Supplementary Information Table of Contents

Materials

All cell lines used in spheroid production were purchased from ATCC, with the exception of tdTomato-expressing B16F10 (donated by Dr. Steven Proulx), GFP-expressing U87MG (donated by Dr. Gang Zheng), GFP-expressing human umbilical vein endothelial cells (HUVECs, purchased from Angio-Proteomie), tdTomato-expressing MDA-MB-231 (donated by Dr. Michael Detmar). Culture media, trypsin-EDTA, and CFDA-SE and CellTracker Red cellular stains were purchased from Life-Technologies. Poly-hydroxyethylmethacrylate (poly-HEMA) was purchased from Sigma-Aldrich. Monomer solution was prepared with either 0%, 2%, 4%, or 8% acrylamide (BioBasic), mixed with 4% paraformaldehyde (Electron Microscopy Sciences), and 2.5% of 2,2'-azobis[2-(2-imidazolin-2-yl)propane] Dihydrochloride radical initiator (Va-044, from Wako Chemicals) in PBS. Clearing solution was prepared with 4% sodium dodecylsulfate (SDS) in 200mM sodium borate (pH 8.5 with pre-dissolved NaOH in deionized water). Refractive Index Matching solution (RIMS) was prepared with 88% Iohexol (purchased from AK Scientific), 2.5% 1,4-diazabicyclo[2.2.2]octane (Sigma-Aldrich), 50 mM sodium borate at pH 8.5, and 0.01% sodium azide.

Methods

Spheroid Production

Spheroids were produced from the MCF-7, MDA-MB-435, B16F10 (endogenously expressing tdTomato), U87MG (endogenously expressing GFP) cell lines, and with HUVECs. Cells were grown to 80% confluence in a T75 cell culture flask, and incubated for 5 minutes with 0.25% Trypsin-EDTA at 37°C to release them into solution. After removal of excess trypsin via centrifugation at 300*g* for 5 mins, cells were diluted in complete media (RPMI with 10% FBS and

1X Penicillin-streptomycin for MDA-MB-435 cell line, DMEM with 10% FBS and 1X Penicillin-streptomycin for MCF-7, U87MG cell lines, and GlutaMAX DMEM with 10% FBS and 1X Penicillin-streptomycin for B16F10 cell line) to 10,000 cells/mL. Using this mixture, 100 µL (1000 cells) were added to each well of a round-bottom 96-well plate that was previously coated with 50 µL of 2.5% poly-HEMA in 95% ethanol. Cells were centrifuged at 1800 rpm for 10 minutes at 4°C, and subsequently grown for 3 days inside a 37°C incubator.

Preparing Fluorescent Spheroids

Cells that were not endogenously expressing fluorescent proteins were stained with CFDA-SE, or with CellTracker Red prior to spheroid formation. After release from the T75 cell culture flask, the cells were incubated with 10 µM of staining solution at 37°C for 20 mins. After washing with sterile PBS (300*g* for 5 minutes), cells were diluted in complete media to 10,000 cells/mL, and spheroids were produced from them as previously described.

Microfluidic Device Fabrication

The microfluidic chip design was originally designed for murine islets of Langerhans, taken with permission from Silva *et al*.[30] For devices intended for use with tumour spheroids, all dimensions were doubled, measuring 600 µm at the 'mouth' of the spheroid capture wells, 100 µm at the outlet of the capture wells, and an average of 350 µm in channel height. Polydimethylsiloxane (PDMS) microfluidic devices were fabricated using standard soft lithography techniques. SU-8 negative photoresist (Microchem) was spin coated onto a silicon wafer and soft baked at 65°C for 8 minutes, 95°C for 70 minutes, and 65°C again for 8 minutes . After selective exposure through a UV photomask, the photoresist was further baked at 65°C for 5 minutes, 95°C for 25 minutes, and 65°C for 5 minutes again. The SU-8 was developed for 30 minutes, and the height of the features

was measured with an optical profilometer (Bruker Contour GT-K). Using this silicon master, Silgard 184 silicone elastomer was mixed with the supplied curing agent (DowCorning) at a 10:1 ratio (by weight), and allowed to solidify on the silicon master under applied vacuum for 1 hour at room temperature. The temperature was then raised to 80°C for 3 hours. After solidification, the silicon master was cooled to room temperature and the PDMS was carefully peeled off of the master. Glass cover slips were then sonicated in 1 M HCl for 10 minutes, followed by sonication in acetone for 10 minutes, followed by an isopropanol rinse, and a methanol rinse. Clean the dust off of the PDMS pieces with repeated application of scotch tape. Both the PDMS pieces and the glass cover slips were then treated with oxygen plasma for 2 minutes and quickly bonded together. The devices were briefly heated to 60°C to further promote bonding.

On-chip monomer infusion

The microfluidic device was first briefly flushed with ethanol, followed by a second flush with spheroid-conditioned media, or by iBMHH buffer (125 mM NaCl, 5.7 mM KCl, 2.5 mM $CaCl_2$, 1.2 mM $MgCl_2$, 10 mM HEPES, 0.1% BSA, 11 mM glucose) for islet experiments. Microtissues were loaded via gravity flow, and confocal images were taken with a Zeiss LSM 710 microscope. The monomer solution was infused at 600-800 µL/hr (200 µL/hr for islets) for 20 mins, then the flow was stopped while another set of post-monomer infusion confocal images were taken. The tubes leading out of the chip were clamped shut, and the chip was disconnected from the syringe pump (New Era) and incubated at 37°C for 2 hours to gel the monomer. After gelation, a brief PBS rinse (iBMHH for islets) was gently flushed through the device to restart flow. This is necessary to flush out the excess monomer that has gelled in the device. Post-gelation confocal images were taken.

On-chip membrane disruption and staining

After monomer infusion and gelation, the MDA-MB-435 spheroids in the microfluidic device were infused with clearing solution for 7-10 minutes at 600-800 µL/hr, followed by a 20 minute wash with PBS. The SDS infusion was omitted for control spheroids. Both membrane-disrupted and non-membrane-disrupted spheroids were stained with phalloidin-488 at 0.165 µM for 20 minutes), DAPI (0.1 µg/mL) and 6 M transferrin-647, for their actin, nuclear DNA, and transferring receptors respectively at 600-800 µL/hr for 20 minutes per stain, followed by a 10 minute PBS rinse. Confocal images were taken again before RIMS was infused at 600-800 µL/hr for 10 minutes, after which the microtissues became transparent, and more sets of confocal images were taken. For on-chip live/dead staining, a fixable Near IR LIVE/DEAD stain (Life Technologies, L10119) was infused at 400 µL/hr for 1 hour, followed by a PBS rinse at the same flow rate for 20 minutes. This live/dead staining was done *before* monomer infusion, since it discriminates between live and dead cells by the intactness of their cellular membranes.

Islet extraction

C57/Bl6 mice were anesthetized with 50 µL of a 100 mg/mL ketamine and 20 mg/mL xylazine mixture via intraperitoneal injection. The pancreas was excised, mechanically diced in cold HBSS, and digested for 10-15 minutes in 1 mg/mL collagenase P (10mL total volume), under vigorous shaking conditions at 37°C. After digestion, the islets were handpicked and placed in RPMI supplemented with 10% FBS, 1X penicillin-streptomycin, 11 mM glucose. Islets were incubated at 37°C until they were infused onto microfluidic device for clarifying treatment and imaging.

Vasculature Dye Conjugation

Dye conjugation performed according to Sindhwani et al.(31), and transcribed with permission here: Alexa Fluor 647 conjugated Griffonia Simplicifolia Lectin 1 (GSL1-A647) was prepared by adding 100 µL of 10 mg/mL unconjugated GSL-1 in 100 mM Sodium Bicarbonate buffer (pH 8.3) to 100 µg of lyophilized Alexa Fluor 647 NHS ester. The mixture was vortexed and incubated at room temperature overnight and purified by size exclusion through a NAP-5 column (Sigma GE17-0853-01) and further washed with PBS and concentrated using Amicon 3 kDa cutoff centrifugal filters (Millipore, UFC500324). Degree of labelling was determined by measuring absorbance at 647 nm and 260 nm after completing purification and concentration steps. Degree of labelling varied from batch to batch between 4-6 dyes per molecule.

Islet Vasculature Imaging

Prior to monomer infusion, GSL1-A647 was infused at 0.1 mg/mL at 200 µL/hr for 25 minutes. After a brief iBMHH rinse, monomer was infused at 200 µL/hr and the rest of the on-chip clearing process proceeded as previously described. SytoxGreen was used to stain the islet nuclei, and was infused for 30minutes at 200 µL/hr at a concentration of 1 µM, just prior to RIMS infusion. Clearing with SDS was not found to affect staining efficacy of SytoxGreen. Therefore, before and after confocal images of RIMS infusion were taken of islets that did not undergo lipid disruption.

On-chip Visualization in 3D

After the desired staining, monomer gelation, or membrane disruption treatments were completed, microtissues were infused with RIMS at 800 µL/hr (for tumour spheroids), or at 400 µL/hr (for murine islets) for 10 – 15 minutes. Confocal images were then taken with a 20X objective lens

using a Zeiss LSM710 confocal microscope. Image stacks were then fed through the MATLAB script for analysis.

Volume shrinkage and imaging depth quantification

The spheroids that were used in volume shrinkage and imaging depth quantification did not undergo SDS lipid disruption. Spheroid images were quantified for their volume shrinkage and imaging depth by MATLAB (Supplementary code SC1 'Z_stack_area_generation.m'). The code analyzes each slice of a z-stack of a spheroid, applies a slight blur and determines the area of the image where the spheroid exists. It lists the intensities of the pixels within the region that are above a threshold generated by a thresholding function and quantifies the amount of positive pixels within that region. For volume shrinkage quantification, this process repeats for each slice in the z-stack image sequence to find the image with the largest cross-sectional area. To collate the data, use Supplementary code SC2 'Collect_Volume_and_Area_data.m', which looks through the directory and extracts the relevant data into a master xlsx file. The largest cross-sectional area of spheroids before and after on-chip RIMS infusion were divided to determine the extent of volume shrinkage. Note that individual spheroids were compared to themselves, and that the shrinkage is not averaged across all the replicates. For quantifying imaging depth increase, the number of positive pixels of each image slice is divided by the total number of pixels within the same region. If this ratio is greater than 25%, it counts this slice as an acceptable image. Spheroids were imaged before and after the RIMS infusion, and the number of acceptable slices obtained after RIMS infusion is divided by the number of acceptable slices before, to generate the ratio seen in Figure S3(i). To prevent bias due to volume shrinkage, only MCF-7 spheroids that were treated with 4% and 8% acrylamide, along with U87MG/GFP spheroids treated with 8% acrylamide were used to calculate the increase in imaging depth. Spheroids that shrink after RIMS infusion would show

very little increase in imaging depth, since the shrinkage decreases the absolute number of slices that exist in the spheroid, and would mask any increase in imaging depth.

Diffusion versus flow rate quantification

To compare the effect of flow rate on interstitial media exchange, B16F10 spheroids were grown, loaded, and treated with 4% acrylamide as described. RIMS was infused into the device briefly before the fluidic flow was stopped by pausing the pump and clamping the tubing. The RIMS was allowed to diffuse into the spheroid for an hour, during which time fluorescent images were taken every 15 sec with 1.5 sec of exposure each time. To ensure that the spheroid had not reached maximum fluorescence after the allotted diffusion period, flow was restarted and media exchange was allowed to proceed (see Figure S4b for fluorescence intensity profile). We chose this method to measure diffusion-induced fluid exchange because if spheroids were loaded onto a device that already had RIMS within it, the fluid flow that the spheroids would experience on the way to the interrogation wells cannot be reproducibly controlled, and would induce interstitial fluid exchange before diffusion can be measured. In a separate device, spheroids were first infused with RIMS at 200 µL/hr. Fluorescent images of the spheroid were then taken as the interstitial RIMS was subsequently exchanged with PBS at 200 µL/hr. Following this, RIMS was re-infused at 400 µL/hr, replaced by PBS at 400 µL/hr, re-infused at 800 µL/hr, and finally replaced by PBS at 800 µL/hr. The fluorescent intensities were quantified by MATLAB, which was written to calculate the average intensity of each image in the time sequence. The time-varying fluorescence intensities can be seen in Figure S4. Fluorescence intensity profiles for 200, 400, and 800 µL/hr flow rates were fitted to an exponential decay model (Figure S4b and Supplementary Table 1), and the half-lives were inverted and graphed in Figure 4. For diffusion, the half-life was extrapolated by taking

the linear slope generated in the diffusion region of Figure S4a and determining the time that is required to increase fluorescence by 50% of the maximum intensity.

Islet Vasculature Analysis and Quantification

Islet Vasculature analysis and quantification was performed in MATLAB, using the DIP image toolbox v2.7, provided in Supplemental Code SC3 "connectivity_analysis.m." A threshold is first applied to the image stack to exclude any background signals. It is then converted into a binary format that detects whether a voxel (pixel in 3D) is fluorescent or not, then looks for neighbouring voxels positively fluorescent to determine the total number of voxels in one inter-connecting vessel network. Since each voxel is a known volume, the total volume of the vessel network can be determined. The code also determines the total islet volume after the convex hull function has been applied to the image stack. Dividing the volume of the three largest vessels by the total islet volume generates the values in Figure 6e. The islet core is determined by applying the distance transform function and finding the inner area that is defined when the computer has gone 50% of the maximum distance possible inwards. The proportion of the largest 3 vessels that are within this inner area is determined to give the values found in Figure 6f. Dividing the total volume of the three largest vessels by the total vasculature volume (not the total islet volume) yields the connectivity values in Figure 6g.

Supplementary Figures

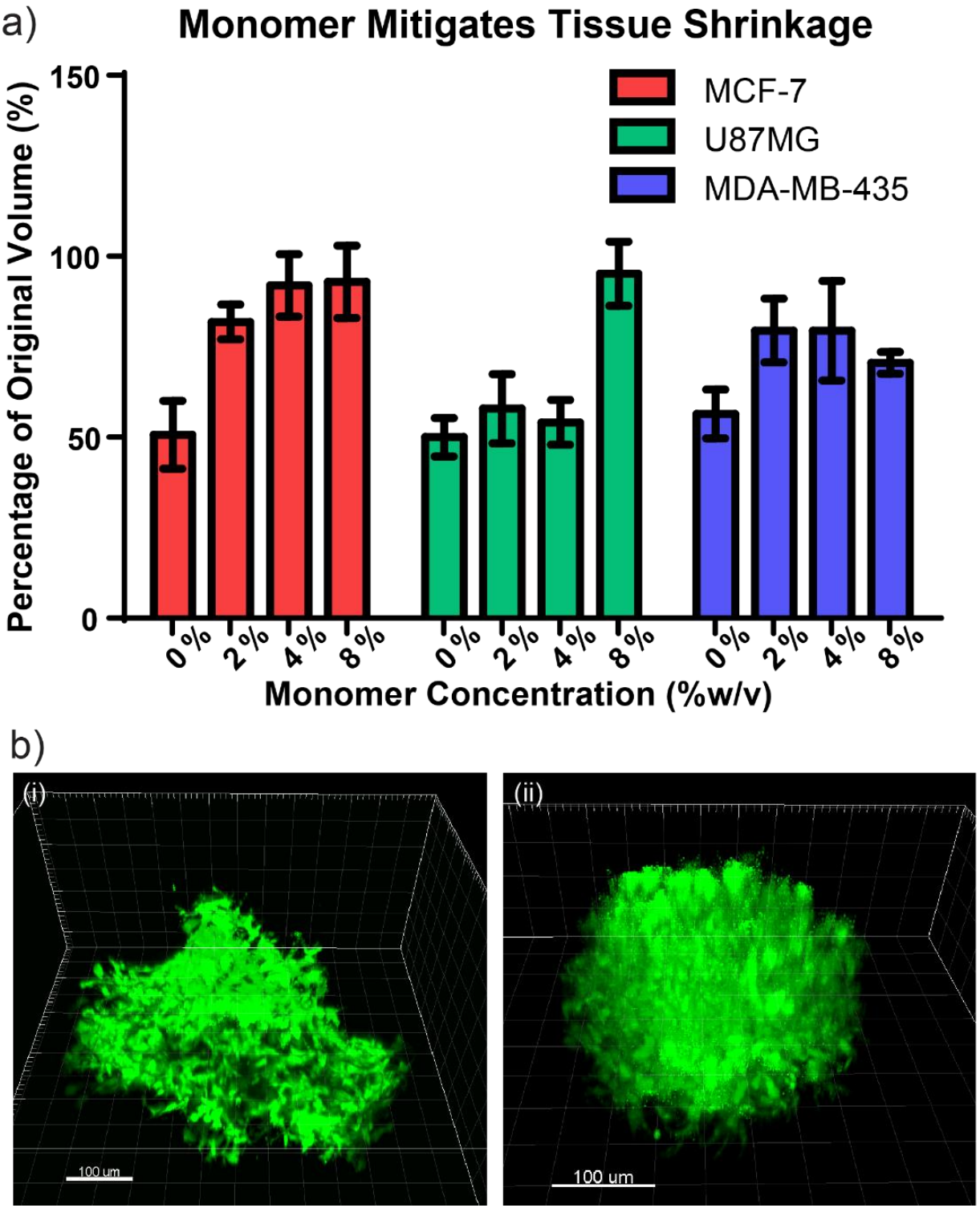


Figure S1: ***Tissue Preservation depends on Monomer Concentration.*** a) The optimal acrylamide monomer concentration to prevent tissue shrinkage varies with tissue type and the cell line of origin. b) Tissues will be distorted by fluidic forces upon RIMS infusion if acrylamide is omitted. i) U87MG/GFP spheroid fixed with 4% formaldehyde with no acrylamide, ii) U87MG/GFP spheroid treated with 4% formaldehyde and 8% acrylamide. Green – green fluorescent protein, ex. 488 nm, em. 493-598 nm. Scale bar is 100 µm.

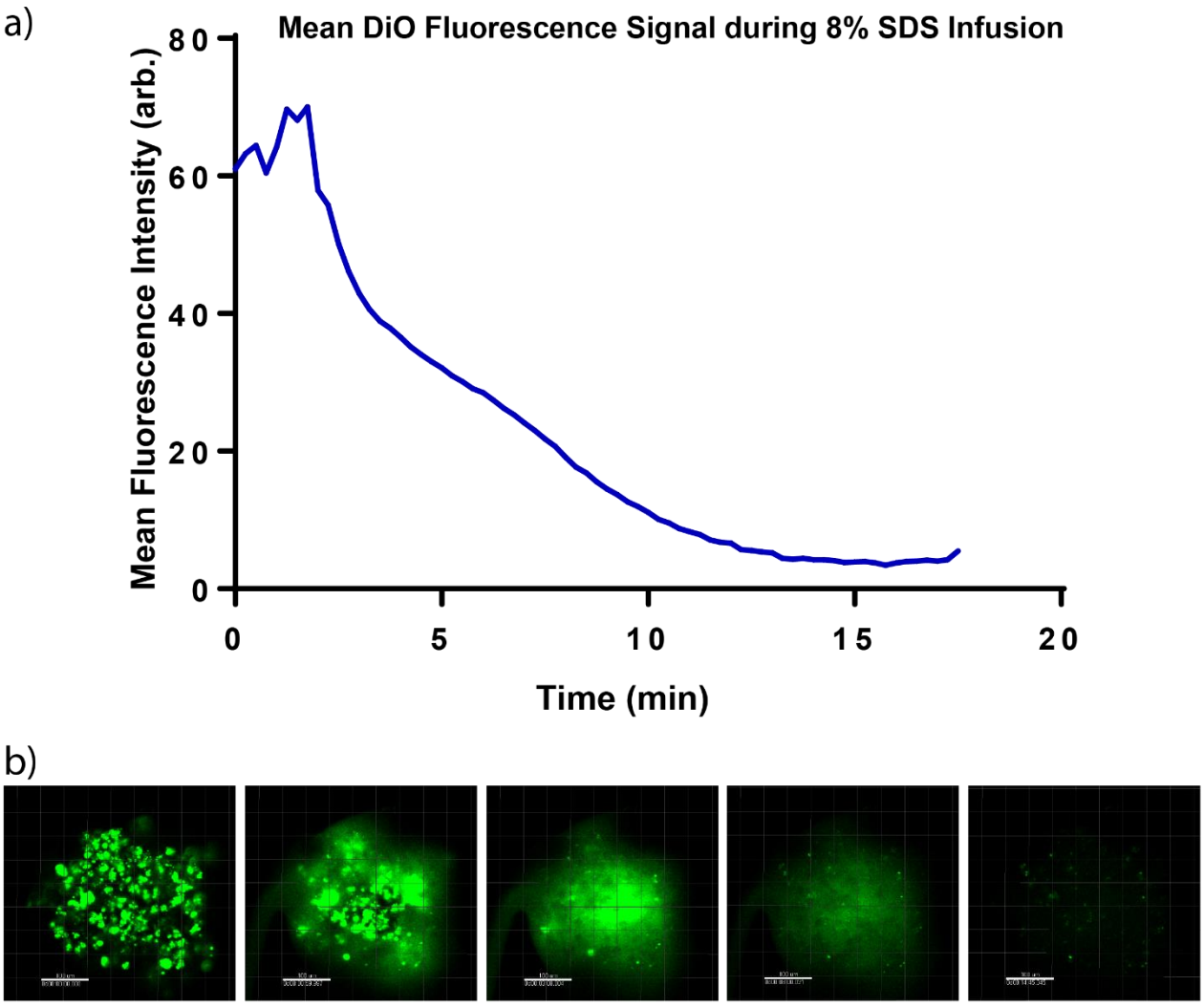


Figure S2: ***Membrane Disruption Completes in Minutes*****.** Membranes of MDA-MB-231 cells are stained with DiO prior to spheroid generation. DiO fluorescence signal (shown in green, ex. 458 nm, em. 481-538 nm) decreases as the lipid bilayer membrane is washed away under 8% SDS infusion at 800 µL/hr. Membrane disruption is completed at approximately 3-5 minutes, and lipid clearance is completed in 12-15 minutes. Scale bar is 100 µm.

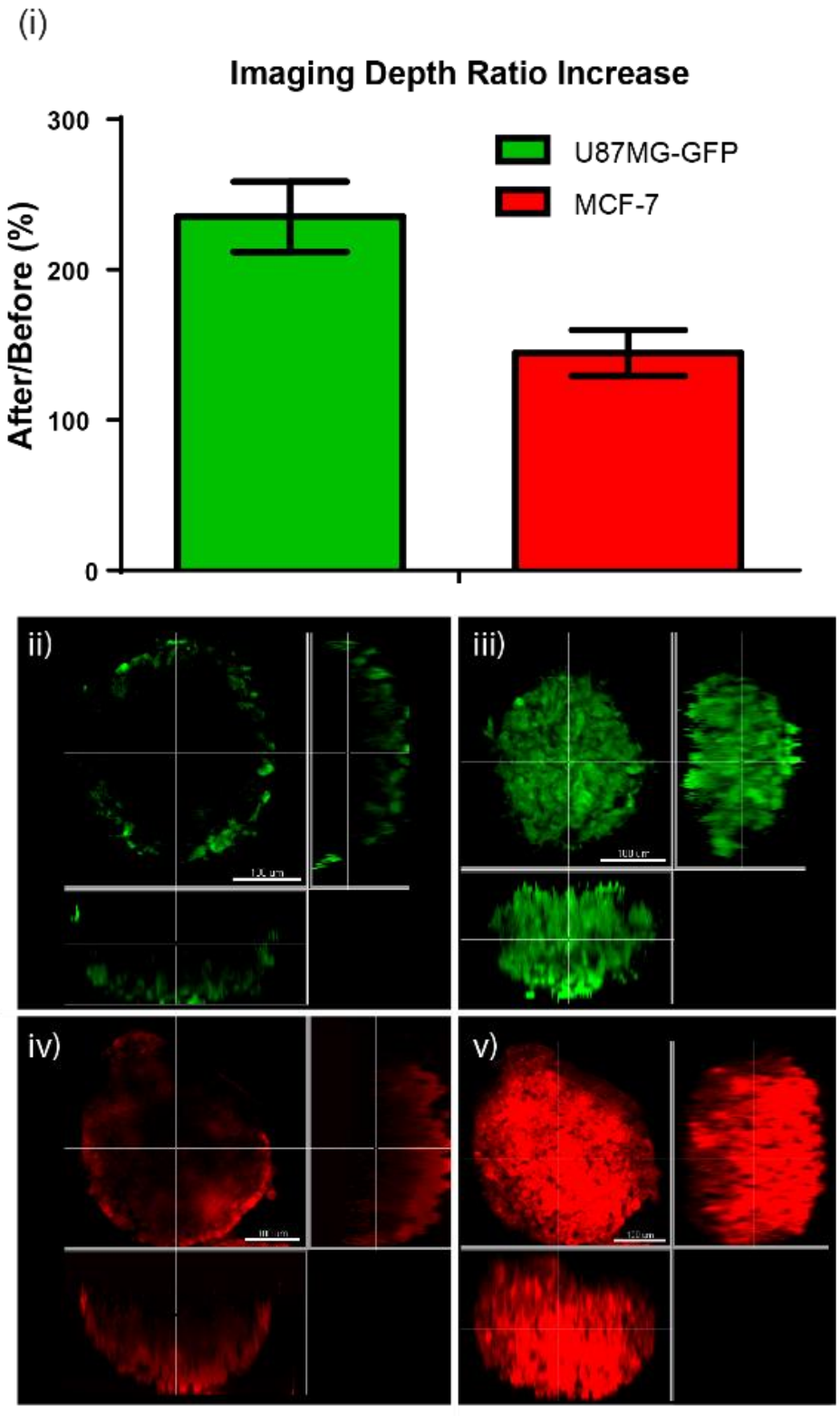
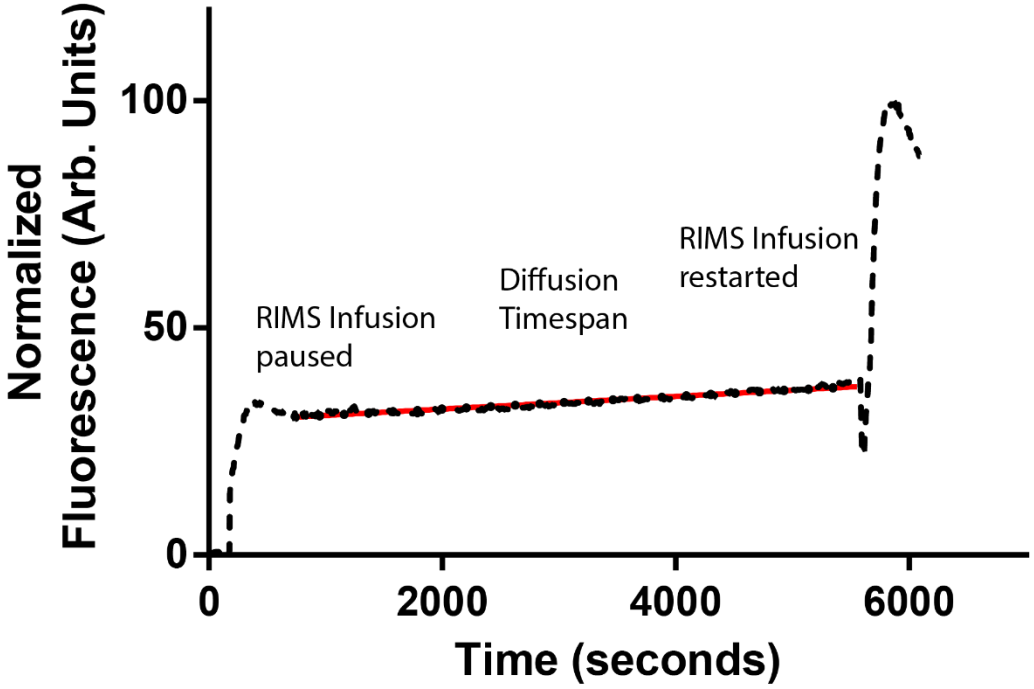


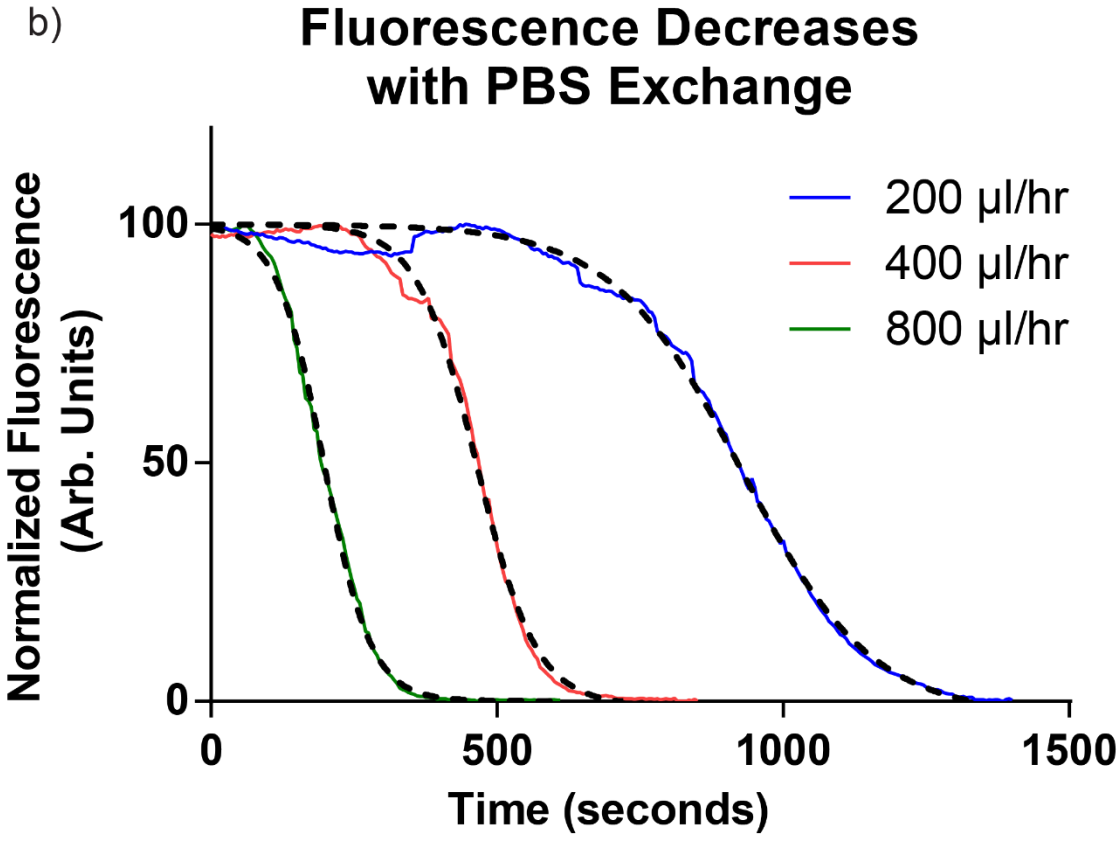
Figure S3*: Imaging Depth is Dramatically Increased after on-chip Clearing.* (i) Imaging Depth increase, calculated by dividing the number of acceptable image slices after treatment with that of

before treatment. An acceptable image slice must have at least positive pixels in 25% of the spheroid area. (ii) Cross section images of U87MG-GFP spheroids before and (iii) after clarifying treatment (green – GFP ex. 488nm, em. 493-598 nm). (iv) Cross section images of MCF7 spheroids, stained with CellTracker Red (ex. 543nm, em. 567-739nm) before and (v) after clarifying treatment. Scale bars are 100 µm.

a) **Diffusion Dependent Fluorescence Increase**



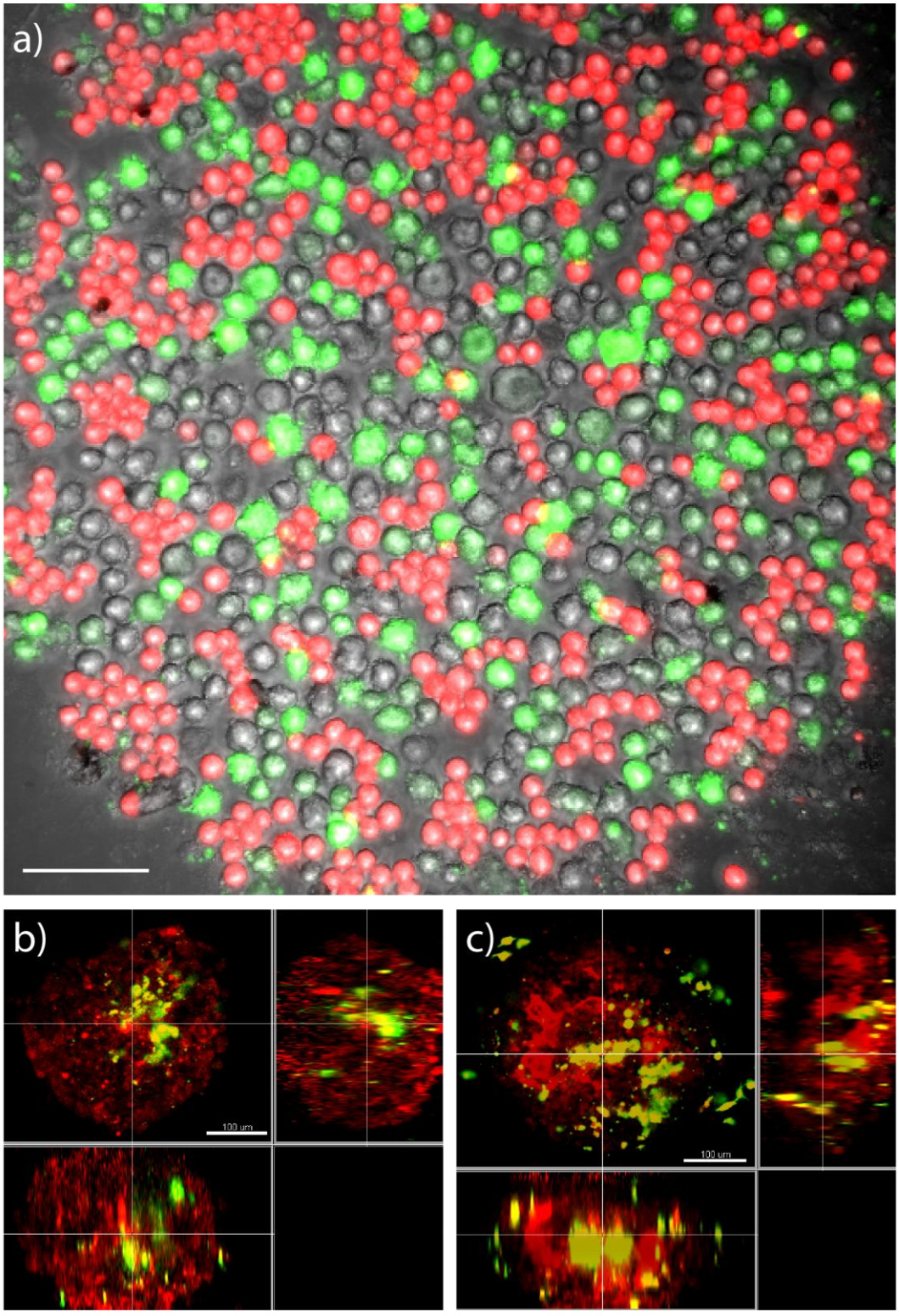
b)



1

Figure S4: ***Interstitial Fluid Exchange under Induced Flow or Diffusion.*** a) Diffusion-driven fluid exchange was characterized by infusing RIMS onto the device briefly before the flow was interrupted. RIMS was allowed to diffuse into B16F10 spheroids while fluorescent images were collected. Flow was restarted to ensure that maximum fluorescence intensity, and therefore complete media exchange, has not yet occurred. Fluorescence intensity (ex. 543 nm, em. 565-702 nm) was quantified using MATLAB as described in Supporting Information. b) RIMS within B16F10 spheroids were exchanged with PBS under different flow rates, leading to a decrease of fluorescence that is measured through imaged time series (5 sec scan interval, ~1 sec scan time).

1



2

Figure S5: ***HUVEC Migration to Spheroid Core varies with Cancer Cell Type.*** a) Cell pellet of a mixture of B16F10/HUVEC hybrid spheroids (Red - B16F10-tdtomato ex. 543 nm, em. 565-685 nm, green – HUVEC-GFP ex. 488 nm, em. 493-536 nm). Cells were initially homogeneously mixed and centrifuged to form a cell pellet, which forms a spheroid and was imaged in Figure 5a on day 3 of incubation and found to have a core of HUVECs. b) Spheroid composed of MDA-MB-231 breast cancer cells (red – tdTomato ex. 543 nm, em. 564-744 nm) mixed with HUVECs (green – GFP ex. 488 nm, em. 493-536 nm) form a core-shell structure, whereas in c), MDA-MB-435 metastatic melanoma cells (red – CellTracker Red ex. 543 nm, em. 565-685 nm) mixed with HUVECs (green – GFP ex. 488 nm, em. 493-536 nm) has the HUVECs congregating along the periphery as well as some parts of the interior. Scale bar is 100 μm.

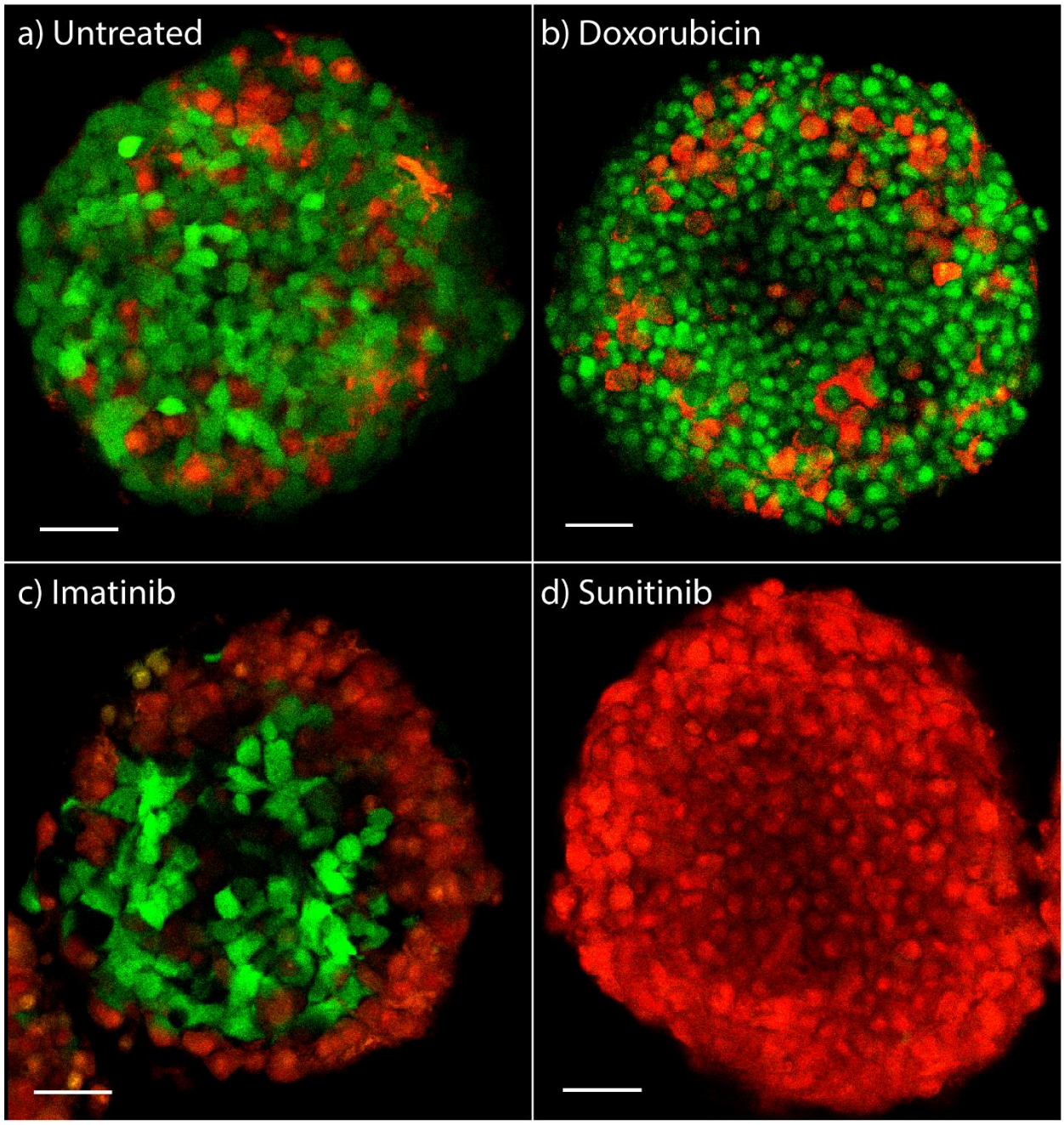


Figure S6: ***Proof-of-Concept Drug Screening Trial.*** Spheroids composed of B16F10 cancer cells were a) untreated, or treated for 3 hours in static, complete media with b) doxorubicin (180 µM), or with c) imatinib (180 µM), or with d) sunitinib (180 µM), before being extracted from static culture and placed on a microfluidic chip. Spheroids were infused with a fixable Near-IR live/dead stain, and dead cells are shown here in red, with live cells shown in green. (Green -

B16F10-tdtomato ex. 543 nm, em. 565-685 nm, red- Near-IR live/dead stain ex. 633 nm, em. 667-753 nm). Images show that anti-cancer activity is highest for sunitinib (complete spheroid death), followed by imatinib (only peripheral death), and followed by doxorubicin (indistinguishable from untreated spheroid control). Scale bar is 50 μm.

Supplementary Table 1: Variable-slope Sigmoidal fitting characteristics from Figure S4b

	200ul/hr	400ul/hr	800ul/hr
Best-fit values			
Bottom	140.7	147.7	163
Top	2140	2106	2471
LogEC50	619.9	277.8	141.7
HillSlope	-0.004115	-0.008516	-0.009063
EC50			
Std. Error			
Bottom	6.689	6.766	4.1
Top	4.714	9.491	16.13
LogEC50	1.387	1.16	1.002
HillSlope	0.00005101	0.0001749	0.0001428
95% Confidence Intervals			
Bottom	127.5 to 153.8	134.4 to 160.9	154.8 to 171.1
Top	2131 to 2149	2088 to 2125	2439 to 2503
LogEC50	617.2 to 622.6	275.6 to 280.1	139.7 to 143.7
HillSlope	-0.004215 to -0.004015	-0.008859 to -0.008173	-0.009346 to -0.008779
EC50		+infinity to +infinity	+infinity to +infinity
Goodness of Fit			
Degrees of Freedom	215	129	110
R square	0.9984	0.9974	0.9987
Absolute Sum of Squares	212220	224162	84338
Sy.x	31.42	41.69	27.69
Number of points			
Analyzed	219	133	114

Supplementary Code

SC1: "Z_stack_area_generation.m"

This MATLAB code analyzes the confocal image stack, and generates a .csv file with the fluorescence intensities. If you have multiple confocal image stack files within your folder, it will go through each image stack file in sequence and analyze them all.

CODE BEGINS

```
%Z_stack_area_generation: Analyzes Spheroid Z-stack Images
%
% This script requires:
%
% 1) folder with raw lsm files from confocal of microtissues (e.g.
% spheroids)
% 2) BioFormats for MATLAB
% (https://www.openmicroscopy.org/site/support/bio-formats5.1/users/matlab/)
%
%
% It outputs:
% 1)Mean fluorescence intensity of the spheroid when it is re-indexed as a
% series of concentric rings (intensity of each ring of that slice)
% 2)Total apparent volume
% 3)Maximum cross-sectional area
% 4)The number of 'imageable' slices
% 5)The mean aspect ratio (although not used for analysis in the paper)
% 6)Mean intensity of each shell, if the spheroid was re-indexed in 3D as a
% series of concentric shells.

clc
```

```
close all
clear all
matlab_folder = pwd;%remembers the current folder so it can come back to it

data_folder = uigetdir('','Select folder with image files');
cd(data_folder);
h = fspecial('disk', 20);
bins = 1:5:250;

files = dir('*.lsm');   %Creates a matrix consisting of the LSM files in your chosen folder
for lsmnum = 1:size(files,1)
    clear currimg mean_values total mean_positive avg
    stack = bfopen(files(lsmnum).name); %opens up the LSM files (one per for loop iteration)
    slicenum = size(stack{1,1},1);

    for p = 1:slicenum
        currimg(:,:,p)= stack{1,1}{p,1};
    end

    img2 = double(currimg(:,:,1:3:end)); %Extracts the channel that you want to analyze ('1' =>
Analyze Channel 1; '3' => Total number of channels in the z stack)
    rebuilt = zeros(size(img2)); %Pre-generating a matrix

    %Defines the region of the whole spheroid and pre-generates some
    %matrices
    blur_img = imfilter(img2,h);
    thresh_blur = multithresh(blur_img,3);
    mean_whole_img = mean(mean(mean(blur_img)));
    x = zeros(size(blur_img));
    positive_signal = zeros(size(blur_img,3),1);
```

```
    total_area_per_slice = zeros(size(blur_img,3),1);
    aspect_ratio = zeros(size(blur_img,3),1)';


    for z = 1:size(blur_img,3) %Slice-by-slice Analysis
        tempimg = img2(:,:,z);
        temprebuilt = rebuilt(:,:,z);
        threshold_spheroid  =  (thresh_blur/mean_whole_img)*mean(mean(tempimg));%normalize
the threshold for the slice
        %Image format changing (adjusts threshold based on 8 or 16-bit
        %image
        if isa(currimg, 'uint16')
            divide_by_this_number = 65535;
            if threshold_spheroid(1)>765

            else
                threshold_spheroid(1) = 765;
            end
        elseif isa(currimg, 'uint8')
            divide_by_this_number = 255;
            if threshold_spheroid(1)>2

            else
                threshold_spheroid(1) = 2;
            end
        end

        bright_pixel = blur_img(:,:,z)>threshold_spheroid(1);
        spheroid = bwconvhull(bright_pixel);
        distancemap = bwdist(~spheroid);
```

```
x(:,:,z) = spheroid;%designates x as the 3D matrix that describes the entire spheroid

%Finds the aspect ratio of each image slice. Not used in the final
%analysis, was originally conceived as a way of quantifying tissue
%deformity.
stats = regionprops(spheroid, 'MajorAxisLength', 'MinorAxisLength');
if isempty(stats) == 1 %Some image stacks are blank. This line prevents them from messing up the data.
    aspect_ratio(z) = 0;
else
    aspect_ratio(z) = stats.MajorAxisLength/stats.MinorAxisLength;
end

positive_signal(z) = sum(sum(tempimg>threshold_spheroid(1)));
total_area_per_slice(z) = sum(sum(x(:,:,z))); %Determines the number of pixels within the spheroid, whether fluorescent or not.

%Takes each image z-slice and re-indexes them as a series of
%concentric rings. Width of rings depends on 'bins' variable from
%line 9.
for  l = 1:(length(bins)-1)

    outside = distancemap<bins(l);
    inside = distancemap>bins(l+1);

    ring = ones(size(distancemap))-outside - inside;
    ring = ring>0;


    values = tempimg(ring);
```

```
1
2           positivevalues = values(values>threshold_spheroid(1));
3           if isempty(positivevalues)
4               positivevalues = 0;
5           end
6
7           mean_values(l,z) = mean(values);
8           total(l,z) = sum(values);
9
10          if sum(sum(ring))>0.5
11              mean_positive(l,z) = mean(positivevalues)+0.0001;
12          else
13              mean_positive(l,z) = mean(positivevalues);
14          end
15
16          temprebuilt(ring) =  mean_positive(l,z);
17
18      end
19      %end of Rings
20
21      rebuilt(:,:,z) = temprebuilt;
22  end
23
24  cd(matlab_folder);
25  [meanshell,~,~,~] = rings_3D(mean_positive);
26  cd(data_folder);
27
28  %This section of the code generates data and other outputs
29  volume = sum(sum(sum(x)));
30
```

```
    for H = 1:size(mean_positive,2)
        avg(H) = mean(nonzeros(mean_positive(:,H)));%mean fluorescence intensity
    end

    appended_data = [mean_positive; avg];
    T = table(appended_data);
    sumslices = sum(sum(x));%determines area of the spheroid (remember x is the spheroid)
    max_area = sumslices(sumslices==max(sumslices)); %finds the maximum area in a given STACK of images. This value is used to compare for volume shrinkage (divide this value in the 'before' image stack with the one obtained from the 'after' image stack
    positive_pixels_percentage = positive_signal./total_area_per_slice; %Determines the percentage of positive pixels for 'imageability' determination

    truncate_ppp = (total_area_per_slice>(max_area(1)*0.1)).*(positive_pixels_percentage);%discounts small area slices
    thresh_positive_pixels = 0.25; %The value 0.25 means that to be considered an image slice that is 'imageable', 25% of the total number of pixels within the spheroid must be positively fluorescent.
    num_positive_pixels_percentage = sum(truncate_ppp>thresh_positive_pixels);%counts number of acceptable slices


    %counts the number of rings in non-zero image slices
    num_rings = sum(mean_positive>0);
    num_rings_nonzero = num_rings(num_rings>0);
    num_rings_processed = num_rings_nonzero';
    %     R = table(num_rings_processed, positive_signal./total_area_per_slice);


    qualify_aspect_ratio = (truncate_ppp>thresh_positive_pixels).*(aspect_ratio)';%get rid of 0 aspect ratio values

    aspect_output = table(aspect_ratio', truncate_ppp, qualify_aspect_ratio);
```

```
mean_aspect_ratio =  mean(qualify_aspect_ratio(qualify_aspect_ratio>0));
V = table(volume,max_area(1),num_positive_pixels_percentage,mean_aspect_ratio);


IDShell = table(meanshell);

currfile = files(lsmnum).name;
foldername = currfile(1:(end-4));

mkdir(foldername);%makes a folder of the same name as the analyzed file
cd(foldername);


%Generates a file with the mean fluorescence intensity (each column is
%an image slice, and each entry of the column is the mean fluorescence
%intensity of that ring, starting from the outer-most ring of the
%image slice)
filepositive = strcat(currfile(1:(end-4)),'_mean_positive.csv');
writetable(T,filepositive);

%Generates a file with the total volume of the spheroid, the maximum
%cross-sectional area, the number of 'imageable' slices, and the mean
%aspect ratio
filevolume = strcat(currfile(1:(end-4)),'_volume.csv');
writetable(V,filevolume);

%Generates a file with the aspect ratio of each slice
fileaspect = strcat(currfile(1:(end-4)),'_aspect_ratio.csv');
writetable(aspect_output,fileaspect);

```

```
    %Generates a file that lists the shells and their mean fluorescence
    %intensity, starting from the outermost shell
    file_meanshell = strcat(currfile(1:(end-4)),'_meanshell.csv');
    writetable(IDShell, file_meanshell);

    for j = 1:size(blur_img,3)
        filerebuilt = strcat(currfile(1:(end-4)),'_rebuilt_',num2str(j),'.tif');
        imwrite(rebuilt(:,:,j)/divide_by_this_number,filerebuilt);
    end


    %end of outputs
    cd(data_folder);

end
cd(matlab_folder);
CODE ENDS
```

SC2: "Collect_Volume_and_Data.m"

This MATLAB code will go through your folder and retrieve all of the values from the previously generated .csv files. This was used to organize spheroids subjected to the same conditions into a table for ease of comparisons between replicates and against varying conditions.

```
CODE BEGINS

clc
close all
clear all
matlab_folder = pwd;%remembers the current folder so it can come back to it

```

```
experiment_folder = uigetdir('','Select top level folder with excel files');
cd(experiment_folder);
previous_data = {'Filename' 'Volume' 'Max Area' 'Imaging Depth' 'Average Aspect Ratio'};
subfolder_list = dir('*z stack*')
tmp = [subfolder_list.isdir];
for i = 1:length(tmp)
    cd(experiment_folder);
    if tmp(i) ==1
        lvl1dir = subfolder_list(i).name;
        cd(lvl1dir);
        recall_to_lvl1 = pwd;
        locate_excel_file = dir('*volume.csv');%finds files with 'volume.csv'
        filename = locate_excel_file.name; %finds the csv file
        find_excel_data = csvread(filename, 1, 0); %opens the csv file and grabs the numbers
        name_with_data = {filename, find_excel_data(1), find_excel_data(2), find_excel_data(3),
find_excel_data(4)}; %generates a set with filename and data
        appended_data = [previous_data; name_with_data]; %adds the name and numbers to the
previous data
        previous_data = appended_data;
    end
end



subfolder_list2 = dir('*gelation*')%finds the folder with 'gelation' in its name
tmp = [subfolder_list2.isdir];

for k = 1:length(tmp)
    cd(experiment_folder);
    if tmp(k) ==1
```

```
        lvl1dir = subfolder_list2(k).name;
        cd(lvl1dir);
        recall_to_lvl1 = pwd;
        sub_subfolder_list = dir('*z stack*');
        tmp2 = [sub_subfolder_list.isdir];
        for j = 1:length(tmp2)
            cd(recall_to_lvl1);
            if tmp2(j) ==1
                lvl2dir = sub_subfolder_list(j).name;
                cd(lvl2dir);
                recall_to_lvl2 = pwd;
                locate_excel_file = dir('*volume.csv');
%                   find_excel_data = readtable(locate_excel_file.name)
                filename = locate_excel_file.name; %finds the csv file
                find_excel_data = csvread(filename, 1, 0); %opens the csv file and grabs the numbers
                name_with_data = {filename, find_excel_data(1), find_excel_data(2),
find_excel_data(3), find_excel_data(4)}; %generates a set with filename and data
                appended_data = [previous_data; name_with_data]; %adds the name and numbers to the
previous data
                previous_data = appended_data;
            end
        end
    end
end
cd(experiment_folder);
collated_filename = strcat(experiment_folder(66:end-4),'_Collated Data.xlsx');
tabulate = table(appended_data);
writetable(tabulate, collated_filename);

CODE ENDS
```

SC3: "connectivity_analysis.m"

This MATLAB code analyzes your vasculature channel, applies a local threshold to determine whether a pixel is positive for signal, then looks at its neighbours to construct the vasculature networks, labels it, then measures volume, size, and mean fluorescence intensity from the original image in the regions defined by the labels.

CODE BEGINS:

```
% CONNECTIVITY_ANALYSIS Analyses the blood vessel network from mouse islets
%
% This script requires:
%
% 1) folder with raw lsm files from confocal of mouse islets.
% 2) dipimage toolbox: http://www.diplib.org/dipimage.
%
%
% It outputs:
% 1) label matrix of blood vessel segments (each segment labelled with a
% unique integer). 'filename.lsm' becomes 'filename.tif'
%
% 2) Outputs of analysis saved as 'filename_connectivity_results.csv' with
% various statistics describing vessel network.


clc
close all
clear all
```

```
matlab_folder = pwd;%remembers the current folder so it can come back to it

data_folder = uigetdir('','Select folder with image files');
cd(data_folder);


files = dir('*.lsm');
    %%% Uncomment this line to run in multithread
% parfor(lsmnum = 1:size(files,1),16)
    %%% Uncomment this line to run in single thread
for lsmnum = 1:size(files,1)

    oldname = files(lsmnum).name;
    filepositive = strcat(oldname(1:(end-4)),'_connectivity_results.csv');

    if exist(filepositive)==0 % only runs if there isn't already a results file

        %%% Read file and metadata
        stack = bfopen(files(lsmnum).name);
        slicenum = size(stack{1,1},1);

        omeMeta = stack{1, 4};
        stackSizeX = omeMeta.getPixelsSizeX(0).getValue(); % image width, pixels
        stackSizeY = omeMeta.getPixelsSizeY(0).getValue(); % image height, pixels
        stackSizeZ = omeMeta.getPixelsSizeZ(0).getValue(); % number of Z slices
        numchannels = size(stack{1,1},1)/stackSizeZ;

        metadata = stack{1, 2};
        physSizeX = metadata.get('VoxelSizeX');
        physSizeY = metadata.get('VoxelSizeY');
```

```
physSizeZ = 1.45*metadata.get('VoxelSizeZ');
%Note: physSizeZ is multiplied by 1.45 to account for the
%refractive index of RIMS compared to air. Images are taken with an
%air objective which artificially reduces the observed z-thickness.


%%% Specifies size of filter for local threshold
h = fspecial('disk', round(stackSizeX/10));


%%% Specifies which channel was used to image vasculature for
%%% multichannel images.
if oldname(1:3)=='Aug'
    ves_ch = 1;
elseif numchannels==1
    ves_ch = 1;
else
    ves_ch = 2;
end


%%% Preprocessing of image
currimg= zeros([size(stack{1,1}{1,1}),stackSizeZ]);
for p = 1:slicenum
    currimg(:,:,p)= stack{1,1}{p,1};
end

%%% Extract vessel channel from the loaded image
img = double(currimg(:,:,ves_ch:numchannels:end));

```

```
%%% Blurred vessel image used to determine local threshold
blur_img = imfilter(img,h);


%%% Create a contrast normalized image for local thresholding
cont_norm = zeros([size(stack{1,1}{1,1}),stackSizeZ]);
for i = 1:size(img,3)
    cont_norm(:,:,i) = img(:,:,i)./(blur_img(:,:,i) + 0.2*(max(img(:))));
end


%%% Threshold the contrast normalized image and cleanup
[~,thresh_blur] = threshold(cont_norm,'triangle');
thresholded_image = cont_norm>thresh_blur;
str_disk = strel('disk',7);
closed_image = imclose(thresholded_image,str_disk);


%%% Each connected region of blood vessels is assigned a label and
%%% size, mean signal and surface area of these networks are
%%% extracted
labeled_image = label(closed_image, 2, 10,0);
connectivity_stats = measure(labeled_image, img, ({'size' 'mean' 'SurfaceArea'}), [], 2,
10,0);


%%% Define the entire region of the islet and calculate the
%%% distance transform inwards to the islet centre
islet_outline = hull(opening(mat2im(img>0)),0);
islet = fillholes(dilation(islet_outline,3));
```

```
        islet_padded = cat(3,islet(:,:,1)>10,islet, islet(:,:,1)>10);
        islet_padded = erosion(islet_padded,4);
        distancemap_padded = dip_edt(islet_padded, [physSizeX physSizeY physSizeZ], 1, 'fast');
        distancemap = distancemap_padded(:,:,1:(end-1));
        minimumradius = max(distancemap);

        %%% Trims away vessel networks so that only vessels in the core of
        %%% the islet are retained
        trimmed_label_image = labeled_image;
        trimmed_label_image(distancemap<(0.5*minimumradius))=0;


        %%% Region recognized as the core of the islet.
        trimmed_islet = islet;
        trimmed_islet(distancemap<(0.5*minimumradius))=0;

        %%% Collect size, mean signal and surface area of the trimmed
        %%% vessel networks (trimmed meaning only within the core of the
        %%% islet)
        connectivity_stats_trimmed = measure(trimmed_label_image, img,({'size' 'mean'
'SurfaceArea'}), [], 2, 10,0);

        %%% Calculates the average distance between any given pixel within
        %%% the spheroid and the nearest blood vessel
        vesseldist = dip_edt(~thresholded_image, [physSizeX physSizeY physSizeZ], 1, 'fast');
        meandist2vessel = mean(vesseldist(islet & ~thresholded_image));

        %%% Debugging code prevents errors in cases where no vessel
        %%% networks are detected
        if size(connectivity_stats,1)==0
```

```
    M = zeros(1,3);
    N = zeros(1,3);
    Intensity_whole = zeros(1,3);
    Surface_whole = zeros(1,3);
else
    M = connectivity_stats.size;
    N = connectivity_stats.ID;
    Intensity_whole = connectivity_stats.mean;
    Surface_whole = connectivity_stats.SurfaceArea;
end

if size(connectivity_stats_trimmed,1)==0
    P = zeros(1,3);
    Q = zeros(1,3);
    Intensity_CORE = zeros(1,3);
    Surface_core = zeros(1,3);
else

    P = connectivity_stats_trimmed.size;
    Q = connectivity_stats_trimmed.ID;
    Intensity_CORE = connectivity_stats_trimmed.mean;
    Surface_core = connectivity_stats_trimmed.SurfaceArea;
end

%%% Processing of statistics for export

unsorted_results_matrix = [N;M;Intensity_whole]';
sorted_matrix = sortrows(unsorted_results_matrix, -2);
unsorted_trim = [P;Q;Intensity_CORE]';
ID_max = sorted_matrix(1,1);
```

```
ID_top3 = sorted_matrix(1:3,1);
volume_member = ismember(unsorted_trim(:,2), ID_top3);
volume_member_max = ismember(unsorted_trim(:,2), ID_max);
mean_member = ismember(unsorted_trim(:,2), ID_top3);
mean_member_max = ismember(unsorted_trim(:,2), ID_max);
sum_trimmed_top3 =  sum(unsorted_trim(volume_member, 1));
trimmed_max =  (unsorted_trim(volume_member_max, 1));

if isempty(trimmed_max)
   trimmed_max = 0;
end

if isempty(sum_trimmed_top3)
   sum_trimmed_top3 = 0;
end

Intensity_top3 =  mean(unsorted_trim(mean_member, 3))+0;
Intensity_max =  mean(unsorted_trim(mean_member_max, 3))+0;

if isempty(Intensity_top3)
   Intensity_top3 = 0;
end
if isempty(Intensity_max)
   Intensity_max = 0;
end
sortedM = sort(M, 'descend');
sorted_surfaceareas = sort(Surface_whole, 'descend');
sorted_surfaceareas_core = sort(Surface_core, 'descend');
sorted_surfaceareas_core = padarray(sorted_surfaceareas_core,[0 3], 'post');

```

```
%%% Final output statistics
% This contains individual stats about vascular connectivity
O = zeros(size(N));

% Proportion of vessels in main network
O(1) = max(M)/sum(M);

% Proportion of vessels in top 3 networks
O(2) = sum(sortedM(1:3))/sum(M);

% Proportion of vessels in spheroid core that are in main network
O(3) = trimmed_max/sum(P);

% Proportion of vessels in spheroid core that are in top 3 networks
O(4) = sum_trimmed_top3/sum(P);

% Proportion of top 3 networks in spheroid core
O(5) = sum_trimmed_top3/sum(sortedM(1:3));

% Proportion of spheroid in spheroid core
O(6) = sum(trimmed_islet)/sum(islet);

% Mean distance to vessel inside spheroid
O(7) = meandist2vessel;

% Proportion of Mean intensities of the main vessel network with the core compared to whole tissue
O(8) = Intensity_max/sorted_matrix(1,3);

```

```
% Proportion of Mean intensities of the top 3 networks with the core compared to whole tissue
O(9) = Intensity_top3/mean(sorted_matrix(1:3,3));

% Proportion of islet volume occupied by main vessel network
O(10) = max(M)/sum(islet);

% Proportion of islet volume occupied by top 3 vessel networks
O(11) = sum(sortedM(1:3))/sum(islet);

% Proportion of islet core occupied by main network
O(12) = trimmed_max/sum(trimmed_islet);

% Proportion of islet core occupied by top 3 vessel networks
O(13) = sum_trimmed_top3/sum(trimmed_islet);

% Ratio of surface area of top 3 networks compared to islet volume
O(14) = sum(sorted_surfaceareas(1:3))/sum(islet);

% Ratio of surface area of top 3 networks inside islet core to
% volume of islet core
O(15) = sum(sorted_surfaceareas_core(1:3))/sum(trimmed_islet);

% Combined matrix of all statistics for output
results_matrix = [N;M;O]';


% Set filename and save label matrix of blood vessels.
labeled_image_16bit = uint16(im2mat(labeled_image));
tif_name = strcat(oldname(1:end-4),'.tif');
```

```
        for p = 1:size(img,3)
            imwrite(labeled_image_16bit(:,:,p),tif_name, 'WriteMode','append');
        end


        % Write out results of connectivity analysis
        T = table(results_matrix);
        writetable(T,filepositive);

    else
        %skips analysis if it finds the results file from a previous run
        continue;
    end
end
cd(matlab_folder);
CODE ENDS
```