A step-by-step instruction for use is as below:

1. Open the file ‘multidroplet.m’ and give an object name for multidroplets by calling the function: ‘objname = multidroplets’. Select the tif stack file to be analyzed from the pop-up window. After the selection, the saving path of the file and the number of the frames are stored in the variables ‘foldername’ and ‘NOF’.
2. Use the function ‘objname = optimizeimprocparams(objname)’ to find suitable combinations of image processing parameters. For example, choose to invert the contrast and binarize the image by answering ‘y’ and answering ‘n’ for binarizing the image and applying gaussian blur. Repeat this process to save multiple sets of parameters combinations, which are prepared for particle detection. Resultant datasets are saved in ‘improcparams’.
3. Call the function ‘objname = optimizefindparams(objname)’ to optimize the particle detection parameters for each data set processed in (2). Choose Hough transform or gaussian peak finding to detect the particles. Hough transform can identify circular shapes of the objects based on the input diameter range, corresponding to the ‘lower bound’ and ‘upper bound’ in the code with unit of pixel. The sensitivity parameter of the Hough transform, in the range of 0 to 1, refers to the threshold of number of edge points that must support a circle candidate. Gaussian peak finding identifies peaks in a distribution of intensity values within an image. Adjust the threshold of the intensity to locate droplet positions. Processed data are saved in ‘findparams’.
4. Process image & detect particles by using the function ‘objname = find\_pos(objname)’. Repeat for each data set. Droplets positions for each set are saved in cells of ‘pos\_all’. Each row shows (x, y) positions of all droplets in one frame and the first row stands for the first frame.
5. Remove duplicate particle detections by calling the function ‘objname = remove\_overlap\_pos(objname, distance)’. Define a second input as a minimum interval distance, where any duplicated positions will be removed within this distance. One set of positions of all droplets is saved in ‘pos\_clean’ with each row representing one frame. Results can be visualized by the function ‘plot\_positions\_all(objname)’.
6. Implement the function ‘objname = split\_types(objname,threshold,R)’ to split positions of two kinds of droplets by particle type based on the input threshold grayscale. The input R defines the radius of the area for calculating the intensity of the droplet. Positions of each type of droplets are arranged in an order of frames in the variable ‘pos\_split’ and can be shown in image by the function ‘plot\_positions\_clean(obj)’.
7. Track the particles by using the function ‘objname = track\_obj\_split(objname)’ for each type of droplet. The max displacement in tracking defines the displacement of a droplet over a continuous frame. This number should be at least smaller than the distance between droplets to prevent spurious tracking. The number of frames a particle can go missing describes the tolerant frames for a droplet can go missing before being retracked. In our work we set this value as 0 so all tracking trajectories of droplets appear as continuous frames. The minimum number of frames for a track to be accepted refers to the minimum length of droplet trajectories in unit of frames. For example, if we set the minimum number of frames for a track to be accepted as 100, droplets appear for less than 100 frames will be ignored. The tracking results are saved in a 12 cell in ‘tr\_split’. Positions of each type of droplets are displayed in an array with four columns: [x position, y position, number order of the frame, number order of droplets]. For example, [9, 167, 3, 7] represents that the position of a droplet in the third frame is x = 9 and y = 167, and this droplet is named as the seventh droplet. This file will be exported and serve as the raw data for neighbor analysis and motility calculations.