Supplemental Materials

A Biomimetic High Throughput Model of Cancer Cell Spheroid Dissemination onto Aligned Fibrillar Collagen

Hossam Ibrahim[1,2], Stephen D. Thorpe[2,3], Michael Paukshto[4], Tatiana S. Zaitseva[4], Wolfgang Moritz[5], and Brian J. Rodriguez[1,2]

[1]School of Physics, University College Dublin, Belfield, Dublin 4, D04 V1W8, Ireland

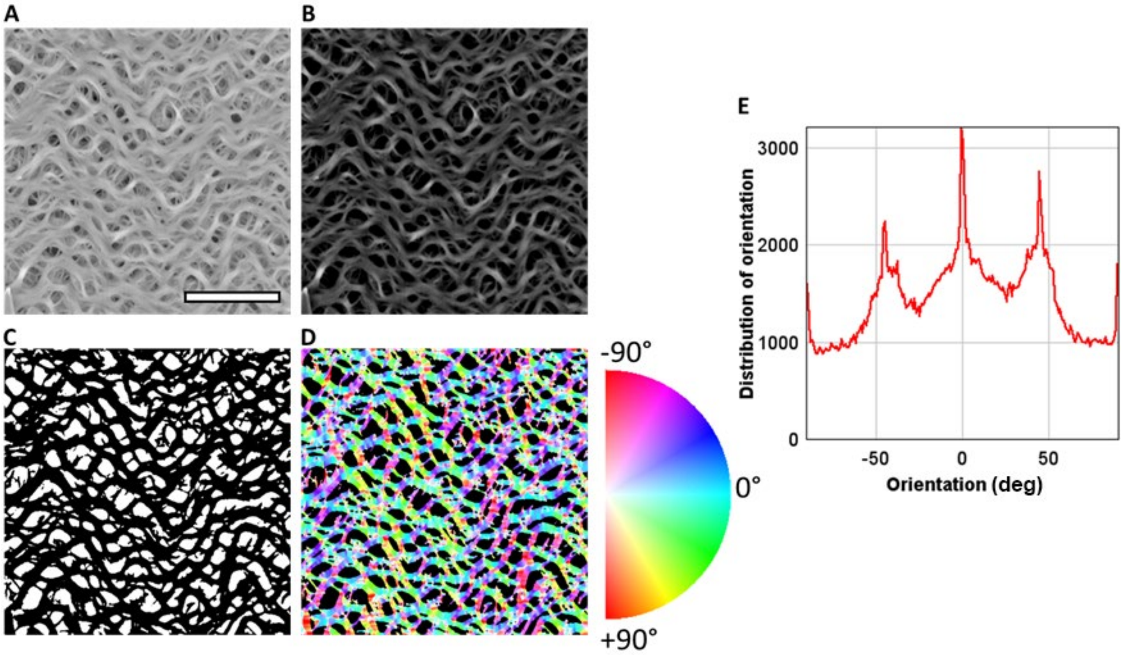
[2]Conway Institute of Biomolecular & Biomedical Research, University College Dublin, Belfield, Dublin 4, D04 V1W8, Ireland

[3]UCD School of Medicine, University College Dublin, Belfield, Dublin 4, D04 V1W8, Ireland

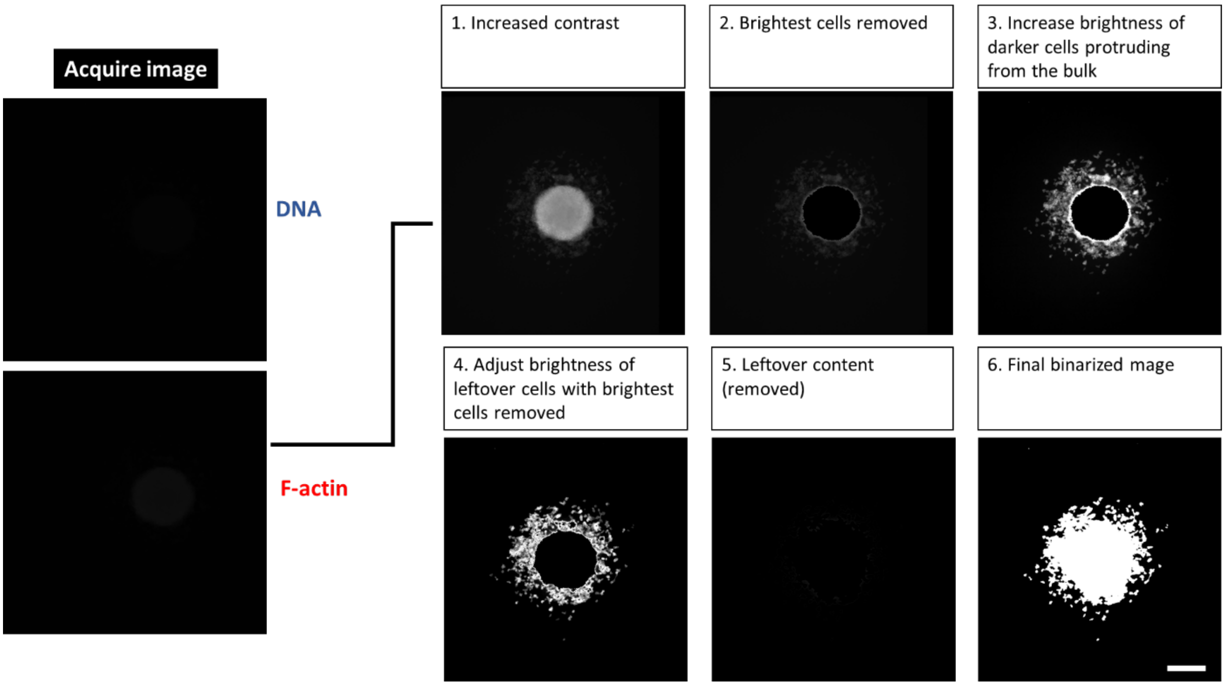
[4]Fibralign Corporation, 32930 Alvarado-Niles Rd Suite 350, Union City, CA 94587, USA.

[5]InSphero AG, Wagistrasse 27, 8952 Schlieren, Switzerland

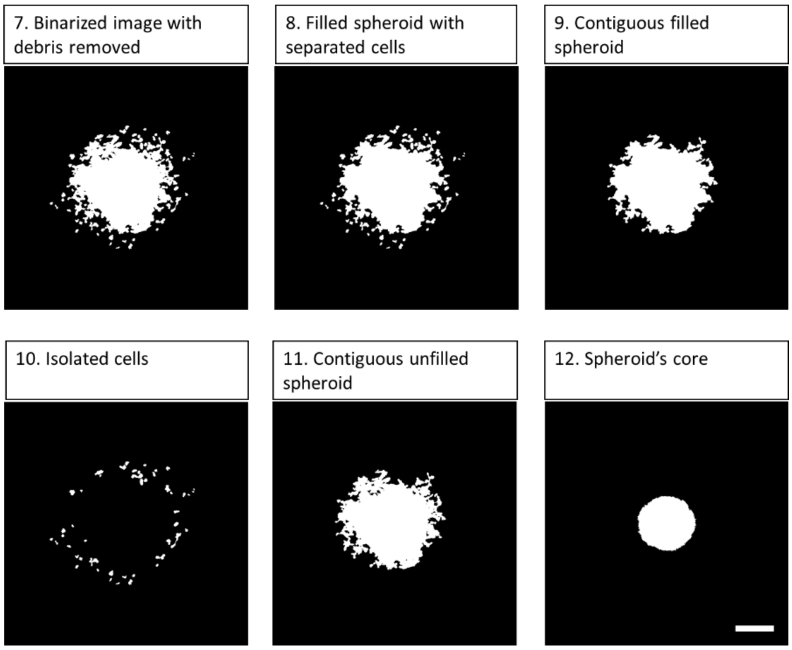
Contents



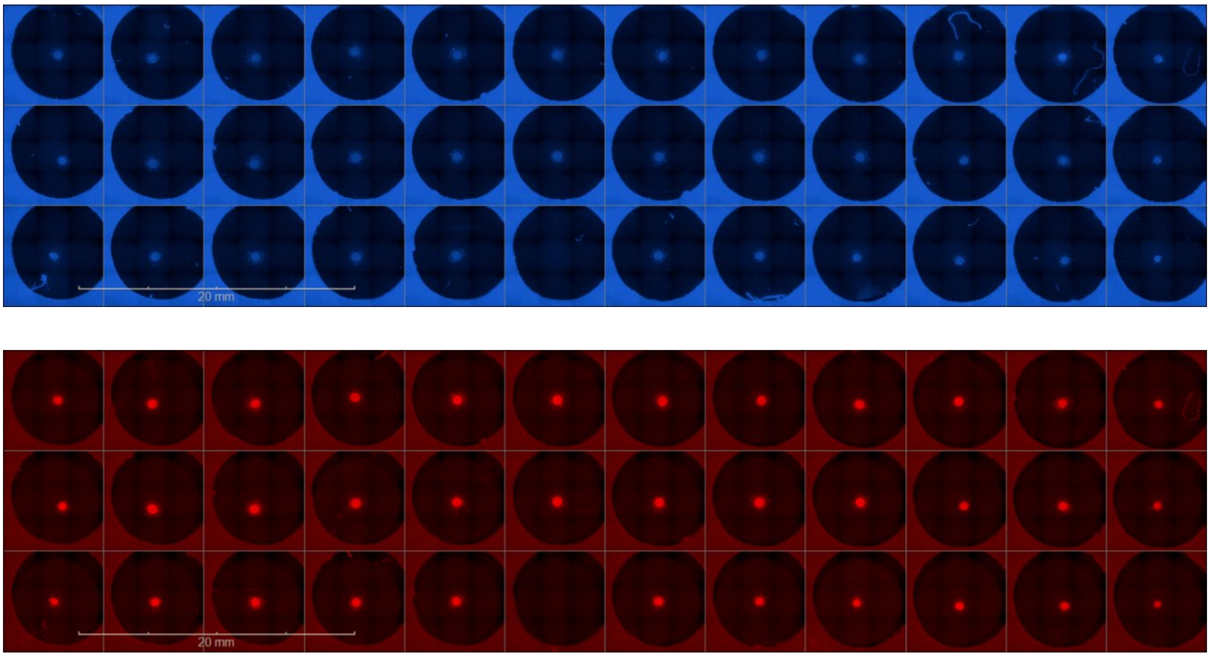
Supplemental Figure S1. Orientation of the fibrillar collagen matrix measured in Fiji using OrientationJ plugin. A: AFM height image (30 µm × 30 µm). Scale bar: 10 µm. B: Height image with brightness adjusted. C: Binarized image. D: Hue, saturation, and brightness color coded map and orientation plot; hue: local orientation, saturation: coherency, brightness: original image. E: Distribution of pixel orientation. Four peaks were detected at -44.5°, -0.5°, +44.5°, and +/-90°.


Acquire image
DNA
F-actin
1. Increased contrast
2. Brightest cells removed
3. Increase brightness of darker cells protruding from the bulk
4. Adjust brightness of leftover cells with brightest cells removed
5. Leftover content (removed)
6. Final binarized mage

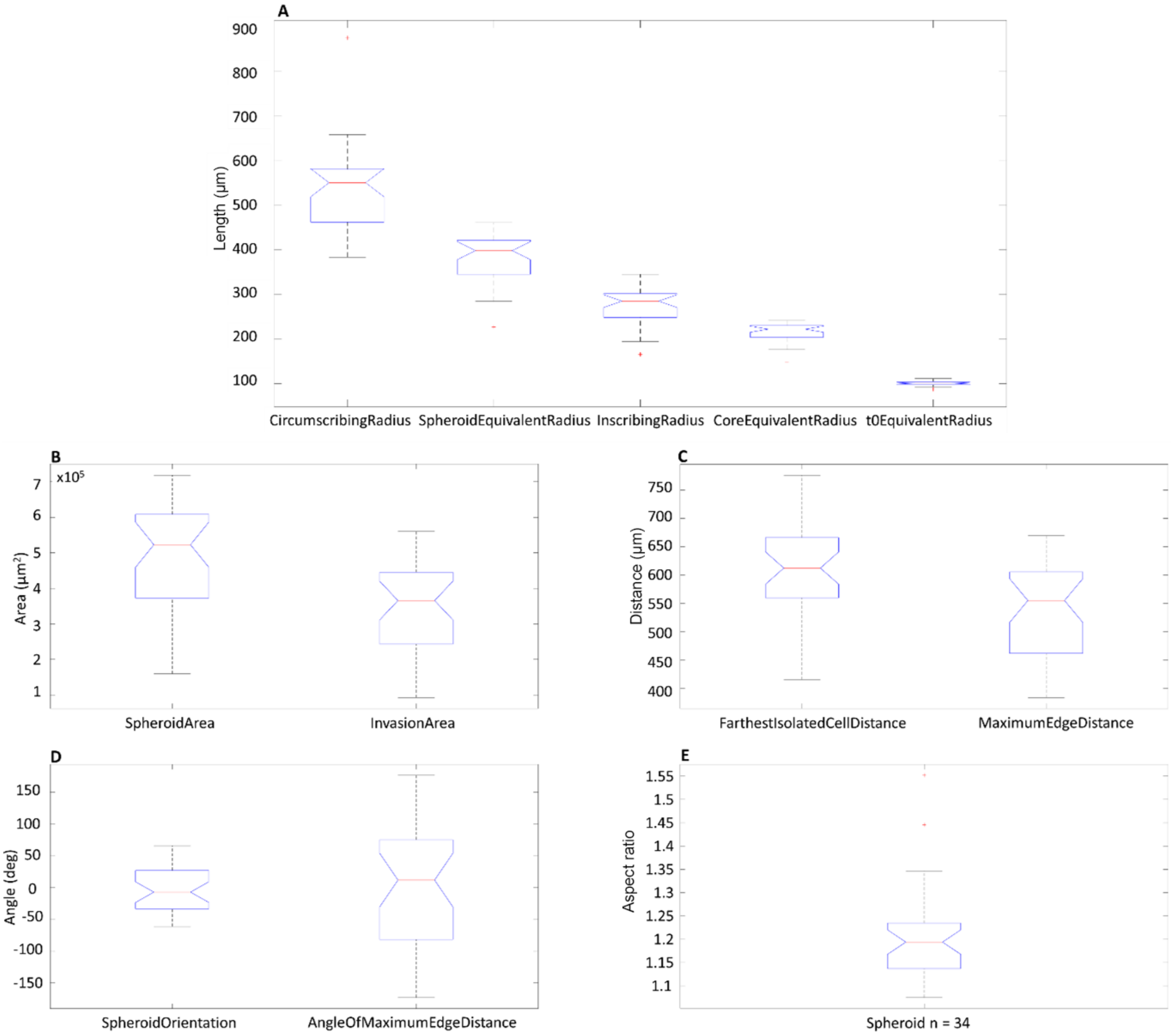
Supplemental Figure S2. Image processing pipeline for the enhancement and segmentation of greyscale F-actin images. Images were acquired at 16-bit and transformed to 8-bit before further processing. To capture all disseminated cells regardless of F-actin staining intensity, three rounds of brightness adjustment and thresholding were applied to generate the final binarized image. Any image content that failed to meet the threshold criteria on the third round was discarded (leftover content, step 5). Scale bar: 300 µm.


7. Binarized image with debris removed
8. Filled spheroid with separated cells
9. Contiguous filled spheroid
10. Isolated cells
11. Contiguous unfilled spheroid
12. Spheroid's core

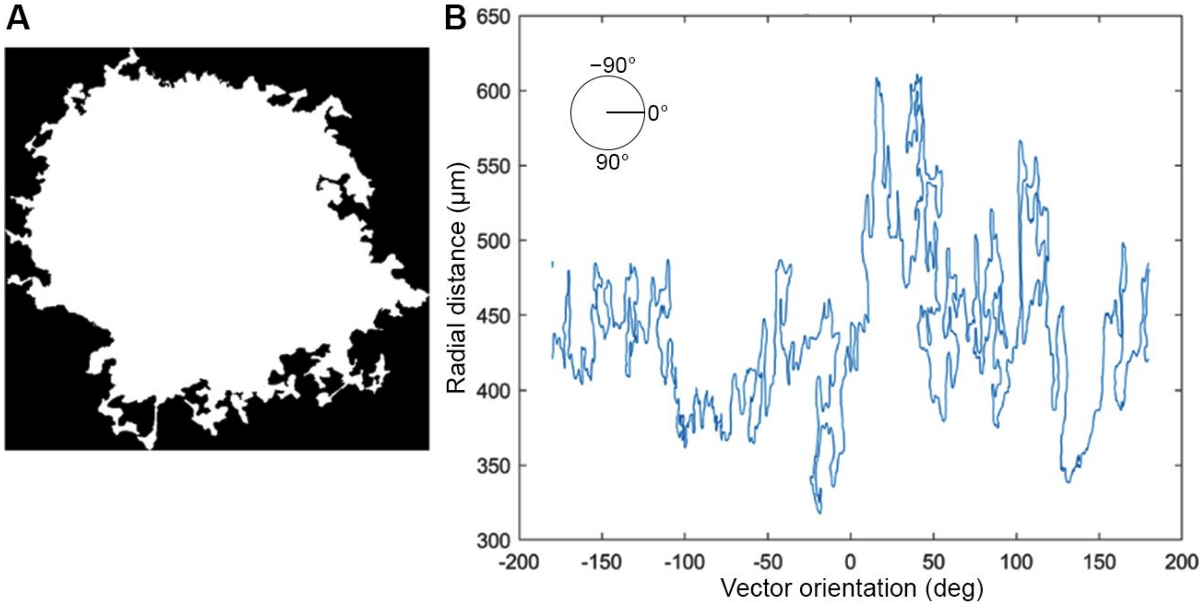
Supplemental Figure S3. Binary image segmentation and detection of spheroid cell migration events. These images were quantified to obtain projected area of spheroid and cell dissemination in addition to shape descriptors. Scale bar: 300 µm.



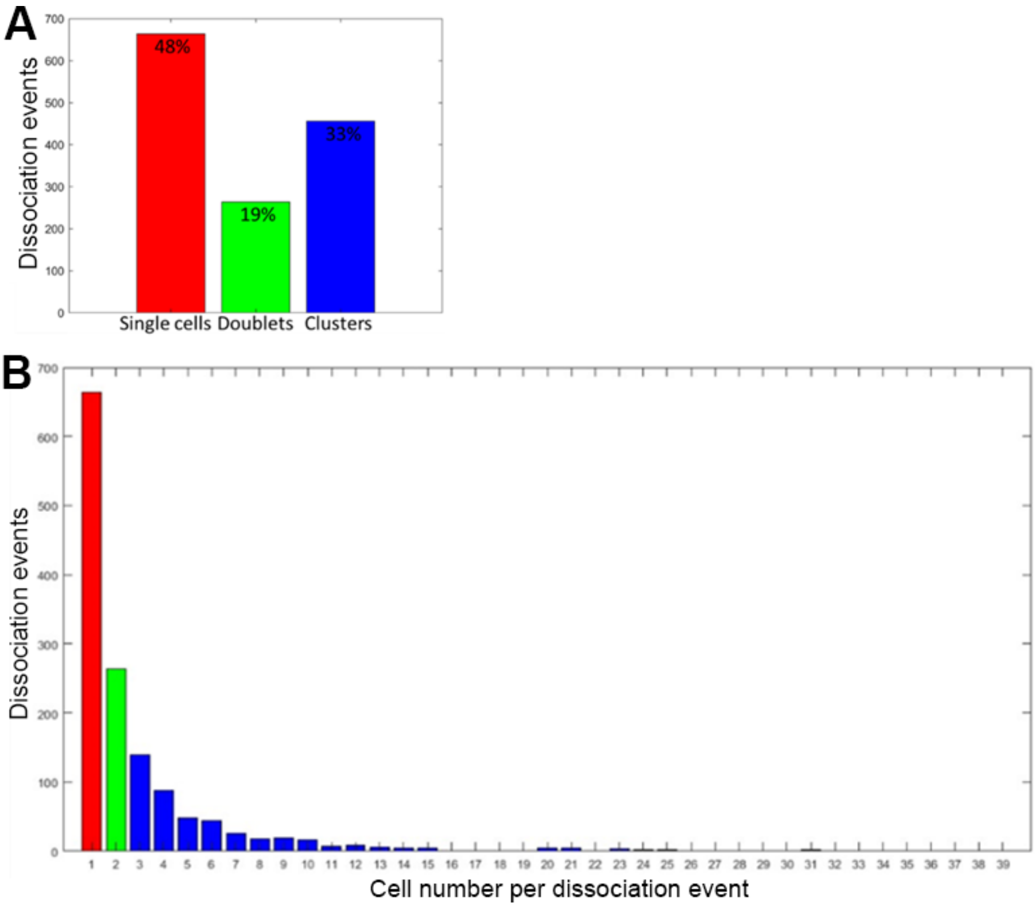
Supplemental Figure S4. High throughput imaging of spheroid dissemination. Images were recorded using 5× objective with each well a tiled composite of 9 fields of view. Top: Hoechst (blue) stained spheroids at 96 h. Bottom: F-actin stained (red) spheroids at 96 h.



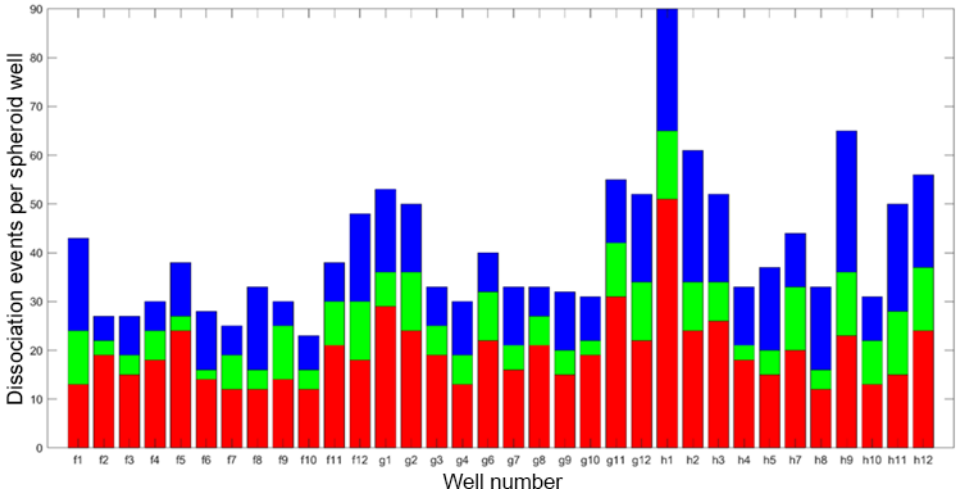
Supplemental Figure S5. Spheroid cell dissemination with distribution of measured parameters (*n* = 34). Boxes represent median and interquartile range, whiskers extend to the max/min data points or 95% confidence interval, with outliers indicated as individual points. A: Circumscribing radius, spheroid equivalent radius, inscribing radius, core equivalent radius, and t_0 equivalent radius. B: Spheroid area and invasion area. C: Farthest isolated cell distance and maximum edge distance. D: Spheroid orientation and angle of maximum edge distance. E: Aspect ratio.



Supplemental Figure S6. Contour analysis of the perimeter of the contiguous cell sheet. A: Binary image depicting a representative contiguous cell sheet disseminated from a U2OS spheroid at 96 h. B: Representative contour line displayed as radial distance from the center of spheroid to migratory front against vector angle. The contour line displayed an irregular shape.



Supplemental Figure S7. Global dissociated events. A: Bar plot indicating number of occurrences of each type of dissociated event. B: Histogram of dissociation events with cell number for each event. Red: single cell, green: doublet, blue: cell cluster. *n* = 34 spheroids with 1,384 dissociated events.



Supplemental Figure S8. Local dissociation events. Stacked bar chart showing distribution of events per well associated with U20S migration on aligned collagen. Red: single, green: doublet, blue: cell cluster. *n* = 34 spheroids with 1,384 dissociated events.

Supplemental Table S1. Definition of parameters used in spheroid dissemination and migration analysis.

Parameter	**Definition**
Area	Number of pixels of segmented spheroid multiplied by squared pixel size.
Perimeter	Distance around the boundary of the spheroid multiplied by the pixel size.
Major axis line	Length of major axis of ellipse that fits the segmented spheroid (i.e., normalized second image moment) multiplied by the pixel size.
Minor axis line	Length of minor axis of ellipse that fits the segmented spheroid (i.e., normalized second image moment) multiplied by the pixel size.
Max/Min feret diameter	Maximum/minimum distance between two antipodal vertices of the convex hull (i.e., caliper diameter) multiplied by the pixel size.
Invasion area	Defined as: $\boldsymbol{Total\ Area - Spheroid\ Core\ Area}$
Maximum invasive protrusion distance	The maximum distance from centroid to farthest connected spheroid boundary.
Average maximum migration (includes isolated cells and cell clusters)	To calculate average maximum migrating distance, first, a circle was drawn matching the spheroid core at t = 0 h (red circle), followed by fitting a convex polygon spanning over the migrating fronts to measure the perimeter (blue line; includes single cells). $\dfrac{\boldsymbol{perim.\ of\ polygon}}{\boldsymbol{2\pi}} - \boldsymbol{radius\ of\ core\ circle}$
Circularity	Is a measure of circularity of a spheroid’s profile on the 2D image. Area-to perimeter ratio: circle is 1, ratio is less than 1 for shapes deviating from a circle; relatively insensitive to irregular boundaries. As the value approaches zero it indicates an increasingly elongated object. Can be obtained as the ratio of the area of an object to the area of a circle with the same convex perimeter. $\boldsymbol{4\pi * Area/Perimeter^2}$
Solidity	A measure of convexity of a spheroid’s profile on the 2D image. The proportion of the pixels in the convex hull that are also in the spheroid, and it is defined as: $\boldsymbol{Area/Convex\ Area}$ A value of 1 signifies a solid object, and a value less than 1 will signify an object having an irregular boundary or containing holes.
Aspect ratio	Denotes the ratio of the major axis to the minor axis: $\boldsymbol{Aspect\ ratio} = \dfrac{\boldsymbol{major\ axis\ legth}}{\boldsymbol{minor\ axis\ length}}$
Orientation (degrees)	Measures the overall direction of the spheroid shape. The angle of the major axis of the ellipse that has the same second moments as the region and rotating clockwise from image x-axis.
Inscribing radius	Defined as the radius of the largest circle that can be inscribed within the spheroid mask, centered at the mask centroid.
Circumscribing radius	Defined from the minimum circle that can encompass the entire spheroid, including any invasive branches.

Image analysis code description

This contains detailed explanation of the custom MATLAB code used for the image processing tasks. The code is structured sequentially as outlined below with each number entry referring to each subsection in the code. To ensure proper running of the code, install the following MATLAB toolboxes: *Image processing toolbox, statistics and machine learning toolbox, curve fitting toolbox*.

1. Set up of path variables and directories to find and isolate the images that are needed for analysis. Afterwards it runs through a loop to go through all these images. It also creates folders to store the images if they are not pre-existing.
2. Image enhancement and refinement to identify relevant information.
3. Isolation of regions of interest, e.g., spheroid core, isolated cells, filled disseminated spheroid, etc.
4. Display of all isolated cells (numbered). User asked to identify non-cellular debris so that it can be excluded from further analysis.
5. Recalculation of isolated features with the debris removed.
6. Centering of the core and disseminated spheroid to image center.
7. Read in t0 image, binarizes and calculate its properties.
8. All the immediately calculatable properties of disseminated spheroid are calculated.
9. Sections 9-19 calculate different graphs and plots, each explained in the code.
20. Write properties of isolated cells in the relevant Excel file.
21. Write average and standard deviation of the properties calculated in section 21 to Excel file. Save box and whisker plots of these properties.
22. Calculate and save property values of the disseminated spheroid to Excel file.
23. Save visualizations of measured properties in measurements folder.
24. Save extent of visualization in measurements folder.
25. Save invasion arrow migration parameter on a single plot.
26. Save the visualized masks that were applied in step 2.
27. Save the visualized masks that were applied in step 3.
28. Save overall combined statistics of all the images from step 1 in a single folder.

Detailed explanation of each portion of code

1. Step 1
   1.1. Algorithm starts with setting up names of all the relevant folders as following:
   - *globalFolder*: A top level folder that contains everything that is needed for the whole analysis. It contains 3 subfolders. These three subfolders are code folder, t96 folder and t0 folder. The exact names for t96 and t0 folder are needed to be provided to the code.
   - *parentFolderPath*: The main folder that contains all the t96 images sorted into subfolders. All images must have the same name, i.e., Alexa 568.tiff.
   - *t0FolderPath*: Folder that contains all the t0 images.
   - *allImageFileNames*: Contains all the names of files that the algorithm needs to look for in each subfolder of parent folder path. It can have multiple variants of same file. The algorithm will perform same procedure regardless of the file name (unless specified in the loop, e.g., for well g2 and g4).

1.2. Read names of all subfolders in the *parentfolderpath* and store in a cell array named subfolders.

1.3. Create the *overallstatistics* folder that holds all the combined statistics of isolated filled spheroids (blobs) of all images.

1.4. Loop over all these subfolders.

1.5. It excludes the *overallstatistics* folder from analysis first as it is created in step 1.4

1.6. Sets up filenames inside each subfolder. More specifically, first set up name of t0 image file path name for current loop iteration and then set up full path for current iteration.

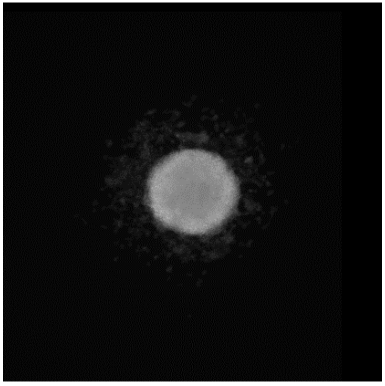
1.7. Creates multiple folders inside each folder to hold relevant images.

2. Step 2

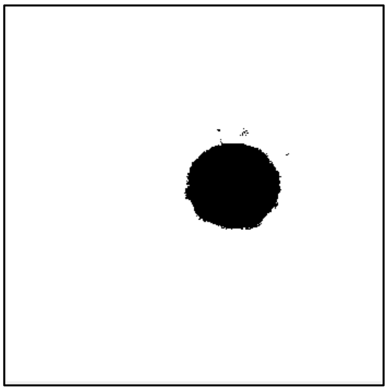
2.1. Read image into workspace as set by 1.5.2. and convert from 8-bit 0-255 range of intensities to 0-1 range of intensities (a). In some special cases (e.g., for well g2, g3 and g4) the image is also cropped.

2.2. Resize the image so that 1 pixel width is equivalent to 1 µm (a).

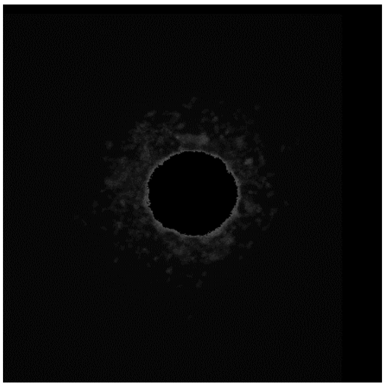
2.3. Adjust the contrast so that the histogram is stretched appropriately (darkest pixel = 0 and brightest pixel = 1). This is done to make images consistent as in the subsequent steps, manual thresholds for intensities are set. Automatic adjustment of the histogram will bring the relevant intensities into the specified range (b).



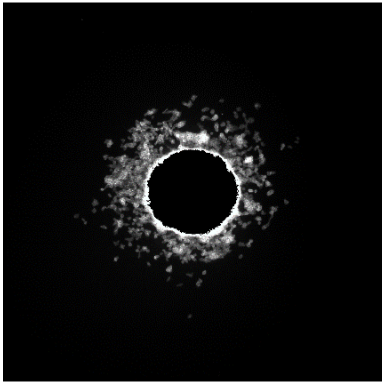
2.4. Brightest cells are identified through intensity thresholding. The threshold is selected manually from histogram (c).



2.5. Brightest cells from 2.4 are removed (d).

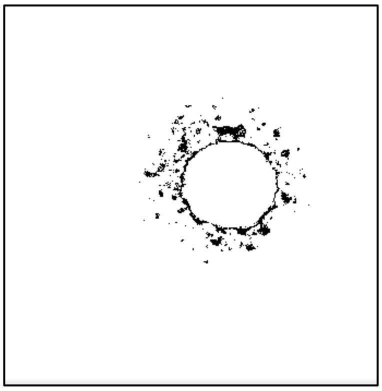


2.6. Intensity of remaining content from 2.5 is readjusted to obtain better contrast. The values for adjusting thresholds are selected through the **imcontrast** tool (e).

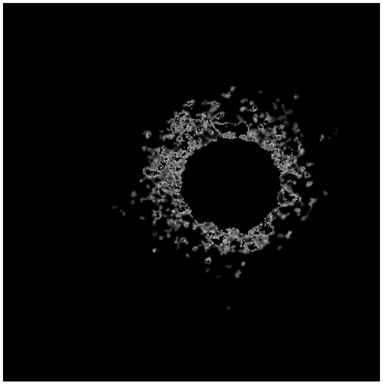


2.7. The subsequent histogram stretching in 2.3 and 2.6 introduces noise in the image. That noise is eliminated with an adaptive low pass filter (f).

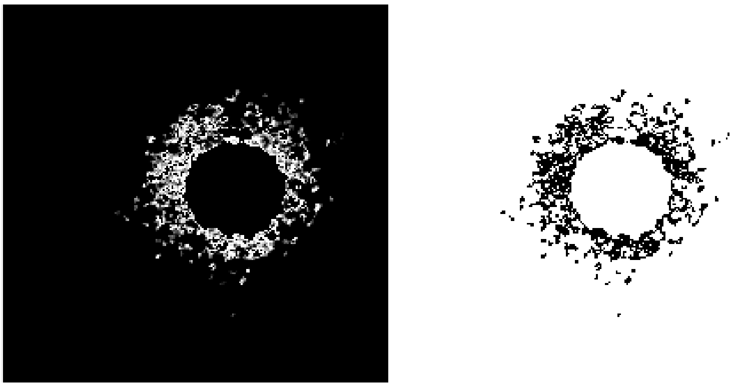
2.8. Intensity thresholding identifies the brightest cells in the image. This step aims to isolate the protruding cells which are clumped together. These cells are generally brighter than relatively separated cells so removing these cells beforehand will improve thresholding in the following step (h).



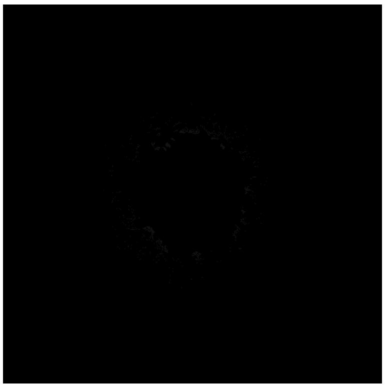
2.9. Removes the brightest cells from 2.8 (i).



2.10. Readjust intensity again to obtain better contrast. Isolate and remove all the useful information (j, k).



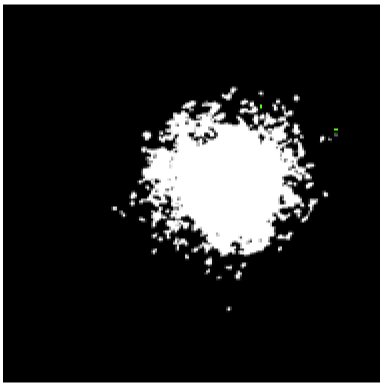
2.11. Any pixels not included after multiple thresholding steps is determined to be leftover content (i.e., not a cell and therefore not relevant to analysis).



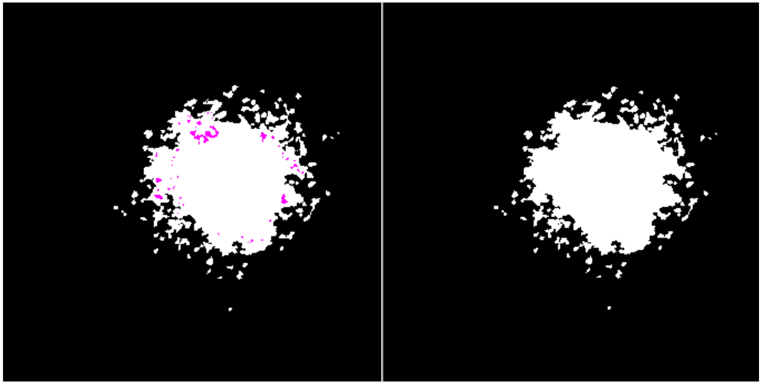
2.12. The combined binary image with all the information is obtained through creating a composite image of core, brightest(clumped) cells, and rest of the protruding cells. Practically, it is done by adding inverted c, h, and k images (2.4, 2.8 and 2.10).

3. Step-3

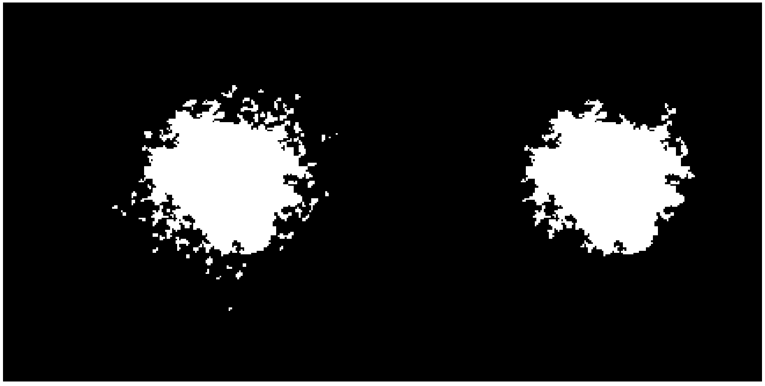
3.1. Objects smaller than 85 pixels in size are removed; green pixels are removed content (*finalWithoutDebris*).



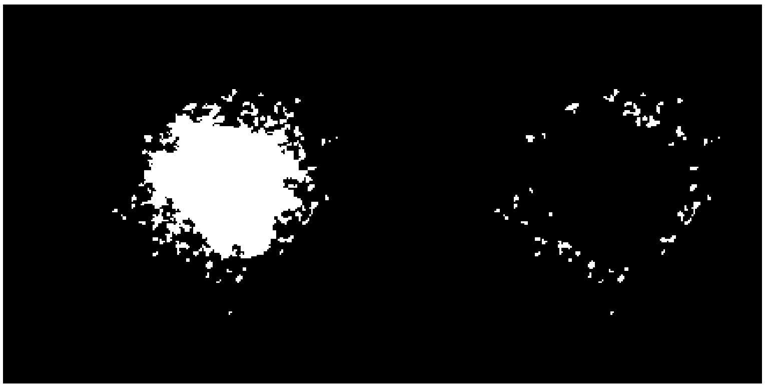
3.2. Holes are identified (magenta, left) and closed (filled, right).



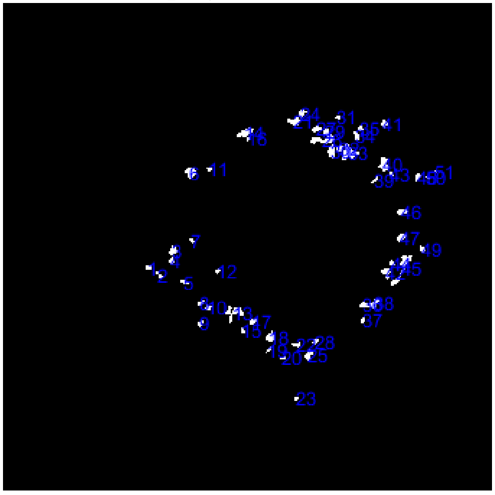
3.3. Connected cells are isolated through extracting the largest object in the image (blob).



3.4. Isolated cells are obtained through identifying everything that is not blob or 3.3 (*isolatedCells*).



4. Step 4
    4.1. Find the centroid of all isolated cells from 3.4 (*cellCenter*).
    4.2. Assign a number to each cell and display it next to the cell.



4.3. Convert 3.4 to a labeled image so that each isolated cell has uniform pixel value. E.g., all the pixels for 2.3 may correspond to a single unique number (say 50). (*bwl*)

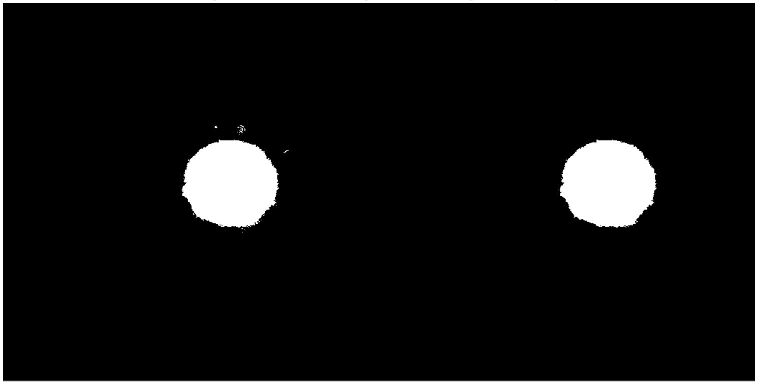
4.4. Ask user to identify the debris from 4.2 and find the label value at the centroid of that number as per 4.1.

4.5. Find label value at that position and turn that unique label to zero so those pixels are turned off.

4.6. Repeats this step till user inputs "-1".

4.7. All isolated cells, removed cells, and filtered isolated cells are stored in separate variables.

5. Step 3 is recalculated to remove debris from all the images.
6. Step-6
    6.1. The core is identified through extracting the largest object from 2.4 or c.



6.2. Holes are filled inside core because in some images, the core isn't filled completely

6.3. Centroid of the core is identified, and blob is recentered with core's centroid at center of image. This is used in some of the subsequent steps which is why it is calculated beforehand.

7. Read t0 image, resize it, and find its area and centroid.
8. All properties that can be calculated directly through **Regionprops** are identified and calculated. **Regionprops** stores all the measurements in a cell array. Blob centroid and isolated cell centroids are then stored in a separate matrix as they will be used in the following steps multiple times so keeping them in a matrix is useful.

- *blobProps* stores properties of blob
- *isolatedCellsProps* stores properties of isolated cells

9. Convex hull is plotted by finding the coordinate points of the convex hull through **regionprops**. These points are then identified and plotted on top of the blob (disseminated spheroid).
10. This step identifies the boundary point coordinates of the blob in both x-y and r-theta coordinates. First the boundary coordinates are traced through **bwboundaries**. The coordinates obtained through this function have origin at top left of the image as top left pixel is registered as first element of the image matrix. The coordinates are then recentered so that the origin is at the core's centroid obtained in 6.3 instead of the image origin.
11. Step-11
    11.1. Plots coordinates obtained in step 10 in three separate figures as follows:
    - x-y plot of filled spheroid
    - x-y plot of spheroid's core
    - x-y plot of spheroid at t0

    11.2. Step 11 also coverts these coordinates to the polar r-theta coordinate system.

To convert to r-theta coordinates, following formulas are used:

$$r = \sqrt{x^2 + y^2} \qquad (1)$$

$$\theta = arctan\frac{y}{x} \qquad (2)$$

11.2.1. *edgeDist* corresponds to *r* for blob, *edgeDist2* corresponds to *r* for core and *edgeDist3* corresponds to *r* for t0 image.

11.2.2. *angs* corresponds to theta for blob, *coreAngs* corresponds to theta for core and *t0Angs* corresponds to theta for t0.

12. Step-12
    12.1. Plot the blob in polar coordinates obtained in 11.2
    12.2. Plot the smoothed polar coordinates, with an averaging window of 20, on top of 12.1
13. Raw signature analysis plotting: plot angles vs. edge distance, obtained from 11.2.1 and 11.2.2 on x-y axis.
14. Set up vectors for following steps as follows:
    14.1. *smEdgeDist* stores the smoothed edge distance from core's centroid.
    14.2. *maxDistance* and *maxDistance2* store maximum distance from core at angle theta. This is stored in the increment of 1 degree.
    14.3. A vector with 0-360 degrees with 1 degree increment.
    14.4. *X-smoothed* is smoothed x-coordinates through a moving average filter.
    14.5. *Y-smoothed* is smoothed y-coordinates. These two are obtained through *edgeDist* because direct smoothing will result in erroneous coordinate points. Doing it this way will give smoothed out coordinate points with higher accuracy.
15. Smoothed radial distance analysis: Find maximum radial distance at a given angle by intersecting an arbitrary vector at that angle by the boundary curve from 16.4 and 16.5 and select the point of intersection that is at maximum distance from center. Points of intersection with vector are found through **InterX** function obtained through MATLAB's shared files.
16. Repeat step 15 with original coordinates instead of smoothed coordinates.

17. Calculate remaining shape properties of isolated cells.

18. Write the isolated cell statistics to an Excel file.

    18.1. Remove the centroid field from *isolatedCellsProps* structure. As it is the only element in the structure that has more than 1 column for a given cell, writing it directly raises an error. Convert the structure to a cell array and then to a matrix.

    18.2. Calculate average and standard deviation for isolated cells statistics. Add two blank columns for centroids and write this new array in the existing Excel file.

19. Calculate remaining blob properties and write to a separate sheet in Excel file.

20. Visualize measurements carried out so far. All the variables that are used for these visualizations are temporary, so they are bunched together in a structure named *mms*. A custom function is used to save the images. Function name is **measurementsSaver** and its syntax is as follows:

Input Number	**Syntax requirement**
1	Primary image on which the measurement is needed to be visualized.
2	Title for the image.
3	Target folder directory.
4	A cell array with *n* elements for styling of *n* different plots that are needed to be overlaid on the primary image.
5 to 5+2*n	X-Y coordinate pairs of the plots that are needed to be overlain on top of primary image. Note that the X and Y vectors need to be provided as separate arguments in function.

21. 21 and 22 use a customized function that saves the masks used in step 2 and 3. This function is named as *imageSaver* and it takes following inputs:

Input number	**Syntax requirement**
1	Image to be printed.
2	Centroid from which the image is needed to be recentered.
3	Target folder directory for saving the image.
4	Image title.
5	If image comes in a numbered series, number of this image (optional).

23. Write the overall combined statistics into the overall statistics folder.