlap * (sqrt(sample_n) - 2) + 2 * px_side) / px2micron #
42  DS_volume <- #Volume in microliters of DS used in the GV preparation, used for encapsula
43  CM_channels <- c()       #vector contianing the channel numbers of the content markers
44  LM_channels <- c()       #vector contianing the channel numbers of the lipid markers
45  CM_names <- c()         #names of the fluorphores used as content markers
46  LM_names <- c()         #names of the fluorphores used as lipid markers
```

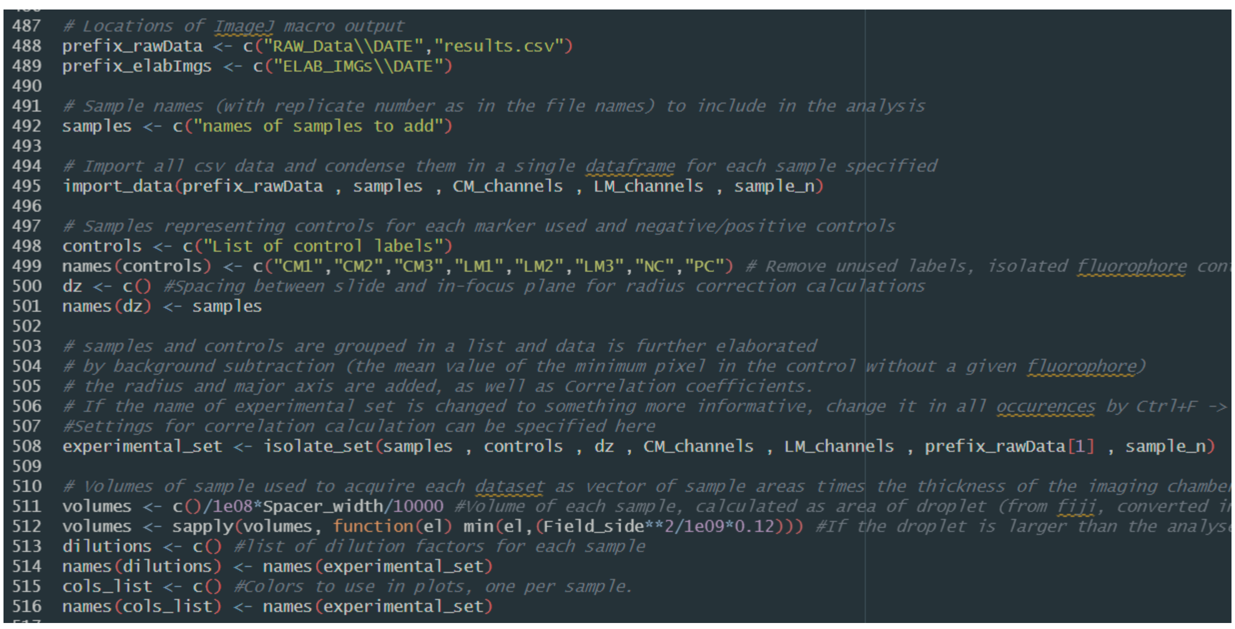
Figure 3 Header of the R analysis template script

Preliminary information on the experiment should be provided in the header of the template. The working directory should be the same output directory used in the ImageJ step. The parameters provided here are not strictly necessary for analysis completion but precise values will make the results more reliable.

The aim of the script is to provide absolute and relative GUV concentrations of various subpopulations and compare them to control samples specified in each experimental set. If more than one element is present in the CM_channels or LM_channels vectors, the template will flag the experiment as a content exchange or lipid exchange experiment respectively (the two flags are independent of each other).

After basic experimental settings, the information depicted in figure 4 should be filled in. A path to the ImageJ macro output is required (by modifying the DATE and other parts of the presets in line 488-489 (figure 3). The vector of sample names in the format “SAMPLENAME_REPLICATE” specifies which files to organize and import in the R environment.

Among the samples specified, some are needed as controls in the vector in line 498. Specifically, a control containing GUVs with isolated lipids and content markers are expected to perform background subtraction and to identify negative/positive GUV signal intensities. A negative control of GUVs from different populations but with no lipid or content exchange must be provided as well. Similarly, a positive control where all content markers or lipid markers are included is also expected to identify content/lipid exchange-positive GUVs from the rest.



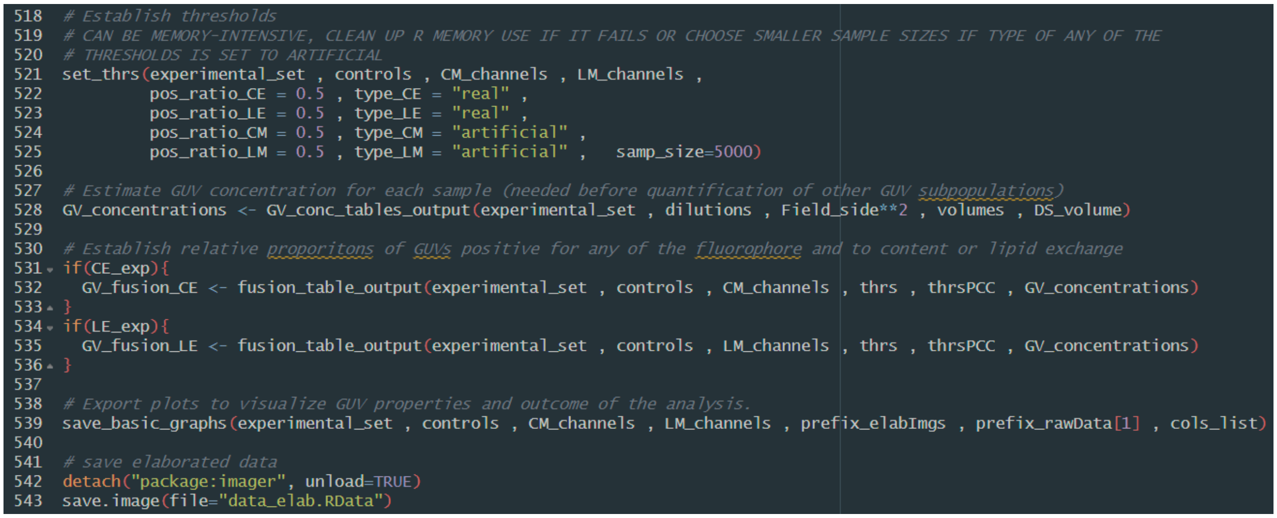
```r
# Locations of ImageJ macro output
prefix_rawData <- c("RAW_Data\\DATE","results.csv")
prefix_elabImgs <- c("ELAB_IMGs\\DATE")

# Sample names (with replicate number as in the file names) to include in the analysis
samples <- c("names of samples to add")

# Import all csv data and condense them in a single dataframe for each sample specified
import_data(prefix_rawData , samples , CM_channels , LM_channels , sample_n)

# Samples representing controls for each marker used and negative/positive controls
controls <- c("List of control labels")
names(controls) <- c("CM1","CM2","CM3","LM1","LM2","LM3","NC","PC") # Remove unused labels, isolated fluorophore con
dz <- c() #Spacing between slide and in-focus plane for radius correction calculations
names(dz) <- samples

# samples and controls are grouped in a list and data is further elaborated
# by background subtraction (the mean value of the minimum pixel in the control without a given fluorophore)
# the radius and major axis are added, as well as Correlation coefficients.
# If the name of experimental set is changed to something more informative, change it in all occurences by Ctrl+F ->
#Settings for correlation calculation can be specified here
experimental_set <- isolate_set(samples , controls , dz , CM_channels , LM_channels , prefix_rawData[1] , sample_n)

# Volumes of sample used to acquire each dataset as vector of sample areas times the thickness of the imaging chambe
volumes <- c()/1e08*Spacer_width/10000 #Volume of each sample, calculated as area of droplet (from fiji, converted i
volumes <- sapply(volumes, function(el) min(el,(Field_side**2/1e09*0.12))) #If the droplet is larger than the analys
dilutions <- c() #list of dilution factors for each sample
names(dilutions) <- names(experimental_set)
cols_list <- c() #Colors to use in plots, one per sample.
names(cols_list) <- names(experimental_set)
```

Figure 4 Further parameters to provide to the template

The z spacing between imaging plane and chamber bottom is used to correct the size of GUVs. One spacing per sample can be specified (figure 6 for schematic). With all these details, the dataset is saved as a list of elaborated dataframes by compensating GUV size, subtracting background values from mean and maximum GUV fluorescence intensities and by computing the correlation score between different markers for each GUV.

In the last step of the analysis, the information in experimental_set is summarized in tables and plots. The volumes vector is set up to contain the areas of samples droplets, adjusted by the spacer width parameter and converted to milliliters. The dilutions vector corrects the concentration estimates by the dividing by the dilution factor of each sample (1:2 dilution is specified for example by 1/3 in the vector or 0.333). cols_list specifies one color for each sample to use in the plots.

Finally, the scrip elaborates the data through the steps represented in Figure 5. For each fluorophore specified in the channel vectors, GUVs positive to that fluorophore are automatically identified based on fluorescence histogram thresholding. For each distinct pairs of channels, the corresponding correlation score of fluorescence signals is added to each GUV. This parameter is thresholded as well to classify GUVs positive for lipid or content exchange. Based on the geometry of the sample (spacer width, sample area ecc.) concentration estimates for GUVs are produced (# of GUVs /volume of sample) and compared to the negative control sample to evaluate vesicle loss upon treatment. GUV fluorescence is also compared between samples and controls to evaluate fluorescence loss due to quenching or leakage, by the difference of normalized mean fluorescence intensity of each GUV.



```r
# Establish thresholds
# CAN BE MEMORY-INTENSIVE, CLEAN UP R MEMORY USE IF IT FAILS OR CHOOSE SMALLER SAMPLE SIZES IF TYPE OF ANY OF THE
# THRESHOLDS IS SET TO ARTIFICIAL
set_thrs(experimental_set , controls , CM_channels , LM_channels ,
         pos_ratio_CE = 0.5 , type_CE = "real" ,
         pos_ratio_LE = 0.5 , type_LE = "real" ,
         pos_ratio_CM = 0.5 , type_CM = "artificial" ,
         pos_ratio_LM = 0.5 , type_LM = "artificial" ,   samp_size=5000)

# Estimate GUV concentration for each sample (needed before quantification of other GUV subpopulations)
GV_concentrations <- GV_conc_tables_output(experimental_set , dilutions , Field_side**2 , volumes , DS_volume)

# Establish relative proporitons of GUVs positive for any of the fluorophore and to content or lipid exchange
if(CE_exp){
  GV_fusion_CE <- fusion_table_output(experimental_set , controls , CM_channels , thrs , thrsPCC , GV_concentrations)
}
if(LE_exp){
  GV_fusion_LE <- fusion_table_output(experimental_set , controls , LM_channels , thrs , thrsPCC , GV_concentrations)
}

# Export plots to visualize GUV properties and outcome of the analysis.
save_basic_graphs(experimental_set , controls , CM_channels , LM_channels , prefix_elabImgs , prefix_rawData[1] , cols_list)

# save elaborated data
detach("package:imager", unload=TRUE)
save.image(file="data_elab.RData")
```

Figure 5 final step of the R analysis

This data is summarized in tables by fusion_table_output and GV_conc_tables_output. Save_basic_graphs makes use of the plotting functions specified in the companion script to save useful information for each sample specified. Fluorescence histogram for every channel, dot plots for pairs of channels, size distribution, histograms of correlation score and color coded correlation-fluorescence dot plots are saved. The quality of fluorescence thresholding can be inspected in overlayed histograms of the control samples for each fluorescence channel. Violin plots for quick comparison of distribution of all these parameters are also available. Two GUV mosaics per sample are saved as well in the case of content or lipid exchange experiments. The FP panel contains the portion of positive GUVs with the lowest correlation scores that are excluded from analysis since considered false positives. TP panel has a random sample drawn from the rest of the positive population, used to visualize the quality of classification. The objects saved in the TP panel are visualized also on the color coded dot plots by tags specifying GUV ID# and image#.

Further elaboration is possible by using the plot functions in specific ways, details for which are provided in section 3 and comments within the code.

For any further clarification, bug reports and discussion feel free to contact adriano.caliari-1@unitn.it or the corresponding authors of the publication.

2. Description of the ImageJ macro

To segment compartments from micrographs, an ImageJ macro was developed using the ImageJ macro language. The large number of pre-built functions and packages accessible by this language were employed to design an automatic thresholding and labelling system that could analyze fluorescence images of Giant Unilamellar Vesicles (GUVs) without user input and regardless of the amount of objects contained in each frame. User input is required upon running the script, to provide the following information:

- An input folder, where the multi-channel multi-point images are saved. We used Nikon's nd2 format, but any other format compatible with ImageJ's Bio-Formats Importer would work as well. A naming convention for the files contained in the input folder is assumed, in the form "DATE_SAMPLENAME_REPLICATE_.nd2".
- An output folder, where the processed files are going to be saved. Two subfolders are created in the output folder. "ELAB_IMGs" is used to save contrast adjusted images as compressed jpg format. Single channels and label maps are saved to inspect image quality and segmentation outcome in the various files. A merged image of the various single channels is also saved and used for visualization purposes in later steps of the analysis. "RAW_Data" will contain GUV information in csv format, tiff files for every channel analyzed and tiff versions of the labelled GUV masks.
- The channels contained in the images, subdivided as content markers or lipid markers, identified by their channel number.
- An overlap percent between neighbouring tiles in case this is set during acquisition. Images are cropped according to their placement in the area scanned. For example, they can be arranged in a snake-by-rows fashion. The overlap issue can be ignored by setting the value to 0, making image indexing unimportant. If a crop percentage above 0 is provided, images will be cropped on appropriate sides by half of the amount indicated, removing overlapping areas from the dataset.

Images in the input folder are opened sequentially in batch mode (by default). Frames are assumed to be arranged in a square. After cropping, contrast adjusted jpgs are saved for all frames. The merged image is generated with up to six channels by automatically assigning lipid markers and content markers to LUTs in the following order:

- LE: 1 – Magenta; 2 – Green; 3 – Blue.
- CE: 1 – Cyan; 2 – Yellow; 3 – Red.

After this, a filtering sequence is applied to each lipid channel to segment GUVs stained with that specific probe. The following steps are applied in sequence:

- ImageJ built in Subtract Background function is used to apply rolling ball background subtraction using a radius of 10 pixels.

- Contrast Limited Adaptive Histogram Equalization (CLAHE) [54] from the homonym plugin is used to enhance contrast between membranes and background signal. Block size was set at 99, histogram bins at 256 and maximum slope at 2.0.

- The Unsharp Mask Filter is applied to the contrasted image to enhance contrast. It subtracts a weighted version of the image blurred by a gaussian blur filter, with the radius of the gaussian filter 1.5 pixels and the weighting factor 0.9.

- Noise in the image is partially suppressed using a gaussian blur filter with standard deviation 0.5 pixels.

The resulting image has contrast-enhanced and sharpened membrane signals and is used to form a binary mask of membrane regions through automatic thresholding. It is converted to 8-bit format and thresholded using Phansalkar's method [55] with the following settings: radius (5), parameter1 (0.25), parameter2 (3). The MorphoLibJ plugin is used to obtain masks of GUV lumens [56]. In the first step, membrane regions touching the image border are removed with the "kill borders" function, excluding from the analysis any GUV that was not completely imaged. Regions bound by membranes are included by a flood fill operation via the "fill holes" function. The difference between the membrane mask without GUVs on borders and the filled in mask provides a content-selective mask where random noise is removed as well. A second clean-up step to remove artefacts that arise by noise amplification in empty images is applied using the Shape Filter plugin [57]. Particle area is filtered between 1 to 500 μm^2, and solidity is limited between 0.85 and 1. If multiple lipid markers are specified, these operations are performed for each one, and the single channel masks are combined with an OR operation, obtaining a final mask containing information from every lipid channel.

The connected component labelling function included in MorphoLibJ is used to assign a univocal label to each compartment detected using 4-way connectivity. By applying a 2-pixel wide dilation to these labels, an inclusive label map is produced, that contains both GUV lumen and membrane. The difference between the initial mask and the dilated one yields 2-pixel wide rings around the vesicles' lumen, which are used as membrane label map. These label maps are used to define membrane fluorescence information, using the membrane label map for each membrane channel, or morphology and lumen fluorescence information using the lumen label map for each content marker. All these values are saved as csv files in the "RAW_Data" folder. Fluorescence information includes mean pixel intensity, standard deviation, maximum, minimum, median, mode, skewness and kurtosis. Morphology of the object is saved as perimeter, area, circularity, centroid X and Y coordinate, fitted ellipse X and Y centre coordinates, minor and major axis, ellipse orientation, elongation, convex area and convexity of the shape (similar to solidity). Values for all parameters used in the filtering steps were optimised with a version of the macro that accepted test values for each parameter as vectors (one per parameter) and performed the filtering and detection sequence on a set of test images. The parameter values were tested sequentially, choosing the value that yielded the highest number of detected objects each time. A version of the optimization routine that checks for each combination of values was also implemented but the run time to test all combinations of parameters was not practical, so it was used to do a small local search, highlighting minimal effect of different combinations of parameters.

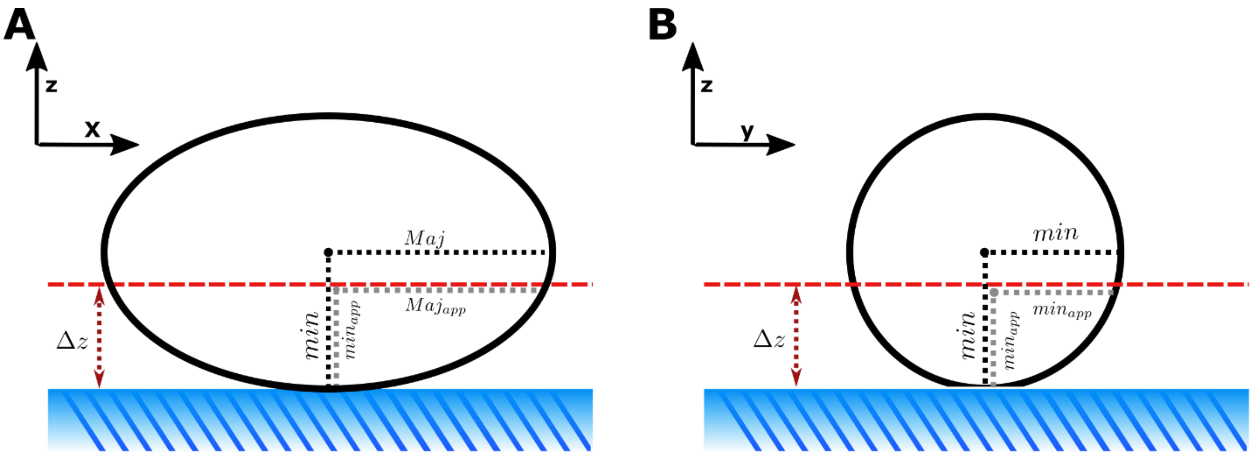
3. Description of the R scripts

Further data elaboration and statistical treatment are carried out using a template R script that requires input of experimental parameters to summarize the results. Specifically, the following information has to be provided in the header of the template to have a complete analysis:

- Working directory (the same as ImageJ output).
- Thickness of the imaging chamber.
- Objective magnification and pixel to micron conversion factor.
- Image pixel width. Images are assumed to have 1:1 aspect ratio.
- Fraction of non-overlapping pixels as set during acquisition.
- Number of the images per sample, for which square arrangement is assumed.
- Volume of DS used in sample preparation.
- List of content markers and lipid markers channels as numeric vectors, indicating the number of the channel in the original image. The elements of these vectors can be named with the type of fluorophore used to correctly label axes in plots produced by the analysis routine.
- path to raw data and elaborated images in a vector containing “RAW_Data\\DATE” and “results.csv” as elements for the raw data folder and “ELAB_IMGs\\DATE” for elaborated images location
- A vector containing the samples names in the format “SAMPLENAME_REPLICATE” that identifies files related to a given sample.
- A vector that specifies which samples to use as controls. The analysis routine expects to find one control for each content marker or lipid marker (if the experiment is content exchange or lipid exchange respectively), a negative control where the fluorophores are in separate populations and a positive control where the fluorophores are present in all vesicles.
- A vector specifying the areas of sample droplets in the imaging chamber. For maximum accuracy these were obtained from low-magnification map of the whole imaging chamber in ImageJ, by thresholding the edge of the sample droplet and filling in the whole area with the fill holes function. The areas thus obtained in μm^2 are multiplied by the spacer width and converted to milliliters.
- A vector containing the z offset used in collecting each sample for size compensation purposes.

- A vector containing the dilution factors used in preparing the samples, to correct the volume of sample calculated by this factor and compare actual concentrations of the samples.

Data in csv format is for each sample specified are imported and combined in a single dataset for each element of the samples vector. These datasets are further elaborated by constructing a list containing the datasets and incorporating information from the control samples specified. A background value for each marker is estimated as the average minimum pixel intensity for GUVs of controls that do not include the marker. This value is subtracted from the mean and max fluorescence signal of all datasets. Besides adjustments to fluorescence values, size compensation is performed. GUVs are modelled as ellipsoids with rotational symmetry around their minor axis. This geometrical assumption is used to compensate the minor and major axis according to the known axial offset. The compartment volume is calculated from these compensated dimensions and the radius is taken as the radius of a sphere of equivalent volume. The shape used here loses validity in deformed GUVs (oblate, prolate, stomatocyte or dumbbell) but should be a good approximation for spherical or near spherical GUVs. A schematic view of the size adjustment assumptions is presented in Figure S11. Finally, correlation data is added for each pairwise combination of channels of the same type (lipid or content markers). For each object in the various datasets, the tiff images of its fluorescence channels are imported. A square region of interest centered on the GUV (fitted ellipse coordinates) is used to define the region on which correlation is computed. Pixel values for the two channels considered are converted to numeric vectors and correlation is computed by the standard correlation test routine in R. This implements correlation estimates by Pearson Correlation Coefficient (PCC), Kendall's coefficient or Spearman's coefficient (the latter two are non-parametric). Alternatively, Mander's colocalization coefficient (MCC) was implemented in two variants. The first uses the pixel median value as threshold, the second uses instead the minimum pixel value in the GUV area. Masks can be used to select areas of the region of interest based on the label value of the GUV of interest. With this method, pixels relative to GUV membrane, content or both areas can be isolated from the rest of the region of interest. Thresholding of fluorescence and correlation histograms is used to define positive GUVs to markers or correlation by using the multimode R package to find the antimode of the histograms. After thresholding, GUV concentrations and relative population percentages are calculated. These functions are contained in the "data_elab_functions.R" script that is imported in the header of the template. Data visualization functions are implemented in "plot_functions.R", allowing rapid visualization of GUV properties via histograms, dot plots, violin plots and GUV image composites to visualize the objects classified in the various populations.



$$Maj = \frac{Maj_{app} \times min}{\sqrt{2 \times min \times \Delta z - \Delta z^2}}$$

$$min = \frac{min_{app}^2 + \Delta z^2}{2 \times \Delta z}$$

Figure 6 Schematic of the size compensation of GUVs assuming an ellipsoidal shape