

MicroBREW

Authors: Bishal Paudel, Oshil Ghimire

Image processing pipeline:

To quantify replicative lifespan (RLS) using a high-content imaging platform, we developed a custom image processing pipeline in MATLAB. The analysis was performed in two main steps. In the first step, each image frame was processed to detect and record the positions of individual microfluidic traps. Briefly, edge detection (“*edge*”, default parameters) was applied to identify trap boundaries, and the resulting images were converted to binary format. Morphological operations, including dilation and hole filling (“*imfill*”, default settings), were used to refine object structures. Labeled binary images were then generated using “*bwlabel*” to assign unique identifiers to each detected object. A size threshold filter was applied to exclude large artifacts, such as those arising from elongated budded cells. For each identified trap, spatial coordinates (x, y) were extracted and used to track the trap location consistently across all frames in the time-lapse movie.

In the second step, four regions of interest (ROIs) were defined for each trap based on the spatial coordinates. These included one central circular region to track the entry and position of the mother cell, and three adjacent kite-shaped regions to capture signals associated with budding events. Within each of the four ROIs, background was corrected by subtracting the mode of the signal intensity distribution, assigning zeros if the values resulted in negative. The background-corrected pixel intensities were then aggregated and mapped to the corresponding trap ID and frame number for downstream analysis.

Signal processing for RLS:

Signals generated in MATLAB were further processed in R to identify mother cell entry, cell death, and budding events for each trap. For each trap, the signal from the central ROI was first smoothed using a simple moving average and normalized to the pixel intensity in the first frame of the time series. Peaks in the normalized signal were detected using the ‘*findpeaks*’ function with a stringent threshold to reduce false positives due to noise. The first identified peak marked the time of mother cell entry, and the corresponding frame ID was recorded. To detect budding events, the signals from the three kite-shaped ROIs were smoothed (*post-entry*) using a moving average, and peaks were identified using ‘*findpeaks*’, with a minimum peak distance of six frames to reflect the expected time between budding events. Mother cell death was inferred from the central ROI by calculating the first-order lag difference; a sustained drop in the signal was used to indicate cell death. For each trap, the number of budding events detected between the mother cell entry and death was recorded per ROI. Additionally, the start and end points of each peak were captured to eliminate overlapping events, and peak heights were recorded to estimate the magnitude (size) of each budding event.

Application to example movies S2 and S3

To show that this pipeline also works on small, single-trap datasets, we applied it to two 34-frame time-lapse movies, Movie S2 and Movie S3. Each movie contains one main microfluidic trap. For every frame, we used the same edge-detection and morphology steps described above to find the trap, selected the largest connected component as the trap, and recorded its centroid and area.

For each movie, we then defined a single polygonal region of interest (ROI) around the trap based on the centroid. Within this ROI, we subtracted a single background value (the mode of the pixel intensities), set any negative values to zero, and summed the remaining pixel values to obtain one signal per frame. We saved the frame number, ROI signal, trap area, and centroid coordinates to CSV files and processed them in R using the same moving-average smoothing and findpeaks-based event detection described above. For Movie S3, we also divided the signal by the trap area before smoothing and peak calling to correct for small changes in the segmented trap size. These single-trap examples (Movies S2 and S3), implemented in the accompanying MATLAB and R code in the MicroBREW_Generalization GitHub repository, demonstrate that the MicroBREW pipeline extends naturally from high-content, multi-trap experiments to focused analyses of individual traps.

Movie Source: [A Microfluidic System for Studying Ageing and Dynamic Single-Cell Responses in Budding Yeast | PLOS One](#)
Github Repo: [yyc2wf/MicroBREW_Generalization](#)