**MATLAB Microbubble Population Analysis User Guide**

Damien Batchelor, Molecular and Nanoscale Physics, University of Leeds, June 2020

**Introduction**

This script is designed to determine concentration and size of microbubbles (MBs) using a series of brightfield microscopy images. This script can also be utilised for detection of other spherical particles; however, it is not optimised for this purpose. The script has been designed with a user-friendly interface such that those without programming experience can run and use the script.

**Disclaimer: This work is “citeware” under an MIT licence, and as such should be cited in any published research that the script has contributed towards using the associated DOI.**

**Software Requirements**

This script requires the MATLAB Image Processing Toolbox (https://www.mathworks.com/products/image.html) and the latest version of MATLAB is recommended (2019b at the time of writing). Additionally, one of either the Financial Toolbox or Statistics and Machine Learning Toolbox are required. Compatibility with older versions is not guaranteed and may be prone to syntax errors, however support is available via the contact details in the Troubleshooting section.

**Experimental Procedure**

To accurately determine concentration, brightfield images of MBs should be obtained in an imaging chamber with known depth such as a haemocytometer or using polymer spacers and a cover slip. The **depth of the chamber**, in combination with **image scaling**, allows for calculation of total image volume and therefore MB concentration. More details on the experiment protocol can be found in previous publications [1–3]. The concentration of MBs during imaging should be optimized such that there is minimal interaction/overlap between MBs. From previous experience this is approximately 108 MBs/mL for a chamber depth of 50 μm. The minimum detectable diameter of MBs will in turn be determined by the resolution of the camera.

**Analysis Procedure**

Prior to running the analysis script, all images to be analysed (e.g. one sample) are required to be in **.jpeg format** and contained in a separate folder. All images in this folder will be analysed as one sample and raw, processed and meta data for the analysis will be saved in this folder after completion of the script.

**Step by Step Analysis Protocol**

**1. Initialising the Script**

* Open the .m MATLAB script file in MATLAB.
* Click the green “Run” button in the Editor tab to start the script.
* The user will be prompted to locate and select the folder for analysis. Once you have selected the folder, press “Select Folder”. All .jpg images in this folder and sub-folders will be analysed as one dataset.
* After this, the user will be prompted to enter a series of values relating to experimental conditions. These are:
  + **Image Scaling** (μm/pixel) – To determine MB size in μm.
  + **Sample Dilution** (eg. 10) – To account for sample dilution in concentration measurements.
  + **Histogram Bin Width** (μm) – The width of bins for histogram plots.
  + **Spacer Height** (μm) – Depth of the imaging chamber to accurately determine MB concentration.

**2. Binarization Threshold**

* The user will then be prompted to determine the binarization threshold. In this process a preview binary (black and white) image will be shown. A slider bar at the bottom will vary this threshold and auto-update the displayed image (Figure 1).

|  |
| --- |
| ***Figure 1*** *– Binarization Threshold selection screen.* ***Left:*** *Binary Image at current binarization threshold value.* ***Right****: Initial Image. The current value for the threshold is determined using the slide bar at the bottom of the window, with the current value displayed to the left. Once the value has been optimised, clicking the rectangular box will confirm the value.* |

* This value should be varied until the threshold is optimised i.e. all bubbles are successfully detected and filled whilst there is no presence of “speckle” noise patterns (Figure 2).
* The optimised value can be confirmed by click in the rectangular box to the left of the slider, which contains the chosen threshold value.

|  |
| --- |
| ***Figure 2*** *– Examples for binarization threshold values that are:* ***Too Low****: MBs not properly detected,* ***Optimal*** *: All MBs detected without any noise and* ***Too High*** *: Presence of speckle noise.* |

**3. Circle Sensitivity**

* A similar process is then followed for determining the sensitivity of the circle detection. This value should be adjusted such that all (or the majority) of MBs are detected but so that false positive MBs are not found (Figure 3).

|  |
| --- |
| ***Figure 3*** *– Example of circle sensitivity values that are:* ***Too Low*** *– MBs not detected,* ***Optimal****: All MBs detected,* ***Too High****: False detections.* |

* Again, the optimised value is confirmed by clicking the rectangular button to the left of the slider.

**4. Image Processing**

* After the previous step, image processing and analysis will run across all images in the sequence.
* By default, the first 3 images in the sequence will appear with the locations of detected MBs shown by a red circle.
* The progress of the analysis is shown in the MATLAB Command Window eg. “Now Reading 8 of 20”.
* The total run time of the script will depend on various factors such as processing power, RAM, number of images and number of MBs per image. However, a ballpark figure for run time is between 30 seconds to a few minutes.

**5. Image Analysis**

* Once the analysis is completed, a bar chat plot of the MB size distribution is shown (Figure 4).
* This figure also shows additional parameters: mean concentration and diameter, relative standard deviation (RSD), the proportion of MBs < 8 and 10 μm and maximum diameter. All errors given are standard deviation from the mean. RSD is the ratio of the standard deviation to the mean, expressed as a percentage.
* This figure is saved into the same folder as the image as both a ‘.tif’ image file and MATLAB figure ‘.fig’ file.

|  |
| --- |
| ***Figure 4*** *- An example MB distribution with a bin width of 0.25 μm. Other parameters such as concentration and mean diameter as shown inset.* |

**6. Raw Data**

* The provided MATLAB histogram is an excellent way to quickly and efficiently determine the size and concentration of your MB distribution. However, for more in-depth analysis, we recommend importing the raw and processed data into other software suites (e.g. Origin Pro).
* The processed analysis data is saved to various files, situated in the same folder as the initial images. These are as follows:
  + Concentration.csv – A .csv file containing the bin positions and corresponding concentrations.
  + Diameter.txt – A .txt text file containing a list of diameters of all detected MBs from analysis. This allows the user to re-plot the population distribution with their desired bin widths.
  + Log.txt – A log file that contains meta data values such as binarization threshold and circle detection sensitivity, as well as the parameter displayed in the population plot.
  + Workspace.mat – A MATLAB workspace file containing the workspace of the script.

**7. Adjustments to the Script**

For those who are more familiar or comfortable with MATLAB and/or other programming languages, small adjustments to the script can be made to improve quality of life.

**Changing Import File Type**

* Currently imported images need to be .jpg format. However this can be changed in the findfolder() function. Replacing .jpg with .tif for example will now read in .tif files.
* However when doing this, the user needs to take care when running the analysis multiple times on the same dataset, as the output figure images are saved as .tif and as such the analysis will run over these too.
* The output format of figure images can be changed in the Plot Bar Chart section in a similar manner.

**Image Previews**

* By default the script displays the first 3 images of the sequence and the positions of detected bubbles.
* The number if images shown can be changed in the main body of the script by adjusting the if statement. Eg. k < 4 shows the first 3 images.

**Default Values**

* At the beginning of the script, the user is prompted to input various constants used throughout the processing (e.g. image scale).
* To save time, the default values can be changed in the calibration() function.
* For example, defaultans = {‘0.16’}; corresponds to the default image scaling as 0.16 um/px.
* This can be repeated for the remaining constants.

**8. Troubleshooting**

If you experience any problems with the script, please feel free to get in touch via email : [py13db@leeds.ac.uk](mailto:py13db@leeds.ac.uk) /microbubbles@leeds.ac.uk. Please include a description of your problem, and a copy of any errors present as shown in your command window.

**9. Bibliography**

1. Peyman, S.A.; Abou-Saleh, R.H.; McLaughlan, J.R.; Ingram, N.; Johnson, B.R.G.; Critchley, K.; Freear, S.; Evans, J.A.; Markham, A.F.; Coletta, P.L.; et al. Expanding 3D geometry for enhanced on-chip microbubble production and single step formation of liposome modified microbubbles. *Lab Chip* **2012**, *12*, 4544, doi:10.1039/c2lc40634a.

2. Abou-Saleh, R.H.; Swain, M.; Evans, S.D.; Thomson, N.H. Poly(ethylene glycol) lipid-shelled microbubbles: Abundance, stability, and mechanical properties. *Langmuir* **2014**, *30*, doi:10.1021/la404804u.

3. Bourn, M.D.; Batchelor, D.V.B.; Ingram, N.; McLaughlan, J.R.; Coletta, P.L.; Evans, S.D.; Peyman, S.A. High-throughput microfluidics for evaluating microbubble enhanced delivery of Cancer therapeutics in spheroid cultures. *J. Control. Release* **2020**, doi:https://doi.org/10.1016/j.jconrel.2020.06.011.