Collective and Individual Invasion of Multicellular Spheroids in 3D Matrix

By

Alejandro Marzoratti

ScM, Brown University, 2023

Thesis

Submitted in partial fulfillment of the requirements for the Degree of Master of Science in the Graduate Program of Biomedical Engineering at Brown University

PROVIDENCE, RHODE ISLAND

May 2023

AUTHORIZATION TO LEND AND REPRODUCE THE THESIS

As the sole author of this thesis, I authorize Brown University to lend it to other institutions for the purpose of scholarly research.

Date 4/25/23 Alejandro Marzoratti

Alejandro Marzoratti, Author

I further authorize Brown University to reproduce this thesis by photocopying or other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Date 4/25/23 Alejandro Marzoratti

Alejandro Marzoratti, Author

This thesis by Alejandro Marzoratti is accepted in its present form by the Center of Biomedical Engineering as satisfying the thesis requirement for the degree of Master of Science.

Date 4/25/23

Dr. Ian Wong, Advisor

Recommended to the Graduate Council

Date 4/25/23

Dr. Vikas Srivastava, Reader

Date 4/25/23

Dr. Sean Lawler, Reader

Approved by the Graduate Council

Date 4/28/2023

Dr. Marissa Gray, Biomedical Engineering Master's Program Director

Acknowledgements

This thesis is the product of an effort that started almost 2 years ago. Several people have contributed to the development and refinement in one way or another. I would like to take the opportunity to thank them in this preface.

I would like to express my deepest appreciation to my advisor Dr. Ian Wong. Your encouragement and guidance throughout my time here at Brown has been invaluable. I also could not have undertaken this journey without the support and encouragement of my lab mate Alex Hruska. Working alongside you in the lab has been an absolute pleasure and thank you for all that you've taught me. I'm extremely grateful to Dr. Vikas Srivastava and Dr. Sean Lawler for serving on my committee. Research reported in this thesis was supported by the National Institute of Health under award number R01GM140108.

Thanks should also go to the Brown University Center for Biomedical Engineering and the Office of Graduate and Postdoctoral Studies for being such incredible resources. I am also thankful to Dr. Jin Yang and Dr. Christian Franck for providing essential support in particle tracking software. I would be remiss in not mentioning the other past and present Wong Lab members that have provided practical support and inspiration in this project: Dr. Amanda Khoo, Dr. Catherine Machnicki, Dr. Blanche Ip, Alex Markoski, and Lily Cordner. Lastly, I'd like to mention my family and friends that have given me the unwavering emotional support that this endeavor would have been impossible without.

Table of Contents

Table of Figures

Table of Tables

Abstract of Collective and Individual Invasion of Multicellular Spheroids in 3D Matrix, by Alejandro Marzoratti, ScM, Brown University, May 2023

The epithelial-mesenchymal transition (EMT) is associated with solid tumor dissemination and metastasis. Directed migration of tumor cells into the surrounding stroma occurs through both collective and individual phenotypes, mediated by varying cell-cell and cell-matrix adhesions. Such invasive phenotypes may utilize spatially and temporally coordinated tractions to deform and remodel extracellular matrix. However, the mechanistic role of EMT in collective and individual invasion remains poorly understood. We utilized the hanging drop method to aggregate mammary epithelial (MCF-10A) cells, with an inducible gene expression construct that induces EMT master regulator Snail-1 in response to stimulation of the estrogen receptor with tamoxifen. Spheroids were transferred into 3D matrix (e.g., bovine collagen I) embedded with tracer particles to create a spheroid invasion model. Over several days, we observed and imaged spheroid invasion profiles using spinning disk confocal microscopy with environmental control, and 3D single cell and particle tracking was performed using custom MATLAB code. Our findings revealed that induction of EMT in MCF-10A spheroids via Snail-1 upregulation decreased cell-cell adhesions, resulting in the loss of leader-follower cell behavior and collective invasion. EMT induction was also associated with a loss of intra-spheroid coordinated cell motion as well as spheroid generated matrix deformations. The use of confocal imaging and 3D cell culture techniques allowed for high-resolution characterization of collective and individual cell dynamics. These findings have intriguing implications for cancer research and therapy development, including EMT as a potential therapeutic target.

Introduction

Cellular mechanotransduction, the process by which cells sense and respond to mechanical cues from their environment, is a fundamental aspect of tissue development, homeostasis, and disease progression (1). The cytoskeleton, composed of various structural proteins, including intermediate filaments such as vimentin, plays a crucial role in cellular mechanotransduction. Vimentin has been shown to influence cellular mechanical properties and motility, and its dysregulation has been implicated in cancer metastasis and tissue remodeling (2). However, the mechanisms through which vimentin affects multicellular tractions and remodeling of the extracellular matrix (ECM) remain poorly understood.

In recent years, 3D cell culture models have gained increasing attention as they more accurately recapitulate the complexity of the in vivo tissue microenvironment compared to traditional 2D cell culture. Among these models, multicellular spheroids have emerged as a powerful tool to study collective behaviors and tissue remodeling in a controlled and biomimetic context (3). In this model, dispersed single cells are aggregated into spheroids that are then embedded within a 3D matrix composed of ECM proteins, such as collagen, to mimic the in vivo tissue architecture.

One intriguing aspect of multicellular behaviors in a 3D microenvironment is the leader-follower cell mechanism, which refers to the coordinated movement of cells within a group, where some cells act as leader cells and guide the collective migration of follower cells (4). This phenomenon has been observed in various physiological and pathological processes, including tissue morphogenesis, wound healing, and cancer invasion. The leader-follower cell mechanism is thought to be regulated by a combination of mechanical and biomechanical cues, and

understanding the role of vimentin in this process may provide insights into the mechanisms underlying multicellular tractions and remodeling (4).

Another important cellular process that is closely related to tissue morphogenesis and remodeling is the epithelial-mesenchymal transition (EMT), a fundamental cellular program that drives the conversion of epithelial cells into mesenchymal-like cells with enhanced migratory and invasive properties (5). EMT is implicated in various physiological and pathological processes, including embryogenesis, tissue repair, and cancer metastasis. Emerging evidence suggests that vimentin plays a key role in EMT, as it is often upregulated during this process and has been implicated in promoting the migratory and invasive properties of mesenchymal-like cells (6). However, the precise mechanisms through which vimentin influences EMT and its impact on multicellular tractions and remodeling in a 3D environment remain poorly understood.

In light of these knowledge gaps, the aim of this thesis project is to investigate how EMT affects multicellular tractions and remodeling in a 3D biomimetic matrix using an embedded spheroid invasion model. We explore the role of EMT in regulating cellular mechanical properties, leader-follower cell behaviors, and matrix deformation. Through a combination of cell culture techniques, biophysical measurements, and imaging analysis, we seek to elucidate the intricate interplay between EMT, multicellular tractions, and tissue remodeling in a 3D biomimetic context.

The findings of this thesis project may contribute to our understanding of the role of vimentin in cellular mechanotransduction, leader-follower cell behaviors, and EMT-related processes, and shed light on the complex mechanisms underlying tissue morphogenesis and remodeling in the tumor microenvironment. Moreover, the 3D embedded spheroid model employed in this study may have broader implications for tissue engineering and regenerative

medicine, as it provides a valuable tool for investigating cellular behaviors and tissue remodeling in a more physiologically relevant context.

Methods and Materials

Cell Culture. Human mammary epithelial cells (MCF-10A) stably transfected with a 4-hydroxytamoxifen (OHT) inducible estrogen receptor-fused Snail-1^{6SA} retroviral construct were a generous gift from D.A. Haber (Massachusetts General Hospital) (7). To induce EMT through Snail-1 gene expression, MCF-10A cells were cultured in 500 nM OHT for at least 6 days at low density to prevent spontaneous reversion to the epithelial phenotype. Cells were examined for elongated, spindle-like morphologies to verify EMT induction. MCF-10A cells were also cultured in DMSO as a vehicle control where Snail-1 expression was expected to be minimal. One variant of this inducible MCF-10A cell line also overexpressed fluorescent proteins in the cytoplasm (GFP) and nucleus (mCherry-H2B) for ease of cell imaging, while the other was uncolored.

Human mammary epithelial cells (MCF-10A) stably transfected with fluorescent proteins in the nucleus (mCherry-H2B) were a generous gift from M.R. Ng and J. Brugge (Harvard Medical School). These MCF-10A cells exhibited a “wild type” phenotype and did not have inducible EMT/Snail-1 gene expression. Detailed descriptions of making media as well as performing routine cell culture are reported in Appendix A.

Spheroid Generation. A hanging drop array system was used to generate uninduced epithelial control and EMT-induced mesenchymal MCF-10A 3D spheroid cultures.

Pass Cells and Prepare Hanging Drop Spheroids. To prepare hanging drop spheroids composed of 500 cells, uninduced and EMT-induced MCF-10A cells were passaged and counted as

previously described (Appendix A) and diluted to 500,000 cell/mL. To make 1,300 μL of spheroid suspension, 978 μL of warmed (37℃) growth medium (Appendix A), 130 μL of syringe sterile-filtered (Fisher 09-719C) 100% Standard Fetal Bovine Serum (10% final concentration) (FBS)(Cytiva SH30088.03HI), 130 μL of 10 mg/mL methylcellulose solution (1 mg/mL final concentration)(Appendix A), and 61.9 μL of 500,000 cell/mL (500 cells / 23 μL final concentration) single cell suspension were added to a 1.5 mL round-bottom microcentrifuge tube (Fisher 21P1017100) with gentle vortexing after adding each component. To prepare mosaic spheroids, varying ratios of WT MCF-10A and OHT-induced MCF-10A single cell suspension were combined to create 61.9 μL of mosaic single cell suspension. For example, to create spheroids with 250 WT MCF-10A cells and 250 OHT-induced MCF-10A cells (50% uninduced, 50% induced), 30.95 μL of each 500,000 cell/mL single cell suspension was combined to form the 61.9 μL of mosaic single cell suspension within the spheroid suspension. The spheroid suspension was then transferred to a sterile basin and a multi-channel pipette was used to make a 7x7 grid of 23 μL drops on the inside lid of a 100 mm/15 mm Sterile Petri Dish (EZ BioResearch) such that once you put the lid back on, the droplets are hanging downwards inside the dish. After carefully adding 7 mL of sterile distilled (DI) water with 1% Pen-Strep to the bottom of the Petri dish to keep the droplets hydrated, the Petri dish with hanging drops was delicately placed in the incubator (NU-8600) (37℃ and 5% CO_2) to compact. Proper aggregation was verified microscopically after 24 hours by carefully transferring the hanging drop lid to an empty Petri dish bottom and gently inverting. If cells were aggregated into a spheroid, the hanging drop lid was carefully transferred to its original bottom (with water reservoir) and returned to the incubator. Following 48 hours of compaction, spheroids were visible by eye as small white imperfections in the droplet and ready to use (8). For larger experiments, it was

necessary to prepare multiple Petri dishes to have enough spheroids for embedding into 3D collagen matrices.

Spheroid Embedding into 3D Collagen Matrices. Uninduced epithelial control and EMT-induced mesenchymal MCF-10A spheroids were embedded in 6.0 mg/mL Type 1 Bovine Collagen (Col-1) (Advanced Biomatrix 5133) based matrices within a Clear Bottom, TC Treated, 96-Well Imaging Plate (PerkinElmer 8793-22141).

Sacrificial Sample of Type 1 Bovine Collagen. Fresh batches of 10 mg/mL Col-1 were ordered approximately every 4 months and stored in the refrigerator (4℃) surrounded by cold packs. Batches were received as soluble atelocollagen in 0.01 N HCL with a pH of ~ 2 (7). When new batches arrived, a sacrificial gel sample with no cells was created to determine how much acid or base will bring the solution to a cell viable pH. To do so, a 1000 μL aliquot of 7.5 mg/mL stock Col-1 was created just as if it were to be used for embedding spheroids. Some batches arrived at 10.2 mg/mL, and the amount of sterile DI water was adjusted to conserve the 6 mg/mL Col-1 concentration in the final gel. First, we took a large cold block out of the freezer (-20℃) and let it sit at room temperature for 15 minutes. While the cold block warmed, we labeled 1X 1.5 mL round-bottom microcentrifuge tube “Sacrificial pH Adjust” and 1X 0.5 mL round-bottom microcentrifuge tube “125mM HEPES”. The labeled tubes were then placed in the ice block to cool. After 15 minutes, 110 μL of DI water, 44 μL of 10x PBS (Sigma D1408), and 10 μL of 1 N Sodium Hydroxide (NaOH) (Fisher S320-1) solution in DI water were added to the chilled 1.5 mL tube labeled “Sacrificial pH Adjust” and vortexed. Subsequently, 437.5 μL of DI water and 62.5 μL of 1 M HEPES, Free Acid Solution (HEPES) (125 mM final concentration) (Cytiva

SH3023701) were added to the chilled 0.5 mL tube labeled “125 mM HEPES” and vortexed. Col-1 can polymerize rapidly in response to heat, it was important that the following steps be done quickly. Before pipetting Col-1 solutions, triturate cold DI water 3 to 4 times to chill the pipette tip. While keeping Col-1 solutions on a cold block at all times, 736 μL of 10.2 mg/mL stock Col-1 was added to the mixture with NaOH and gently triturated to mix while carefully avoiding bubbles. When bubbles occurred, they were removed by gently flicking the 1.5 ml 7.5 mg/mL Col-1 stock tube. After allowing the solution to homogenize for 3 minutes, the pH was measured using a calibrated Seven Compact pH meter S220 (Mettler Toledo). If the pH was in the 7.35 +/- 0.2 range, no adjustment was necessary. If below the range, the pH was adjusted by adding NaOH in 0.5 μL intervals, testing every 3 minutes until physiological pH was achieved. The pH meter probe was rinsed with DI water and stored in Orion pH Electrode Storage Solution (Fisher 910060) between each measurement. If above the range, the process was restarted with a lower starting volume of NaOH. If excess NaOH was needed to adjust the gel pH, the gel component ratios were adjusted by subtracting the excess NaOH volume from the volume of DI water to conserve the 1000 μL total volume.

Prepare 1000 μL Aliquot of 7.5mg/mL Stock Col-1 Solution and Coat Wells in 0.05 mg/mL Col-1. To prepare 1000 μL of 7.5 mg/mL stock Col-1 solution, we took a large cold block out of the freezer (-20℃) and let it sit at room temperature for 15 minutes. While the cold block warmed, we labeled 1X 2.0 mL microcentrifuge tube “0.05 mg/mL Coat”, 1X 1.5 mL round-bottom microcentrifuge tube “7.5 mg/mL Col-1”, 1X 0.5 mL round-bottom microcentrifuge tube “125mM HEPES”, and 2x 0.5 mL round-bottom microcentrifuge tubes “6.0 mg/mL Col-1 TFM”. The labeled tubes were then placed in the ice block to cool. After 15 minutes, 110 μL of

DI water, 44 µL of 10x PBS (Sigma D1408), and 10 µL of 1 N NaOH solution in DI water were added to the chilled 1.5 mL tube labeled "7.5 mg/mL Col-1" and vortexed. Subsequently, 437.5 µL of DI water and 62.5 µL of 1 M HEPES (125 mM final concentration) were added to the chilled 0.5 mL tube labeled "125 mM HEPES" and vortexed. Col-1 can polymerize rapidly in response to heat, it was important that the following steps be done quickly. Before pipetting Col-1 solutions, triturate cold DI water 3 to 4 times to chill the pipette tip. While keeping Col-1 solutions on a cold block at all times, 736 µL of 10.2 mg/mL stock Col-1 was added to the mixture with NaOH and gently triturated to mix while carefully avoiding bubbles. When bubbles occurred, they were removed by gently flicking the 1.5 ml 7.5 mg/mL Col-1 stock tube. Subsequently, 100 µL of 125 mM HEPES solution (12.5 mM final concentration in Col-1 solution) was added to the Col-1 solution then gently triturated and stirred to homogenize the entire Col-1 solution. For large experiments, it was necessary to prepare multiple 1000 µL aliquots of 7.5 mg/mL stock Col-1 solution to have enough for making 6.0 mg/mL Col-1 solutions. 7.5 mg/mL Col-1 stock solutions were then set aside on the ice block and placed in the refrigerator (4°C) for 1 hour to pre-polymerize. The spheroid washing step was planned to finish at the same time as the 1 hour pre-polymerization step.

The chilled 2.0 mL tube labeled "0.05 mg/mL Coat" was then taken from the cold rack and returned to the tissue culture hood. Next, 1990.2 µL of 1X PBS and 9.8 µL of stock Col-1 were added and vortexed (0.05 mg/mL final concentration in coat). To keep collagen gels from disengaging from the wells and prevent lifting throughout the duration of the experiment, 150 µL of 0.05 mg/mL Col-1 coat was added to each well and rocked for 1 hour at room temperature. After rocking for 1 hour, excess Col-1 coat was aspirated. Thorough aspiration was confirmed visually to prevent bubble formation during spheroid embedding due to residual Col-1 coat. For

larger experiments, it was necessary to make multiple 2.0 mL tubes of 0.05 mg/mL Col-1 coat to have enough to coat all wells. To stabilize the temperature in the inner Col-1 solution-filled wells during polymerization and imaging, 300 μL of growth medium or DI water was put in each of the outer walls of the 96 well plate to be left for the duration of the experiment.

Spheroid Wash. Uninduced epithelial control and EMT-induced MCF-10A spheroids were washed 2 times to remove residual methylcellulose. To do so, the Petri dish with compacted hanging drop spheroids was taken from the incubator and the lid carefully inverted to not disturb the integrity of the drops. Using a P100 pipette (ErgoOne) set to 100 μL, all 49 spheroids were carefully harvested from their respective drops and the spheroid suspension transferred to a 0.5 mL tube. 400 μL of warmed (37℃) growth medium was then added and the process was repeated for the remaining Petri dishes prepared for the experiment. After allowing 3 minutes for the spheroids to sink to the bottom of the tube, carefully remove 400 μL of growth medium while being cautious as to not aspirate any spheroids and discard the waste. Another 400 μL of warmed (37℃) growth medium was added and the process was repeated once more for all spheroid suspension tubes, leaving the spheroids suspended in 100 μL of growth medium at the end.

Spheroid Embedding. Uninduced epithelial control and EMT-induced MCF-10A spheroids were embedded with 1 *u*m diameter Fluorescent Microspheres (FMS)(1.401e+10 microspheres/mL)(Bangs Laboratories FCFFR006) for traction force microscopy (TFM). To prepare a 2 mg/mL concentration of FMS, 24 μL of DI water and 6 μL of FMS stock were added to a 0.2 mL PCR Tube (Genesee 24-154A), vortexed, and covered with tinfoil to avoid photobleaching. Then, the 2 mg/mL FMS solution was sonicated for 10 minutes in a Model 75D

Ultrasonic Cleaner (VWR) and vortexed again before use. At this time, the 7.5 mg/mL Col-1 stock solution had pre-polymerized for 1 hour and optimal pH (~7.4) was confirmed by pipetting 10 μL of 7.5 mg/mL Col-1 stock solution on a Plastic pH Indicator Strip (Hydrion 228720V). If more than one 1000 μL of 7.5 mg/mL Col-1 stock solution was prepared, the pH of each was evaluated separately. Thereafter, the 96-well plate was retrieved from the rocker and the excess 0.05 mg/mL Col-1 coat was aspirated ensuring no droplets were left behind to avoid bubbles underneath the Col-1 matrix. Then, 79 μL of gently triturated spheroid suspension was transferred to the chilled 0.5 mL tubes labeled "6.0 mg/mL Col-1 TFM". Subsequently, 11.25 μL of 2 mg/mL FMS solution was added and gently mixed to distribute the particles. This process was repeated for the remaining spheroid suspensions and "6.0 mg/mL Col-1 TFM" chilled 0.5 mL tubes. Once the FMS solution had been added to all of the spheroid suspensions, FMS and spheroids were again redistributed using gentle mixing. Immediately after, 320 μL of 7.5 mg/mL Col-1 stock solution was added using a chilled pipette tip and the 6.0 mg/mL Col-TFM solution was very gently triturated and stirred before plating 95 μL of solution per well for 3 wells. Col-1 solutions were very viscous, so the solution was allowed to pool at the pipette tip and ejected at least 3 times to minimize loss. When plating, it was important to evenly distribute the solution on the bottom of the well while avoiding bubbles and removing them by aspiration when they occur. If there is excess 6.0 mg/mL Col-TFM solution after 3 wells have been seeded, disperse the remaining solution evenly among the 3 plated wells. The plating process was repeated for the remaining spheroid suspensions. Once all 7.5mg/ml Col-1 solutions were plated, they were

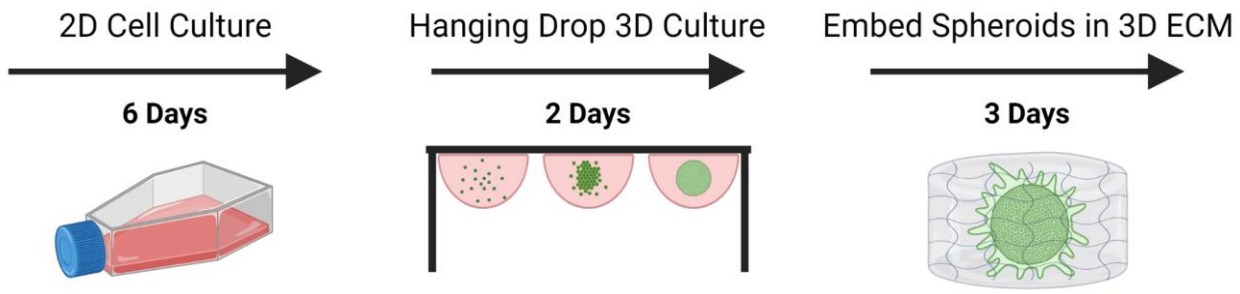

Fig. 1 Experimental Timeline. Dispersions of uninduced epithelial control and EMT-induced mesenchymal MCF-10A cells were dispensed into hanging drop suspensions. Typically, cells assemble into compact multicellular spheroids over 48 h. Multicellular spheroids were then transferred to 3D matrix (e.g., bovine collagen I, Matrigel etc.) labeled with far red fluorescent microparticles (1 μm diameter) and would invade over several days.

immediately cured in the incubator for 1.5 hours (37°C). Once cured, 150 μL of growth medium was placed on top of the cured collagen with either 1:2000 DMSO for uninduced MCF-10A spheroids or 1:2000 OHT for EMT-induced MCF-10A spheroids. At this point, the embedded spheroids are ready for imaging or to be incubated for imaging later.

Time-lapse Imaging of Spheroid Invasion. Time lapse images of uninduced epithelial control and EMT-induced mesenchymal MCF-10A spheroids were taken every 30 minutes for up to 72 hours using a Nikon Eclipse Ti fluorescence microscope with a spinning-disk confocal head (Crest Optics X-light V2), a light-guide coupled solid-state illumination system (Lumencor Spectra X), scientific complementary metal-oxide-semiconductor camera (sCMOS)(Andor Neo), and 20x Plan Apo Fluor objective (NA 0.75). Embedded spheroids were preserved in a humidified environmental chamber at 37°C and 5% CO_2. Automated image acquisition with z steps of 2.5 μm and 150 μm range was done using NIS Elements. Further detail of imaging properties and capturing techniques are reported in Appendix A. Spheroids embedded ~3900 μm above the objective lens allowed for the use of Perfect Focus (PFS) and were considered optimal

for acquiring high-quality time lapse images. Spheroids embedded too close to bubbles, the well wall, or the well bottom responded to non-ECM material stiffness and were therefore not imaged for analysis.

Fixing and Staining of Collagen Matrices. After 72 hours of time-lapse imaging, immunofluorescence was used to confirm and characterize vimentin expression within multicellular spheroids (9). Detailed descriptions of the collagen matrix fixation, permeabilization, and blocking methods are reported in Appendix A.

Primary Antibody. To detect endogenous levels of total vimentin protein, we stained the matrix only with Vimentin (D21H3) XP Rabbit mAB (rabbit anti-vimentin) (Cell Signaling Technology 5741S). To do so, a 0.5% (v/v) of goat serum in 1X PBS was prepared. Subsequently, a 1:200 (v/v) dilution of the rabbit anti-vimentin in 0.5% goat serum was created. After aspirating the PBS from each well, 100 μL of rabbit anti-vimentin solution was added and the plate wrapped in parafilm and returned to the rocker plate for 2 nights. Then, a 0.01% (v/v) solution of Tween (Sigma 9005-64-5) in 1X PBS was prepared. After aspirating the rabbit anti-vimentin solution, each well was washed 3 times with 200 μL of 0.01% TWEEN with the plate being returned to the rocker table for several hours between washes. A fourth wash was performed with only 1X PBS to prevent TWEEN from interfering with subsequent Hoechst 33258 staining.

Secondary Antibody. To aid in the detection of the target antigen vimentin, we used Anti-rabbit IgG (H+L) Fragment $(ab')_2$ (Alexa Fluor 647 Conjugate) (anti-rabbit) (Cell Signaling Technology 4414S) as a secondary antibody. When staining the ECM with anti-rabbit, all steps

were performed in the dark to prevent fluorophore photobleaching. First, a 0.5% (v/v) solution of goat serum in 1X PBS was prepared for use in creating a 1:200 (v/v) dilution of the anti-rabbit in 0.5% goat serum. After aspirating the PBS from each well, 100 μL of anti-rabbit solution was added. The plate was then wrapped in parafilm and placed on the rocker table for 2 nights. After aspirating the anti-rabbit solution, each well was washed 2 times with 200 μL of 1X PBS with the plate being returned to the rocker table for 8 hours after the first wash and overnight for the second.

Hoechst 33342. To stain the cell nuclei within the ECM, we used the cell-permeable benzimidazole dye Hoechst 33342 (Hoechst) (Cayman Chemical Company 15547). In addition, we used Alexa Fluor 488 Phalloidin (Phalloidin) (Cell Signaling Technology 8878S) to stain the cell cytoskeleton within the ECM. To do so, a solution of 1:500 Hoechst and 1:60 Phalloidin in 1X PBS solution was initially prepared. After aspirating the PBS from each well, 100 μL of the Hoechst/Phalloidin solution was added and the entire plate wrapped in parafilm and aluminum foil. After being placed on the rocker table overnight, the Hoechst/Phalloidin solution was aspirated and each well washed 2 times with 200 μL of 1X PBS with the plate being returned to the ricker for 1 hour between washes. At this point, the hydrogels were fixed, stained, and ready for fluorescence imaging. When not being imaged immediately, hydrogels were preserved in a 0.2% (w/v) Sodium Azide in 1x PBS solution, wrapped in parafilm and aluminum foil, and stored in the refrigerator (4°C) until use.

Quantitative Analysis of Spheroid Invasion. NIH-Elements annotations and measurements software was used to gather strand length, maximum strand width, strand base width, and

invasion area data as well as to count strands and single detached cells. Starting at the lowest z-plane, the number of single-detached cells and protrusive strands was counted, and each protrusive strand was labeled. Then, again starting at the lowest z-plane, the max strand width and length of each labeled strand was measured and recorded. Strands were defined as multicellular protrusions that had at least 2 nucleus signals. Single-cell protrusions stemming from cells in the early stages of individual detachment were not counted as strands. The points where the edges of the protrusive strands intersected with the circular spheroid periphery were deemed the strand base edges, and the strand base width was measured as the distance between these 2 edges (Fig. 2). The max strand width was the base strand width for most epithelial strands. For EMT-induced mesenchymal strands, the max strand width was measured at the widest point along the strand's entire length, which varied for each strand. Both mesenchymal and epithelial strand length were measured from the center of the strand base to the tip of the protrusion (Fig. 2). In the case of branching mesenchymal strands, the length was measured from the center of the strand base to the furthest branch tip. The z-plane at the center of the spheroid was used to measure total invasion area. The NIH-Elements measurements polygon function was used to outline and quantify the area of the main spheroid body and all associated protrusions at designated timepoints. Statistical significance was determined via ANOVA and Tukey post-hoc tests. Statistical reports are included in relevant figure captions.

Nuclei and Particle Tracking. ImageJ software was used to process time-lapse images for optimal performance of particle tracking software. Detailed descriptions of all crops, filters, and adjustments are reported in Appendix A. Imaris 8.2.1 particle tracking software and custom MATLAB code were used to track nuclei and tracer particle position over time. Nuclei and tracer

particle positions were compared to the previous frame to generate particle displacement vectors and magnitudes using 32 x 32 pixel interrogation windows (0.64 μm/pixel). MATLAB scripts used for processing spheroid data are reported in Appendix A.

Results

Spheroids Invade Collectively Without EMT Induction. We first analyzed the invasion profiles of uninduced MCF-10A spheroids and found that they exhibit collective invasion patterns when embedded in a 3D matrix. We show a representative example of an epithelial

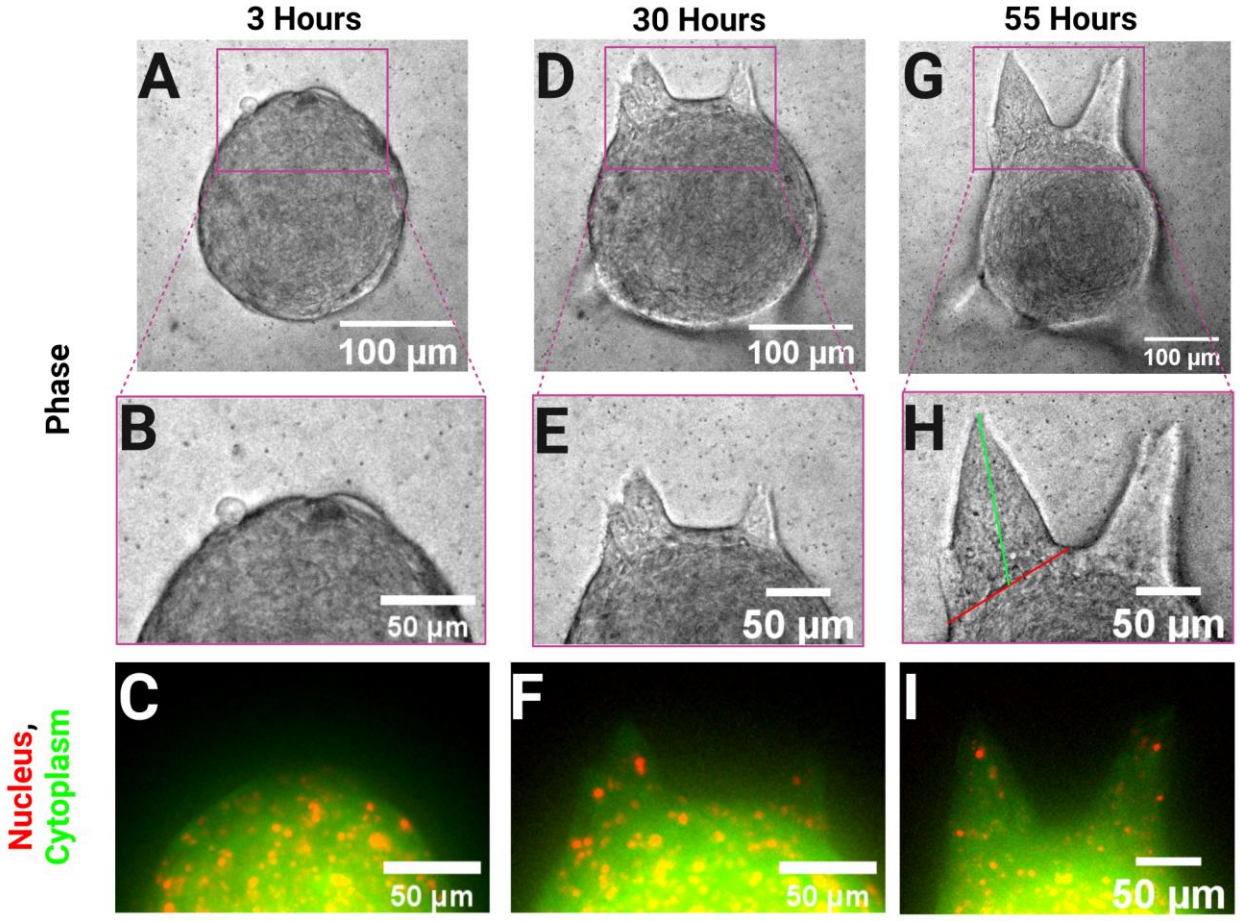

Fig. 2 Epithelial Spheroid (-OHT) Invasion. Phase images of an uninduced epithelial MCF-10A spheroid (~500 cells) embedded in 6 mg/mL Col-1 3D matrix deforming the ECM and invading over the course of 55 hours (A-C).

Close-up phase images of collective invasion and the formation of multicellular strands over 55 hours (insets) (D-F). Representation of strand length (green line) and strand base width (red line) quantification method (F). Merged images of live cells with fluorescent proteins in the nucleus (red, mCherry-H2B) and cytoplasm (green, GFP) (G-I).

spheroid invading into a 6 mg/mL Col-1 3D matrix over a 55 hour period. At early time points, epithelial spheroid morphology is highly circular as cells have yet to begin deforming the ECM (Fig. 2A-C). After 30 hours, multicellular protrusive strands appeared that were wide at the base and taper towards the leading end (Fig. 2D, E). Fluorescence imaging reveals that cells collectively initiate protrusions and the base of the strand starts as multiple cells wide (Fig. 2F). From 30 to 55 hours, strands propagate in a highly organized manner and retain their shape as more follower cells were recruited (Fig. 2G-I).

This collective invasion pattern was conserved across multiple individual spheroids and their strands. Quantitative analysis of epithelial strand profile over 55 hours reveals that both strand length (Fig. 3A) and base width (Fig. 3B) increase rapidly at early time points but level

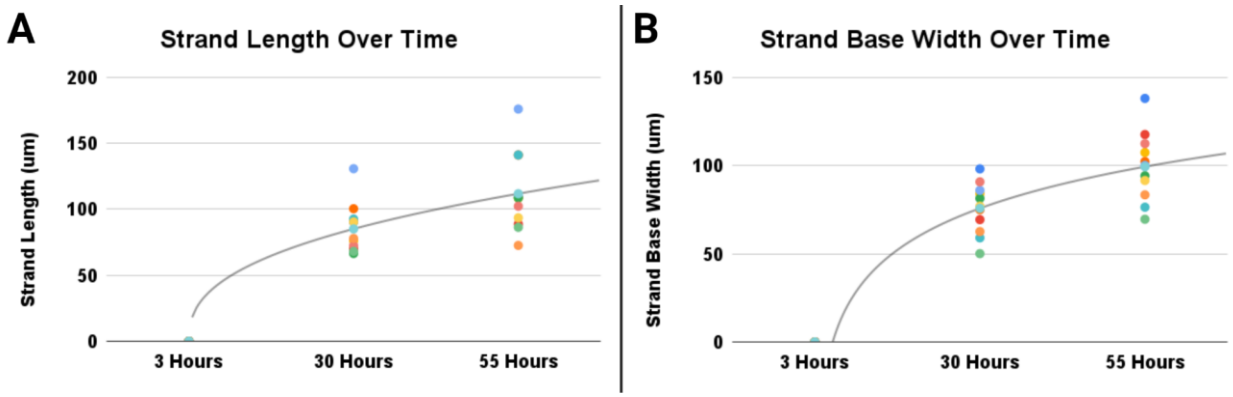

Fig. 3 Epithelial Strand (-OHT) Quantification. Length (A) of 11 individual epithelial strands over a 55-hour period fit to a logarithmic trendline (black line) (y=85.1+38.5 ln x). Base width (B) of 11 individual epithelial strands over a 55 hour period fit to a logarithmic trendline (black line) (y=75.8+34.1 ln x). Data are summarized form 3 independent experiments, one-sided ANOVA (all $p<0.05$).

out in a logarithmic fashion. Strand length increases at approximately 2.8 μm/hour for the first 30 hours but decreases to 1.1 μm /hour for the subsequent 25 hours. Similarly, base width increases at about 2.5 μm /hour for 30 hours at which point the rate significantly decreases to .94 μm /hour.

EMT Induction Causes Individual Spheroid Invasion. Next, the invasion profiles of EMT-induced mesenchymal MCF-10A spheroids were analyzed to determine the effect of EMT

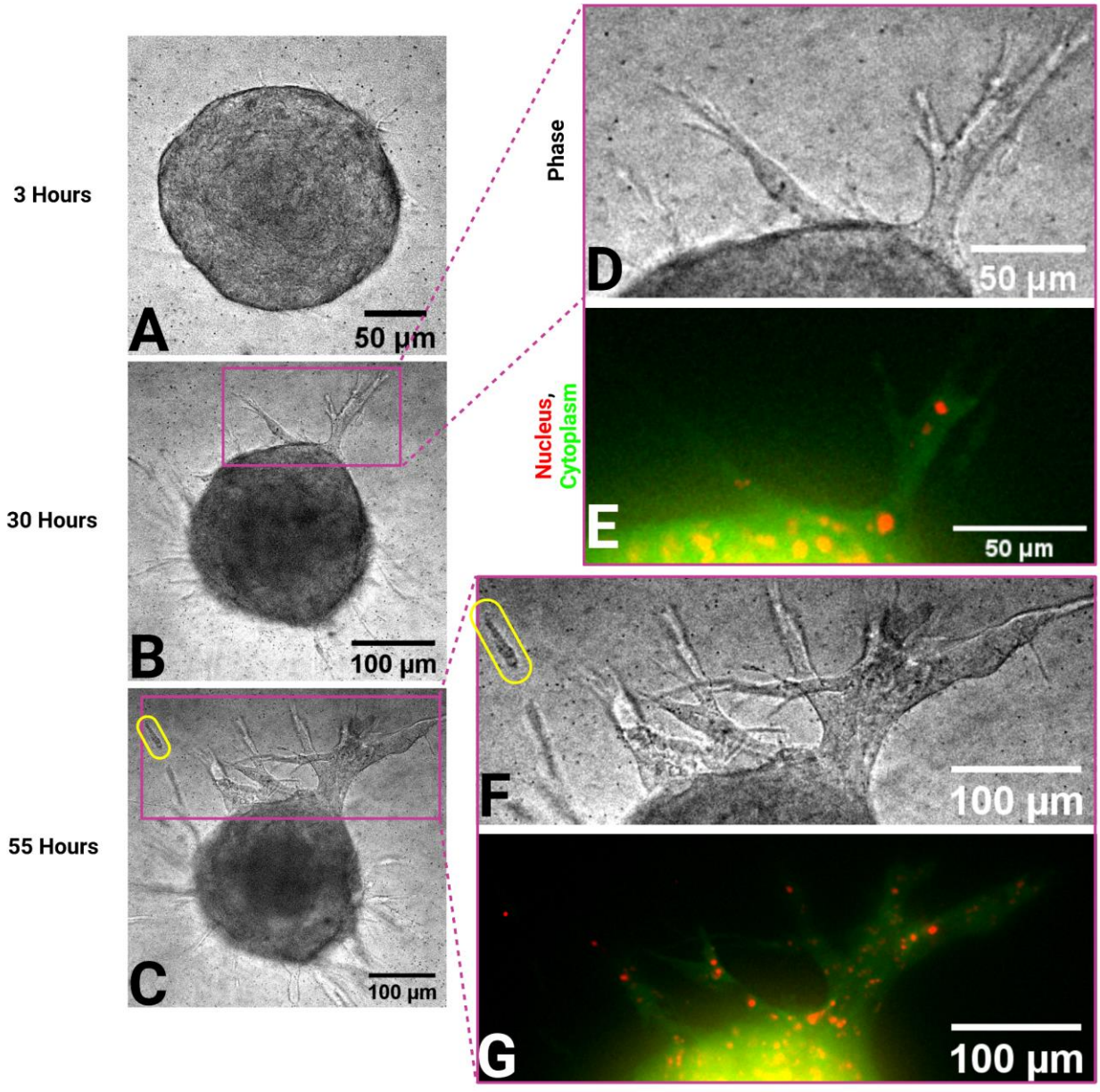

Fig. 4 Mesenchymal Spheroid (+OHT) Invasion. Phase images of a mesenchymal MCF-10A spheroid (~500 cells) embedded in 6 mg/mL Col-1 3D matrix deforming the ECM and invading over the course of 55 hours (A-C).

Close-up images of single-cell detachment (yellow ellipse), individual invasion, and the formation of multicellular mesenchymal strands over 55 hours (insets) (D-G).

induction on invasion dynamics. EMT-induced mesenchymal spheroids exhibited mostly individual but also collective invasion when embedded in a 3D matrix. We show a representative example of a mesenchymal spheroid invading into a 6 mg/mL Col-1 3D matrix, supplemented with induction media (1:2000 OHT), over a period of 55 hours. Like uninduced epithelial spheroids, EMT-induced spheroids display a circular morphology at early time points (Fig. 4A). After 30 hours, narrow spindle-like protrusions about 11 μm wide at the base begin to appear (Fig. 4B, D). Fluorescence images show that cells enter strands in a single-file manner, causing strands to be as wide as about 1 cell (Fig. 4E). Between 30 and 55 hours, more cells on the periphery initiate thin, single-cell width protrusions. In addition, some strands initiated at earlier time points have increased in base width and multicellular strands are able to form (Fig. 4C, G). A hallmark of EMT-induced spheroid morphology is strand branching. Leader cells often individually diverge from other cells in their strand and initiate secondary and even tertiary protrusion (Fig. 4F). Single cells with highly elongated morphologies detach from the main body

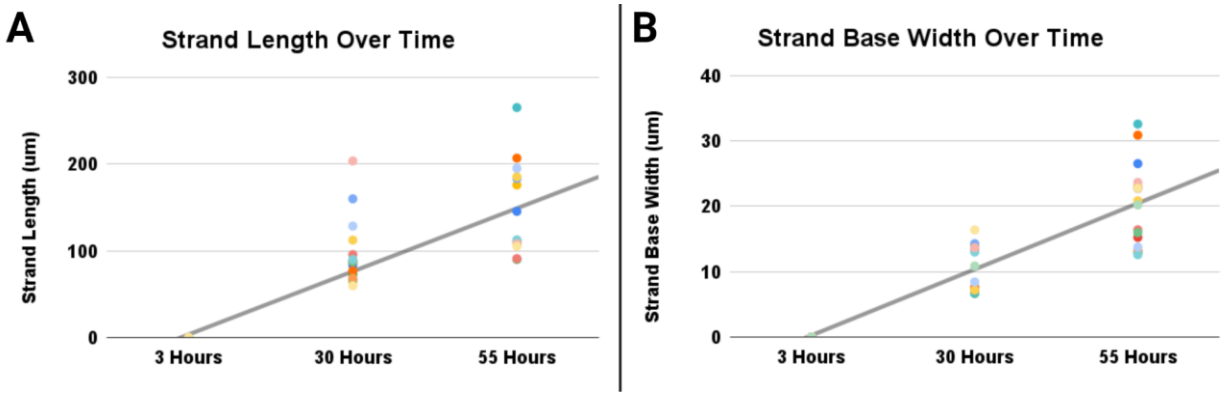

Fig. 5 Mesenchymal Strand (+OHT) Quantification. Length (A) of 15 individual epithelial strands over a 55 hour period fit to a linear trendline (black line) (y=72.7x + 3.51). Base width (B) of 15 individual epithelial strands over a

55 hour period fit to a linear trendline (black line) ($y=10.1x + 0.252$). Data are summarized form 3 independent experiments, one-sided ANOVA (all $p<0.05$).

of mesenchymal spheroids and can be seen migrating more than 100 μm into the ECM in some cases (Fig. 4F).

The rate of strand base and strand base width increase showed a linear trend for EMT-induced spheroids. For the first 30 hours, the average rate of strand length increase was approximately 3.6 μm /hour. For the next 25 hours, this rate decreased to approximately 1.9 μm /hour (Fig. 5A). Interestingly, the base width expansion of EMT-induced strands was consistent over the entire 55 hour invasion period, with an increase of about .4 μm /hour for the first 27 hours and .37 μm /hour for the subsequent 25 hours (Fig. 5B).

EMT Induction Increases Strand Number and Individually Detached Cells but Decreases Strand Width. Representative images of spheroid morphology after 72 hours embedded in 6 mg/mL Col-1 matrix further demonstrate wider, more populated invasive strands in uninduced epithelial spheroids (Fig. 6A). In addition, closer depictions of individual mesenchymal strands further illustrate the highly branched morphology seen following EMT induction (Fig. 6B).

Spheroids induced to undergo EMT exhibited an average of 7 times as many individually detached (single) cells as epithelial spheroids. In addition, OHT-induced spheroids typically had twice as many strands per spheroid than uninduced spheroids. More specifically, epithelial spheroids commonly exhibited an average of 10 strands and 3 single cells per spheroid, while mesenchymal spheroids usually exhibited an average of 20 strands and 22 single cells per spheroid (Fig. 6C, i & ii).

Representative images of spheroid morphology at 72 hours allowed for further analysis of differences between strand length and width between epithelial and spheroids induced to undergo EMT. Results indicated that mesenchymal strands were ordinarily 148 μm long, 28 μm wide at the base, and maintained an approximately 2 cell wide diameter along the length of the strand. On the other hand, uninduced strands were shown to be about 140 μm long, 87 μm wide, and taper off towards the strand tip (Fig. 6C, iii & iv).

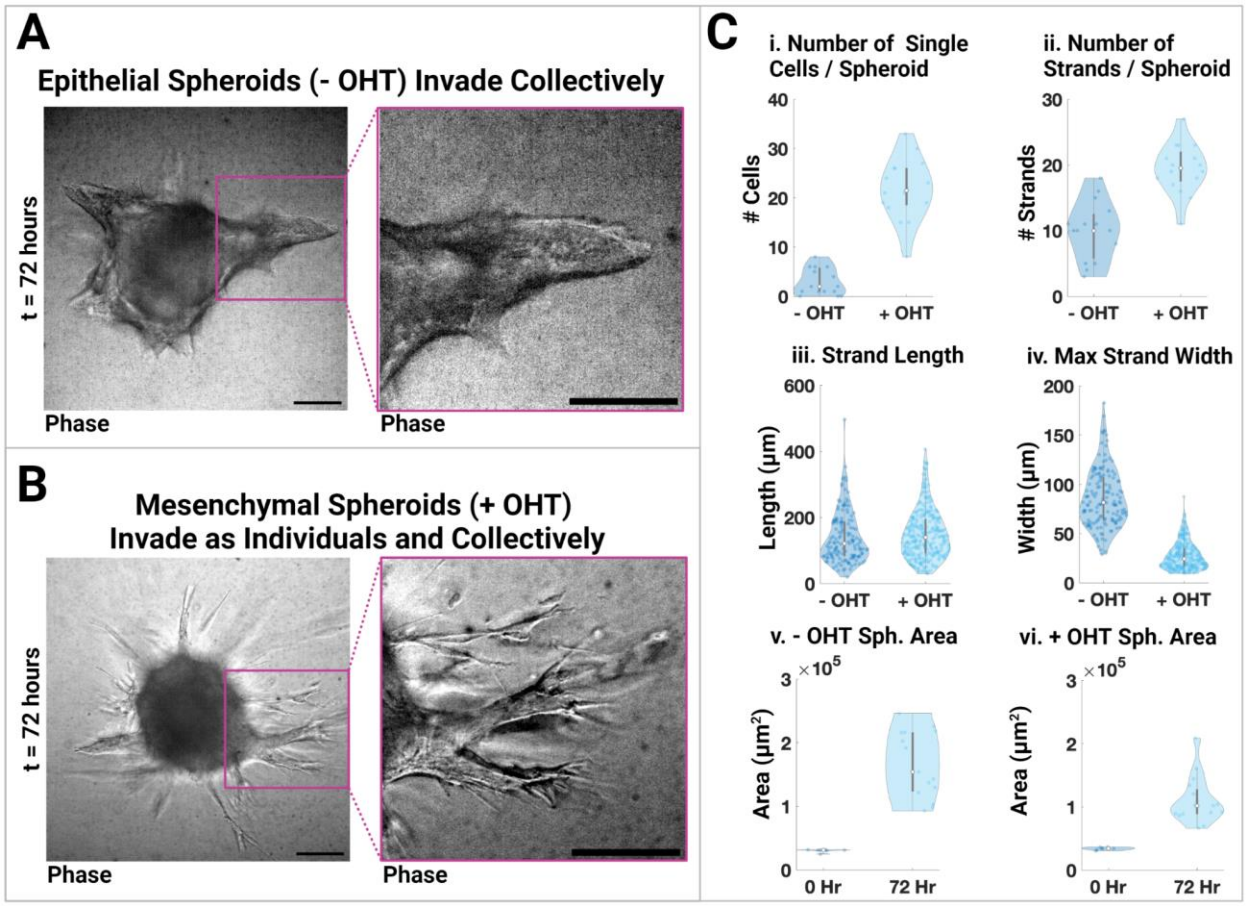

Fig. 6 Spheroid Morphology Comparison. Epithelial Spheroids embedded in 6 mg/mL Col-1 3D matrix invade collectively over 72 hours. Inset closely depicts epithelial strand morphology (A). Mesenchymal Spheroids embedded in 6 mg/mL Col-1 3D matrix invade collectively over 72 hours. Inset closely depicts Mesenchymal strand morphology (B). Epithelial spheroids show a greater overall invasion area, a few large protrusive strands that are multiple cells wide, and a small number of individually detached cells. Mesenchymal spheroids display a slightly

lower invasion area, many spindle-like strands, and numerous individually detached cells (C). Data are summarized from 15 independent spheroids in 1 experiment, one-sided ANOVA (all $p<0.05$) Scale bar = 100 μm.

When spheroids were initially embedded into the Col-1 3D matrix, they consistently had similar cross-sectional areas with uninduced spheroids having an average cross-sectional area of 3E4 μm^2 and induced spheroids having an average cross-sectional area of 3.3E4 μm^2. However, over the course of 72 hours, EMT induction caused a decrease in invasion area with epithelial spheroids showing an average invasion area of 1.7E5 μm^2 and mesenchymal spheroids having an average invasion area of 1.1E5 μm^2 (Fig. 6C, v & vi).

Pronounced Vimentin Expression in EMT-Induced Spheroids. Rabbit anti-vimentin staining allowed the use of fluorescence imaging to verify vimentin expression in induced and uninduced spheroids. Results indicated that the uninduced epithelial control spheroids did not express vimentin in the interior but upregulated vimentin expression along invasive strands (Fig. 7A). Moreover, immunofluorescence staining indicated that the EMT-induced spheroids not only

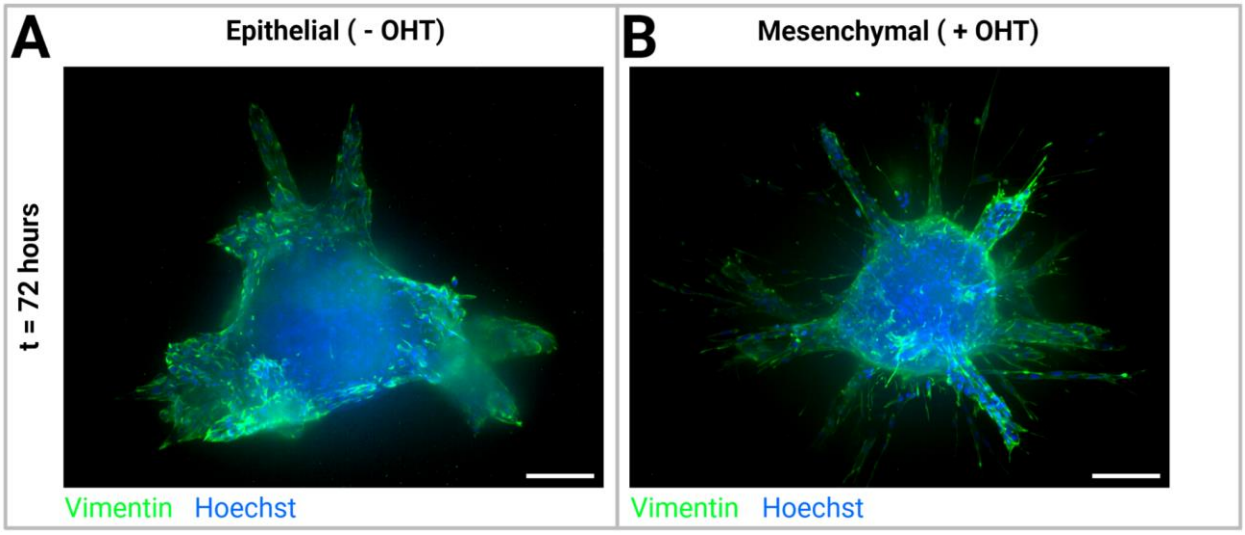

Fig. 7 MCF-10A Spheroids (-/+OHT) Express Vimentin Along Invasive Strands. Merged images of fixed cells stained for vimentin (green, Vimentin XP Rabbit mAB) and the nucleus (blue, Hoechst).

upregulated vimentin along invasive strands but also within the interior. Fascinatingly, vimentin expression was especially upregulated in detached single cells as indicated by a strong intensity signal (Fig. 7B).

Self-Sorting Mosaic Spheroids Invade Collectively and Individually. To expand on our 3D invasion model, we performed preliminary work on creating mosaic spheroids composed of both WT MCF-10A and EMT-induced MCF-10A cells. Mosaic spheroids were created by combining cultures of epithelial WT MCF-10A mCherry-nuc cells (nuclear label only) and OHT-induced MCF-10A cells (nuclear and cytoplasm labels). Spheroids composed of 50% epithelial (WT) and 50% EMT-induced cells were embedded in 6 mg/mL Col-1 3D matrix and cultured for 72 hours. Although both cell populations had the mCherry nuclear label, the red fluorescence signal from the OHT-induced cells was far brighter than that of the mCherry-nuc cells. This allowed us to adjust the exposure such that only the OHT-induced cell nuclei were visible. Nuclei co-localization with the green cytoplasm signal indicates cells belonging to the EMT-induced cell population (Fig. 8, E-H). At the initial timepoint, the center-concentrated cell signal suggests that EMT-induced mesenchymal cells localized towards the core of the spheroid and that the epithelial cells surrounded them along the periphery of the spheroid (Fig. 8A, E). After 24 hours, the cell populations remained sorted, and protrusive strands exhibiting the wide-based epithelial shape appeared (Fig 8B, F, I).

After 36 hours, the green cytoplasm signal from the mesenchymal cell population reaches the periphery of the spheroid and new protrusions are initiated that exhibit mesenchymal strand morphologies. Compared to the strands initiated at 24 hours, the strand seen protruding from the

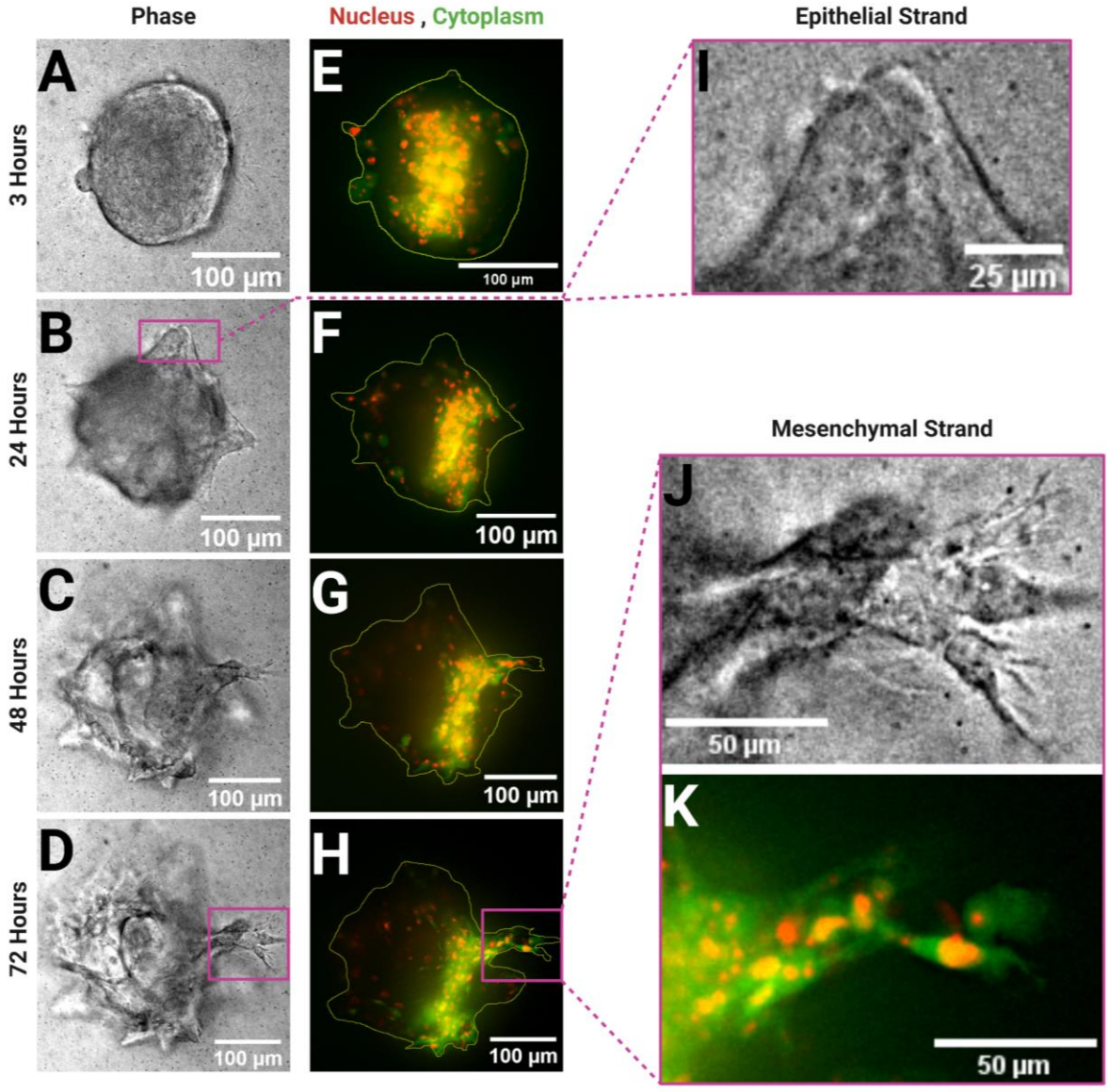

Fig. 8 Mosaic MCF-10A Spheroids Self Assemble. MCF-10A Spheroids (~500 cells) composed of 50% epithelial (mCherry-nuc) and 50% EMT-induced cells embedded in 6 mg/mL Col-1 3D matrix self-assemble, deform the ECM, and invade over the course of 72 hours (A-D). Merged images of live cells with fluorescent proteins in the

nucleus (red, m-Cherry H2B) and cytoplasm (green, GFP) (E-H). Close-up image of epithelial (I) and mesenchymal (J, K) strands protruding from the same mosaic spheroid.

right side of the spheroid at 36 hours is more spindle-like and has a thinner base (Fig. 8C). The single cell wide mesenchymal strand observed at 36 hours exhibits a fluorescent signal indicating that it is composed of EMT-induced cells (Fig. 8G). At the final time point, strands populated by EMT-induced cells more clearly exhibit the branched morphology seen prior in fully induced mesenchymal spheroids (Fig 8J, K). In addition, strands exhibiting both epithelial-like and mesenchymal-like morphologies are both observed in the same mosaic spheroid at the final 72 hour timepoint.

Discussion and Conclusion

In this study, we use an invasion model to elucidate how EMT affects multicellular tractions and remodeling in a 3D biomimetic matrix. Our experiments reveal that inducing EMT induction in MCF-10A spheroids via Snail-1 gene expression weakens cell-cell adhesions and inhibits leader-follower cell behavior, thus decreasing ECM remodeling capability and coordinated cell motion.

Epithelial control spheroids exhibited concerted multicellular invasion. Appreciable cell-cell adhesions in epithelial spheroids led to the formation of wide-based protrusive strands that taper towards the leading end. As leader cells at the periphery of epithelial control spheroids naturally experienced EMT, the presence of follower cells with tight cell-cell junctions allowed for leader-follower behavior to conserve the tapered morphology of multicellular strands as they invaded further into the ECM. The ability for invasive leader cells to recruit follower cells for collective ECM deformation in the uninduced case allowed for strands initiated at early timepoints to rapidly increase length and base width. Over time, interactions amongst the increasing population of cells within epithelial strands decrease the elongation rate. Essentially, the growing number of cells in multicellular strands created “traffic” for the leader-follower cell mechanism hypothesized to drive ECM and facilitate collective invasion. At early stages, there are limited points where follower cells can enter the strands, so many enter near the strand edge

and contribute to the growing strand base width. As time goes on, epithelial cells can enter strands through the center of the base without contributing to base widening. This coincides with the observed slowing of epithelial strand base width expansion rate over time. We hypothesize that the collective orbiting motion of cells within spheroids is not only dependent on sufficient cell-cell adhesions but also on a circular periphery. As a result, the widening strand bases act as "defects" along the spheroid perimeter that disrupt the coherent rotational motion. In other words, as more epithelial cells inside the spheroid are recruited as followers, there are less to facilitate collective orbiting and the behavior is gradually lost. Furthermore, vimentin expression in epithelial control spheroids was localized to invasive strands and cells on the periphery with no vimentin signal detected from interior cells that do not interact with the ECM. This indicates that EMT activation in uninduced spheroids was naturally triggered by the biochemical and physical properties of the ECM.

In comparison, induced EMT and increased vimentin expression in mesenchymal spheroids resulted in more randomized individual invasion compared to the epithelial control. Diminished cell-cell adhesions and strengthened cell-matrix adhesions promoted the outward radial migration of individual cells and led to individuals breaking off from the main spheroid body and the initiation of more numerous protrusions. Mesenchymal spheroids developed spindle-like protrusions that cells enter in a single-file manner and are much narrower than epithelial strands. EMT-induced spheroids were unable to form large multicellular protrusions that could significantly contribute to total invasion area, resulting in lower overall invasion area following EMT induction. After EMT induction, cell-matrix interaction preference over cell-cell interaction inhibits collective invasion and mesenchymal strands remain narrow along their entire length as opposed to the tapered morphology of epithelial strands.

The decreased abundance of tight cell-cell junctions following EMT induction had an inhibitory effect of leader-follower cell behavior. We hypothesize that the absence of epithelial follower cells in EMT-induced spheroids prohibited the formation of multicellular strands. This coincides with our observation of increased strand count following EMT induction as preference for cell-matrix adhesion over cell-cell adhesion caused cells to initiate new protrusions instead of following another cell's tractions into the ECM. This is consistent with our finding that the rate of mesenchymal strand length extension decreased at a constant rate over time. Weakened cell-cell adhesion in EMT-induced strands prevents the intracellular "traffic" that more rapidly stymies epithelial strand elongation. Interestingly, the average strand base width of mesenchymal spheroids increased at a constant rate. We suggest that base width expansion of EMT-induced strands relied on individual propensity to deform the ECM as opposed to the coherent collective rotation of cells. At early time points, there was room on the spheroid periphery for EMT-induced cells to initiate single-cell protrusions that had very narrow strand base width. As the spheroid perimeter became more saturated with protrusions, mesenchymal cells emerging from the interior began contributing to base width instead of strand initiation and caused a gradual increase in the rate of strand base width expansion. Interestingly, EMT-induced strands exhibited branching as low cell-cell adhesion compared to cell-matrix adhesion caused individual cells to diverge from increasingly multicellular strands and initiate secondary or even tertiary protrusions.

Decreased intra-spheroid nuclei displacement vectors following EMT further demonstrates how loss of cell-cell adhesion prevents collective invasion. Because the motion of cells within mesenchymal spheroids is less coordinated, cells bump into each other as they all move individually and prohibit each other from being able to travel far distances within the

spheroid interior. Decreased particle displacement magnitude following EMT induction indicates loss of ECM remodeling capabilities. The elongated morphology allows invading mesenchymal cells to squeeze and weave through gaps in the collagen matrix in a snake-like motion. This coincides with the lack of tracer particle displacement as EMT induction allows single cells to invade without the need for generating significant matrix deformations. The epithelial morphology of most cells in uninduced spheroids prevents them from squeezing through gaps in the ECM, making leader-follower cell behavior necessary for invasion.

Fascinatingly, EMT induction caused vimentin expression to persist further into the spheroid interior. This phenomenon coincides with our observation that the individualized behavior of mesenchymal cells in the spheroid periphery disrupted coordinated cell motion. The particularly strong vimentin signal from individually detached cells coincides with their observed elongated morphology and invasive behavior. Since vimentin provides cells with a viscoelastic framework that protects internal organelles, EMT-induced vimentin upregulation gave single cells the structural integrity to squeeze through gaps in the ECM.

Epithelial control and vimentin overexpressing cells displayed different invasive behaviors even when co-cultured. Self-sorting after embedding of mosaic spheroids can be explained by weakened cell-cell adhesions in EMT-induced population. Mosaic spheroids appear sorted immediately upon embedding, indicating that the process is initiated during hanging drop formation. Further investigation is required to determine why epithelial cells encompass the mesenchymal cells instead of forming a tightly packed core and pushing mesenchymal cells to the periphery while aggregating in a hanging drop. However, the emergence of epithelial-type protrusions at early time points after mosaic spheroid embedding coincides with epithelial cells being localized around the periphery. The appearance of branched mesenchymal-type

protrusions at 48 hours suggests that the invasive nature of mesenchymal cells has allowed them to break through the barrier posed by cell-cell adhesion between epithelial cells. At later time points, distinct regions of mesenchymal and epithelial invasion exist as the sorted cell populations exhibit their respective behavior patterns as they interact with the ECM.

Future endeavors will utilize our established invasion model to further elucidate the role of how vimentin affects multicellular tractions and remodeling in 3D biomimetic matrix. For example, staining for other intermediate filaments may paint a more wholistic picture of how vimentin cooperates with other contributors to EMT in the tumor microenvironment. This has significant implications as future technologies that specifically target may have the ability to prevent tumor growth and improve therapeutic outcomes. We have also done preliminary work on embedded mosaic spheroids with varying ratios of induced and uninduced cell populations. It would be interesting to know the minimum amount of EMT-induced mesenchymal cells needed to inhibit intra-spheroid collective orbiting. Furthermore, we have also done preliminary work probing the internal forces exerted by rotating and invading cells using highly compliant microparticles (HCMPs). HCMPs were designed to have the same size (13 μm diameter) and stiffness (E = 250 Pa) of a cell as well as be coated in Col-1. The bioactive coating allowed for HCMPs to be incorporated into the main spheroid body during hanging drop. After embedding, the HCMPs were deformed and displaced as cells collectively invade around them. By quantifying these deformations and displacements we can determine the internal forces exerted by cells during invasion. Simultaneously use of HCMPs and TFM would provide a novel method for mapping both the internal and external stresses exerted by multicellular spheroids and deepen our understanding of cancer invasion. Lastly, the high-throughput and translatable nature of our

spheroid invasion model means it's widely applicable to the investigation of the underlying mechanisms that drive various diseases.

In summary, we demonstrate that inducing MCF-10A cells to undergo EMT via Snail-1 gene expression causes a shift from collective to individual invasion due to decreased cell-cell adhesions and elongated phenotype. Uninduced spheroids showed leader-follower cell behavior as naturally activated EMT allowed leader cells to recruit follower cells into large multicellular strands. Loss of leader-follower cell behavior after EMT led to more narrow protrusive strands that cells enter in a largely single-file manner. Furthermore, we demonstrate how decreased cell-cell adhesions following EMT inhibited coordination cell motion within the spheroids as cells were more individually determined to invade instead of act as a cohesive unit. Interestingly, we also show preliminary data that co-culture of EMT-induced and uninduced cells can cause spontaneous cell sorting and differential invasion profiles within a single spheroid. Finally, we show compelling evidence that EMT-induced vimentin upregulation provides individually detached mesenchymal cells with the structural integrity to squeeze through small gaps in the 3D matrix. These physical signatures of spheroid invasion have further applicability to elucidate phenotypic transitions in wound-healing, development, and tumor progression.

References

1. Lee JY, Chang JK, Dominguez AA, Lee H-P, Nam S, Chang J, Varma S, Qi LS, West RB, Chaudhuri. Yap-independent mechanotransduction drives breast cancer progression. Nat Commun. 2019;10(1):184-9.
2. Legget, S. E., Hruska, A. M., Guo, M. & Wong, I. Y. The epithelial-mesenchymal transition and the cytoskeleton in bioengineered systems. Cell Commun. Signal. 2021:19(32).
3. Tevis, T. M., Colson, Y. L., Grinstaff, M. W. Embedded spheroids as models of the cancer microenvironment. Advanced Biosystems. 2017;1(10).
4. Hruska, A. M., Yang H., Legget, S. E., Guo, M., & Wong, I. Y. Mechanobiology of collective cell migration in 3D microenvironments. arXiv preprint. 2022; arXiv:2202.02849.
5. Yang, J. et al. Guidelines and definitions for research on epithelial-mesenchymal transition. Nat Rev Mol cell Biol. 2020; 21:341-352.
6. Thomas, P. A. et al. Association between keratin and vimentin expression, malignant phenotype, and survival in postmenopausal breast cancer patients. Clin Cancer Res. 1999; 5: 2698-2703.
7. Javaid, S. et al. Dynamic chromatin modification sustains epithelial-mesenchymal transition following inducible expression of Snail-1. Cell Rep. 2013; 5:1679–1689.
8. Leung, B. M., Lesher-Perez, S. C., Matsuoka, T., Moraes, C., & Takayama, S. Media additives to promote spheroid circularity and compactness in hanging drop platform. Biomaterials science. 2015; 3(2); 336-344.

9. Legget, S. E., Neronha, Z. J., Bhaskar, J. Y., Sim, T. M., & Wong, I. Y. Motility-limited aggregation of mammary epithelial cells into fractal-like clusters. Proc Natl Acad Sci USA. 2019; 116;17298-17306.
10. Debnath, J., Muthuswamy, S. K., Brugge, J. S. Morphogenesis and oncogenesis of MCF-10A mammary epithelial acini grown in three-dimensional basement membrane cultures. Methods. 2003; 30; 256–268.
11. Naber, H. P., Wiercinska, E., ten Dijke, P., van Laar, T. Spheroid Assay to Measure TGF-β-induced Invasion. J. Vis. Exp. (57), e3337, doi:10.3791/3337 (2011).

Appendix A: Supplementary Methods

Cell Culture

Making Growth and Resuspension Media. Routine culture of WT MCF-10A cells was performed in a high EGF growth medium consisting of the components listed in Table 1 (10). Fresh batches of growth and resuspension media were made monthly by combining all reagents in a sterile 250 mL or 500 mL PES Membrane (0.22 μm) Vacuum Filter and stored at 4°C. Media components were added in order of highest to lowest volume and kept at the storage conditions indicated by the manufacturer until use. Induction medium was formulated by supplementing growth medium with a 4-hydroxytamoxifen (OHT) in DMSO solution (500 nM final OHT concentration) (Sigma T5648) to induce Snail-1 expression and initiate epithelial-mesenchymal transition (EMT) (1).

Reagent	Final Concentration (Growth Media)	Final Concentration (Resuspension Media)	Manufacturer
DMEM/F12 with HEPES Buffer	NA	NA	Fisher 11330032
Horse Serum	5%	20%	Fisher 16050122
Animal-Free Recombinant Human Epidermal Growth Factor (EGF)	20 ng/mL	Not Included	PeproTech AF-100-15

Hydrocortisone	0.5 μg/mL	Not Included	Sigma H0888
Cholera Toxin	100 ng/mL	Not Included	Sigma C8052
Insulin from Bovine Pancreas	10 μg/mL	Not Included	Sigma I6634
Penicillin-Streptomycin (Pen-Strep)	1%	1%	Corning 30-002-CI

Table 1. Growth Media Components (2).

Routine Cell Culture. WT MCF-10A cell stocks were cryopreserved at -196℃ in growth medium with 10% Dimethyl Sulfoxide (DMSO)(BP231-100). Cell stocks were first thawed by hand to remove ice buildup on the outside of the cryovial (Fisher 0337421). Thawing was carefully continued at 37℃ with the cryovial moved to room temperature once the last ice crystal melted to prevent heat shock. To resuspend the cells, a 5 mL serological pipette (Genesee 12-102) was filled with 5 mL of warm (37℃) growth medium that was added dropwise to the cryovial, again to avoid heat shock. The cell suspension was then slowly transferred to a 15 mL conical tube (Genesee 28-103) and centrifuged at 125 XG for 6 minutes using a Sorvall Legend X1R centrifuge (Fisher). After aspirating the supernatant, cells were resuspended in 1 mL of warm (37℃) growth medium and added to a T-75 cell culture flask (Genesee 25-209) with 9 mL of warmed (37℃) growth medium at a density of 100,000 cells/mL to be incubated at 37℃ and 5% CO_2.

Prior to spheroid generation, WT MCF-10A cells were routinely cultured at 37℃ and 5% CO_2 (passed after about 48 hours of incubation) in T-25 cell culture flasks (Genesee 25-204) at a high cell density and grown to confluence to prevent spontaneous EMT-like changes and preserve the epithelial phenotype (2,3). To do this, a single cell suspension of MCF-10A cells was created by first washing the 2D monolayer 3 times with 3 μL of warmed (37℃) 1X PBS. Cells were then dissociated from the T-25 tissue culture flask using 1 mL of warmed (37℃) Accumax (Innovative Cell Technologies) and 1 mL of warmed (37℃) Trypsin/EDTA solution (ATCC PCS-999-003). Once dissociation was microscopically confirmed (~10 minutes), Accumax and Trypsin/EDTA were neutralized with 3 mL of warmed (37℃) growth medium and 3 mL of warmed (37℃) resuspension medium. The solution was then transferred from the T-25 tissue culture flask into a 15 mL conical tube. The single cell suspension was then centrifuged at 125 XG for 6 minutes and resuspended in 1 mL of growth medium after aspiration of the supernatant. For routine culture, WT MCF-10A cells were passed at a 1:5 split. In the case of spheroid generation, 20 μL of single cell suspension was transferred into each chamber of a Cellometer Cell Counting Slide (Nexcelom Bioscience) and a Cellometer Auto 1000 Counter (Nexcelom Bioscience) was used to quantify the concentration of the single cell suspension. The final cell concentration reported was an average of the 2 counting slide chambers. Then, based on when the next routine subculture was planned for, the cell suspension was split at an optimal ratio and seeded in a fresh T-25 flask with 5 mL of warmed (37℃) growth medium. For example, passing a confluent T-25 at 1:5 was ideal if the cells were to be passed again after 2 days. In the case that WT MCF-10A cells were to be used immediately for spheroid generation, the cell suspension was diluted to 500,000 cells/mL using a warmed (37℃) growth medium.

Prior to spheroid generation, OHT-induced MCF-10A cells were routinely cultured (passed after about 72 hours of incubation) in T-75 cell culture flasks at a low cell density (30,000 cells/mL) and low confluence (~20%) to limit cell-cell contact and preserve the mesenchymal phenotype (2,3). To confirm EMT induction, OHT-induced MCF-10A cells were cultured in induction medium for at least one full passage before use for spheroid generation and microscopically examined to ensure they exhibit an elongated mesenchymal phenotype. To do this, a single cell suspension of OHT-induced MCF-10A cells was created by first washing the adhered cells 3 times with 5 μL of warmed (37℃) 1X PBS. Cells were then dissociated from the T-75 tissue culture flask using 2 mL of warmed (37℃) Accumax and 1 mL of warmed (37℃) Trypsin/EDTA solution. Once dissociation was microscopically confirmed (~5 minutes), Accumax and Trypsin/EDTA were neutralized with 4 mL of warmed (37℃) growth medium and 5 mL of warmed (37℃) resuspension medium. The solution was then transferred from the T-75 tissue culture flask into a 15 mL conical tube. The single cell suspension was then centrifuged at 125 XG for 6 minutes and resuspended in 1 mL of growth medium after aspiration of the supernatant. For OHT-induced MCF-10A cells, 20 μL of single cell suspension was transferred into each chamber of a Cellometer Cell Counting Slide and a Cellometer Auto 1000 Counter was used to quantify the concentration of the single cell suspension for both routine cell culture and spheroid generation. Then, based on when the next routine subculture was planned for, the cell suspension was split at an optimal ratio and seeded in a fresh T-75 flask with 10 mL of warmed (37℃) induction medium. For example, passing an approximately 30% confluent T-75 at 1:15 was ideal if the cells were to be passed again after 3 days. In the case that MCF-10A cells were to be used immediately for spheroid generation, the cell suspension was diluted to 500,000 cells/mL using warmed (37℃) growth media.

Spheroid Generation

Methylcellulose Solution Preparation. To prepare the methylcellulose solution, 500 mg of MethoCel A4M (Sigma 94378) powder was put into a glass beaker with a magnetic stirrer bar. After covering the top of the beaker with aluminum foil and autoclave tape, the powder was autoclaved using a 30-minute gravity cycle. 50 mL (Genesee 28-103) of a 10 mg/mL stock methylcellulose solution was made in a sterile biosafety cabinet by adding 45.5 mL of pre-warmed (60℃) sterile 1X Phosphate Buffered Saline (PBS) (Fisher SH30256) with 0.5 mL 100X Pen-Strep (1% final concentration) to the 500 mg of autoclaved powder. Then, the powder was dissolved by setting the solution on a 60℃ hot plate with magnetic stirring for 20 minutes. To completely dissolve the powder, the hot plate (heat turned off) was moved to a refrigerator (4℃) and stirred overnight. Subsequently, the methylcellulose solution was transferred to a 50 mL conical tube and centrifuged at 5000 XG for 1 hour at 4℃. The clear, highly viscous, solution was then transferred into fresh 15 mL conical tubes and stored at 4℃ until use (11).

Fixing and Staining of Collagen Matrices.

Fixation. To fix the ECM, a 4% (v/v) solution of paraformaldehyde (PFA) (BICCA 3180-16) in 1X PBS was first prepared. Then, an aspirator was used to remove all growth media from each of the wells while being careful to not damage the hydrogels. 100 μL of 4% PFA was subsequently added to each well, and the entire plate was placed on a rocker table for 45 minutes at room temperature. Next, after aspirating the PFA/PBS, each well was washed 3 times with 100 μL of 1X PBS with the plate being returned to the rocker table for 5 minutes between washes. After the

final wash, hydrogels were preserved in a 0.2% (w/v) Sodium Azide (Sigma S2002-5G) solution in 1X PBS and stored at 4℃ until further use.

Permeabilization. For this step and all subsequent steps in the fixing and staining protocol, the rocker table was placed in the refrigerator (4°C). To permeabilize the ECM, a 0.25% (v/v) solution of Triton-X (Sigma 9036-19-5) in 1X PBS was first prepared. PBS/Sodium Azide was then aspirated from each well, again being careful not to disturb the hydrogel. Next, 100 μL of 0.25% Triton-X was added to each well, and the plate and placed on the rocker table for 45 minutes. After aspirating the Triton-X/PBS, each well was washed 4 times with 100 μL of 1X PBS with the plate being returned to the rocker table for 10 minutes between washes.

Blocking. To block the ECM from non-specific binding, a 5% (v/v) solution of goat serum (Sigma G9023) in 1X PBS was first prepared. After aspirating the PBS from each well, 100 μL of 5% goat serum was added and the plate was wrapped in parafilm (BEMIS PM-992). Next, the plate was placed on the rocker table overnight. After aspirating the goat serum/PBS, each well was washed 3 times with 100 μL of 1X PBS with the plate being returned to the rocker table for 5 minutes between washes.

Nuclei and Particle Tracking.

Cropping and File Preparation. Time lapse video files were cropped on NIS-Elements in preparation for image processing. First, time lapse video data files were cropped into individual spheroid multipoints. Then, while keeping all timepoints, the image was cropped in X and Y to

center the spheroid and reduce file size. All images were cropped into squares to facilitate image overlays. Z-stacks that were over or below the main spheroid body and associated protrusions were cropped out. Images at the last time point were used for image processing and parameters were applied to all other time points.

Image processing. NIS-Elements time lapse video files were imported to ImageJ using the Bio formats plugin. Phase processing. Using the last time point image, the phase channel was zoomed in on a strand to find the z-plane in which the strand was in focus. A median filter was applied to the phase channel with Kernel size set to 3. SD-GFP processing. SD- GFP channel was zoomed in on a strand to find the z-plane in which the strand was in focus. A median filter was applied to the SD-GFP channel with Kernel size set to 3. Next, local contrast was adjusted using degree and radius sliders to make the image appear clearer (~ 12.8 um radius and 10 degree). The contrast between strands and the background was increased without overexposing the center. The preview button was used to make sure the filters were improving image quality. Next, an unsharp mask was applied to the SD-GFP channel. The power and area of the unsharp mask were adjusted to create a border between the strand and background (power=.51, area=3). Then, the "subtract background using constant" function was used to further decrease background noise while preserving the signal from cells on the spheroid periphery (~ 150). SD-RFP processing. The SD- RFP channel was zoomed in on a strand to find the z-plane in which the strand was in focus. Next, looking at the focused strand and spheroid core, local contrast was adjusted use degree and radius sliders to make the image appear clearer (~ 10 um radius and 30 degree). Then, an unsharp mask was applied to the SD-RFP channel. The power and area of the unsharp mask were adjusted to create a border between the strand and background (power=.4,

area=23). Next, the “subtract background using constant” function was used to further decrease background noise while preserving the signal from cells on the spheroid periphery (~ 150). Finally, advanced denoising was applied to the SD-RFP channel with the power set to ~12. Cy5 processing. To allow individual FMS particle detection, the local contrast was first adjusted with the radius set to 2 and the degree set to 2. Then, an unsharp mask was applied to the SD-CY5 channel with the area set to 3 and the power set to 0.5. Next, the “subtract background using constant” function was used to further decrease noise while being careful not to exclude the signal from any tracer particles (~800). Then, a median filter was applied to the SD-CY5 channel with Kernel size set to 3. Finally, the LUTS of each channel was adjusted to facilitate the qualitative observation of small features as well as optimize nuclei and particle signal for detection. For the SD-Cy5 channel, the LUTS was adjusted to make the FMS brighter. For the SD-RFP channel, the LUTS was adjusted to make the nuclei brighter. For the SD-GFP channel, the LUTS was adjusted to increase contrast without overexposing the spheroid core.

Custom MATLAB scripts were used for nuclei and tracer particle tracking as well as plotting strand quantifications.

```
% 2022.09.29 phase projection of strands, spheroid quantification of strand data
folder = '/Users/Alexhruska1/Downloads';
filename = 'MCF10A_strand_crop_phase_MTLB.tif';
phase_stack_10A = [];
for i = 1:41
newimg = imread(fullfile(folder,filename),i);
phase_stack_10A = cat(3,phase_stack_10A,newimg);
end
%%
imtool3D(phase_stack_10A(:,:,:))
%%
%img3D = %input 3D image stack
FileName = '10A_strand.tif'; %makes a new image
PathName = '/Users/Alexhruska1/Downloads/';
ButtonName = 'Widefield'; %'Confocal';
costval = 0.0000001;
[smooth2Dimage_10A] = smooth2Dproj2022(phase_stack_10A,FileName,PathName,ButtonName,costval);
```

```
figure,imshow(smooth2Dimage_10A)
%%
PathName = '/Users/Alexhruska1/Downloads/';
imwrite(smooth2Dimage_OHT,fullfile(PathName,'MCF10A_OHT_smooth_new.tif'));
%%
% load excel data
folder = '/Users/Alexhruska1/Desktop/PhD/Research/Vimentin Project 2021/2022.08.31 TFM Exp/';
filename = 'Spheroid_Data_20220929.xlsx';
% c1 = #single cells
% c2 = #strands
% c3 = strand length
% c4 = strand width max
% c5 = strand width min
DMSO_Data = readmatrix(fullfile(folder,filename),'Sheet','10A','OutputType','double','Range','F5:J168');
OHT_Data = readmatrix(fullfile(folder,filename),'Sheet','OHT','OutputType','double','Range','F5:J332');
DMSO_AreaInitial = readmatrix(fullfile(folder,filename),'Sheet','Area','OutputType','double','Range','D7:D13');
DMSO_AreaFinal = readmatrix(fullfile(folder,filename),'Sheet','Area','OutputType','double','Range','H7:H21');
OHT_AreaInitial = readmatrix(fullfile(folder,filename),'Sheet','Area','OutputType','double','Range','E7:E13');
OHT_AreaFinal = readmatrix(fullfile(folder,filename),'Sheet','Area','OutputType','double','Range','I7:I22');
%%
% find the locations of non-NaN data in the second column to get the number of spheroid to analyze, also number of
columns of data to look for
nanvec = isnan(DMSO_Data(:,2));
[ridx,~] = find(nanvec == 0);
for i = 1:length(ridx)
Spheroid_Data.DMSO(i).NumSingleCells = DMSO_Data(ridx(i),1);
NumStrands = DMSO_Data(ridx(i),2);
Spheroid_Data.DMSO(i).NumStrands = NumStrands;
Spheroid_Data.DMSO(i).StrandLengths = DMSO_Data(ridx(i):(ridx(i)+NumStrands-1),3);
Spheroid_Data.DMSO(i).StrandWidthMax = DMSO_Data(ridx(i):(ridx(i)+NumStrands-1),4);
Spheroid_Data.DMSO(i).StrandWidthMin = DMSO_Data(ridx(i):(ridx(i)+NumStrands-1),5);
end
%%
nanvec2 = isnan(OHT_Data(:,2));
[ridx2,~] = find(nanvec2 == 0);
for i = 1:length(ridx2)
Spheroid_Data.OHT(i).NumSingleCells = OHT_Data(ridx2(i),1);
NumStrands = OHT_Data(ridx2(i),2);
Spheroid_Data.OHT(i).NumStrands = NumStrands;
Spheroid_Data.OHT(i).StrandLengths = OHT_Data(ridx2(i):(ridx2(i)+NumStrands-1),3);
Spheroid_Data.OHT(i).StrandWidthMax = OHT_Data(ridx2(i):(ridx2(i)+NumStrands-1),4);
Spheroid_Data.OHT(i).StrandWidthMin = OHT_Data(ridx2(i):(ridx2(i)+NumStrands-1),5);
end
%% plot num single cells
DMSO_SS = [Spheroid_Data.DMSO.NumSingleCells]';
OHT_SS = [Spheroid_Data.OHT.NumSingleCells]';
All_Data = [DMSO_SS;OHT_SS];
DMSO_SS_Str = repmat({'- OHT'},length(DMSO_SS),1);
OHT_SS_Str = repmat({'+ OHT'},length(OHT_SS),1);
All_Str = [DMSO_SS_Str;OHT_SS_Str ];
figure('Renderer', 'painters', 'Position', [500 500 300 400]) %figure size %width, %height
violinplot(All_Data,All_Str,'GroupOrder',{'- OHT','+ OHT'})
set(gca,'FontSize',30,'FontWeight','Bold')
DMSO_SS_Ave = mean(DMSO_SS);
OHT_SS_Ave = mean(OHT_SS);
%% plot num strands
```

```
DMSO_Strands = [Spheroid_Data.DMSO.NumStrands]';
OHT_Strands = [Spheroid_Data.OHT.NumStrands]';
All_Data_Strands = [DMSO_Strands;OHT_Strands];
DMSO_Strands_Str = repmat({'- OHT'},length(DMSO_Strands),1);
OHT_Strands_Str = repmat({'+ OHT'},length(OHT_Strands),1);
All_Str_Strands = [DMSO_Strands_Str;OHT_Strands_Str ];
figure('Renderer', 'painters', 'Position', [500 500 300 400]) %figure size %width, %height
violinplot(All_Data_Strands,All_Str_Strands,'GroupOrder',{'- OHT','+ OHT'})
set(gca,'FontSize',30,'FontWeight','Bold')
DMSO_Strands_Ave = mean(DMSO_Strands);
OHT_Strands_Ave = mean(OHT_Strands);
%% organize length data
DMSO_Strand_Lengths = [];
for i = 1:length(Spheroid_Data.DMSO)
Strandi = [Spheroid_Data.DMSO(i).StrandLengths];
DMSO_Strand_Lengths = [DMSO_Strand_Lengths;Strandi];
end
OHT_Strand_Lengths = [];
for i = 1:length(Spheroid_Data.OHT)
Strandi = [Spheroid_Data.OHT(i).StrandLengths];
OHT_Strand_Lengths = [OHT_Strand_Lengths;Strandi];
end
%% plot length data
All_Data_Strand_Length = [DMSO_Strand_Lengths;OHT_Strand_Lengths];
DMSO_StrandL_Str = repmat({'- OHT'},length(DMSO_Strand_Lengths),1);
OHT_StrandL_Str = repmat({'+ OHT'},length(OHT_Strand_Lengths),1);

All_Str_StrandL = [DMSO_StrandL_Str;OHT_StrandL_Str ];

figure('Renderer', 'painters', 'Position', [500 500 300 400]) %figure size %width, %height
violinplot(All_Data_Strand_Length,All_Str_StrandL,'GroupOrder',{'- OHT','+ OHT'})
set(gca,'FontSize',30,'FontWeight','Bold')
DMSO_StrandL_Ave = mean(DMSO_Strand_Lengths);
OHT_StrandL_Ave = mean(OHT_Strand_Lengths);
%% organize max width data
DMSO_Strand_Widths = [];
for i = 1:length(Spheroid_Data.DMSO)
Strandi = [Spheroid_Data.DMSO(i).StrandWidthMax];
DMSO_Strand_Widths = [DMSO_Strand_Widths;Strandi];
end
OHT_Strand_Widths = [];
for i = 1:length(Spheroid_Data.OHT)
Strandi = [Spheroid_Data.OHT(i).StrandWidthMax];
OHT_Strand_Widths = [OHT_Strand_Widths;Strandi];
end
%% plot width data
All_Data_Strand_Widths = [DMSO_Strand_Widths;OHT_Strand_Widths];
DMSO_StrandW_Str = repmat({'- OHT'},length(DMSO_Strand_Widths),1);
OHT_StrandW_Str = repmat({'+ OHT'},length(OHT_Strand_Widths),1);
All_Str_StrandW = [DMSO_StrandW_Str;OHT_StrandW_Str ];
figure('Renderer', 'painters', 'Position', [500 500 300 400]) %figure size %width, %height
violinplot(All_Data_Strand_Widths,All_Str_StrandW,'GroupOrder',{'- OHT','+ OHT'})
set(gca,'FontSize',30,'FontWeight','Bold')
DMSO_StrandW_Ave = mean(DMSO_Strand_Widths);
OHT_StrandW_Ave = mean(OHT_Strand_Widths);
%% DMSO area initial and final plot
```

```
%DMSO_AreaInitial = readmatrix(fullfile(folder,filename),'Sheet','Area','OutputType','double','Range','D7:D13');
%DMSO_AreaFinal = readmatrix(fullfile(folder,filename),'Sheet','Area','OutputType','double','Range','H7:H21');
DMSO_Area_All = [DMSO_AreaInitial;DMSO_AreaFinal];
DMSO_AreaI_Str = repmat({'0 Hr'},length(DMSO_AreaInitial),1);
DMSO_AreaF_Str  = repmat({'72 Hr'},length(DMSO_AreaFinal),1);
All_Str_DMSOA = [DMSO_AreaI_Str;DMSO_AreaF_Str ];
figure('Renderer', 'painters', 'Position', [500 500 300 400]) %figure size %width, %height
violinplot(DMSO_Area_All,All_Str_DMSOA,'GroupOrder',{'0 Hr','72 Hr'})
set(gca,'FontSize',30,'FontWeight','Bold')
ylim([0 3E5])
DMSO_AreaI = mean(DMSO_AreaInitial);
DMSO_AreaF = mean(DMSO_AreaFinal);
%% OHT area initial and final plot
%OHT_AreaInitial = readmatrix(fullfile(folder,filename),'Sheet','Area','OutputType','double','Range','E7:E13');
%OHT_AreaFinal = readmatrix(fullfile(folder,filename),'Sheet','Area','OutputType','double','Range','I7:I22');
OHT_Area_All = [OHT_AreaInitial;OHT_AreaFinal];
OHT_AreaI_Str = repmat({'0 Hr'},length(OHT_AreaInitial),1);
OHT_AreaF_Str  = repmat({'72 Hr'},length(OHT_AreaFinal),1);
All_Str_OHT = [OHT_AreaI_Str;OHT_AreaF_Str];
figure('Renderer', 'painters', 'Position', [500 500 300 400]) %figure size %width, %height
violinplot(OHT_Area_All,All_Str_OHT,'GroupOrder',{'0 Hr','72 Hr'})
set(gca,'FontSize',30,'FontWeight','Bold')
ylim([0 3E5])
OHT_AreaI = mean(OHT_AreaInitial);
OHT_AreaF = mean(OHT_AreaFinal);

% segment spheroid phase images to use as a border for PIV
%% 10A
folder = '/Users/Alexhruska1/Desktop/PhD/Research/Vimentin Project 2021/2022.08.22 TFM Exp data for image correlation';
filename = '3 hr TL C3D cropz16-20 phase.tif';
FullFilename = fullfile(folder,filename);
numimgs = size(imfinfo(FullFilename),1)
%% OHT
folder = '/Users/Alexhruska1/Desktop/PhD/Research/Vimentin Project 2021/2022.08.22 TFM Exp data for image correlation';
filename = '3 hr TL D4C cropz16-20 phase.tif';
FullFilename = fullfile(folder,filename);
numimgs = size(imfinfo(FullFilename),1)

%%
%read multiple stacks per image, then per time
% 5 images per time , 38 time points
% get the third one,
vec = 3:5:numimgs;
I = [];
for i = 1:length(vec)
    Ii = imread(FullFilename,vec(i));
    I = cat(3,I,Ii);
end
imtool3D(I)
%%
%% improved watershed segm
%check 14,16 and 23
I1 = I(:,:,16);
I2 = medfilt2(I1);
```

```
gmag = imgradient(I2);
figure,imshow(gmag,[]);
%%
[~,threshold] = edge(I2,'canny');
fudgeFactor = 3;
I3 = edge(I2,'canny',threshold * fudgeFactor,sqrt(10));
figure,imshowpair(I2,I3,'montage');
%%
len = 10;
se90 = strel('line',len,90);
se0 = strel('line',len,0);
I4 = imdilate(I3,[se90 se0]);
I5 = imfill(I4,'holes');
figure,imshowpair(I3,I5,'montage');
%% now get the foreground marker by area filt and erode the shape
I8 = bwareaopen(I7,5000);
I9 = imerode(I8,strel('disk',20));
figure,imshowpair(I7,I9,'montage');
FGND = I9;
%% now compute background markers
D = bwdist(I7);
DL = watershed(D);
BGM = DL == 0;
figure,imshow(imadjust(DL))
title('Watershed Ridge Lines')
%%
gmag2 = imimposemin(gmag, BGM | FGND);
L = watershed(gmag2);
labels = imdilate(L==0,ones(3,3)) + 2*BGM + 3*FGND;
IL = labeloverlay(imadjust(I1),labels);
figure,imshow(IL)
title('Markers and Object Boundaries Superimposed on Original Image')
%% create mask from watershed segmenetation
IL2 = imfill(~logical(L),'holes');
IL3 = imopen(IL2,strel('disk',2));
IL4 = bwareaopen(IL3,5000);
figure,imshowpair(IL2,IL4,'montage')
%maski = imfill(~logical(L),'holes')
%%
B = bwboundaries(IL4,'noholes');
im = zeros(size(IL4,1),size(IL4,2));
for k = 1:length(B{1,1})
   im(B{1,1}(k,1), B{1,1}(k,2))=1;
end
%figure,imshow(im)
im2 = imdilate(im,strel('disk',3));
Ilabel = labeloverlay(imadjust(I1),im2);
figure,imshow(Ilabel)
%%
%% loop version - works well
Imask = [];
ImaskOL = [];
for i = 1:size(I,3)
I1 = I(:,:,i);
I2 = medfilt2(I1);
gmag = imgradient(I2);
```

```
[~,threshold] = edge(I2,'canny');
fudgeFactor = 3.5;
I3 = edge(I2,'canny',threshold * fudgeFactor,sqrt(10));
len = 10;
se90 = strel('line',len,90);
se0 = strel('line',len,0);
I4 = imdilate(I3,[se90 se0]);
I5 = imfill(I4,'holes');
I6 = bwareaopen(I5,5000);
FGND = imerode(I6,strel('disk',20));
D = bwdist(I5);
DL = watershed(D);
BGM = DL == 0;
gmag2 = imimposemin(gmag, BGM | FGND);
L = watershed(gmag2);
IL2 = imfill(~logical(L),'holes');
IL3 = imopen(IL2,strel('disk',2));
maski = bwareaopen(IL3,5000);
maski = ~maski;
B = bwboundaries(maski,'noholes');
im = zeros(size(maski,1),size(maski,2));
for k = 1:length(B{1,1})
   im(B{1,1}(k,1), B{1,1}(k,2))=1;
end
im2 = imdilate(im,strel('disk',3));
Ilabel = imoverlay(imadjust(I1),im2,'yellow');
Imask = cat(3,Imask,maski);
ImaskOL = cat(3,ImaskOL,Ilabel);
end
imtool3D(Imask)
%% export masks
WriteFolder = '/Users/Alexhruska1/Desktop/PhD/Research/Vimentin Project 2021/2022.08.22 TFM Exp data for
image correlation/D4C 3hr OHT/Masks';
for i = 1:size(I,3)
WriteName = strcat('D4C_3hr_frame_',num2str(i),'.tif');
imwrite(Imask(:,:,i),fullfile(WriteFolder,WriteName));
end
%% 3D cell tracking analysis from imaris
% FN inputs = folder, filename, t_step,
% script under development 2/21/23
%folder = '/Users/Alexhruska1/Desktop/PhD/Research/Vimentin Project 2021/2022.08.22 TFM Exp data for image
correlation/imaris cell tracking';
%folder = '/Users/Alexhruska1/Desktop/PhD/Research/Vimentin Project 2021/2023.02 spheroid cell tracking
analysis/2023.03.03';
%filename = 'C3D_10A_3hrTL_RFPOnly_positiondata.xlsx';
%folder = '/Users/Alexhruska1/Desktop/PhD/Research/Vimentin Project 2021/2022.08.22 TFM Exp data for image
correlation/imaris cell tracking';
folder = '/Users/Alexhruska1/Downloads/';
filename = '2023.03.08_TL_M2_10A_stats_Position.csv';
%filename = '2023_02_24_stats_Position.csv';
%filename = 'D4C_OHT_3 hrTL_RFPOnly_positiondata.xlsx';
%Data = readmatrix(fullfile(folder,filename),'Sheet','10A','OutputType','double','Range','F5:J168');
%Data = readmatrix(fullfile(folder,filename),'Sheet','Position','OutputType','double'); %c1 = x pos (um), c2 = y, c3 =
z, c7 = time, c8 = trackID
Data = readmatrix(fullfile(folder,filename),'OutputType','double'); %c1 = x pos (um), c2 = y, c3 = z, c7 = time, c8 =
trackID
```

```
%% make a data structure
UniqueTracks = unique(Data(:,8));
for i = 1:size(UniqueTracks,1);
[ridx,~] = find(Data(:,8) == UniqueTracks(i));
CellTracks(i).TrackID = UniqueTracks(i);
CellTracks(i).XYZ = Data(ridx,1:3);
CellTracks(i).Time = Data(ridx,7);
end
%% calculate velocity
t_step = 0.5; %in hours, each time frame taken 30 minutes apart - verify this, could vary by experiment
%t_step = 0.25;
for i = 1:length(CellTracks)
   for j = 1:size(CellTracks(i).XYZ,1)-1
   % calculate displacements
   del_x = CellTracks(i).XYZ(j+1,1) - CellTracks(i).XYZ(j,1);
   del_y = CellTracks(i).XYZ(j+1,2) - CellTracks(i).XYZ(j,2);
   del_z = CellTracks(i).XYZ(j+1,3) - CellTracks(i).XYZ(j,3);
   tdiff = CellTracks(i).Time(j+1,1) - CellTracks(i).Time(j,1); %number of time steps between the two points

   tot_disp = sqrt(del_x^2+del_y^2+del_z^2); % in microns

   CellTracks(i).Velocity(j,1) = tot_disp/(t_step*tdiff);
   end
end
%% RE CALCULATE THE MEAN SPHEROID POSITION AT EACH TIME POINT
% Make SpheroidData a seperate struct - one row for each timepoint - Keep vel corr and other data in here
MaxTime = max(Data(:,7));
%for each timepoint, find all cells, then average each X,Y,Z coordinate
MeanSpheroidCentroid = [];
for i = 1:MaxTime
[ri,~] = find(Data(:,7)==i);
meanX = mean([Data(ri,1)]);
meanY = mean([Data(ri,2)]);
meanZ = mean([Data(ri,3)]);
MeanSpheroidCentroid = [MeanSpheroidCentroid;meanX,meanY,meanZ];
SpheroidData(i).MeanSpheroidCentroid = [meanX,meanY,meanZ];
end
%% transform coordinates to make spheroid centroid at the origin, then transform cartesian to polar coordinates
numTracks = length(CellTracks);
for i = 1:numTracks
   for k = 1:size(CellTracks(i).XYZ,1) %for each timepoint per cell
   %get the mean centroid coords from the timepoint
   tk = CellTracks(i).Time(k,1);
   Xtk = CellTracks(i).XYZ(k,1) - MeanSpheroidCentroid(1,1); % ACTUALLY just use the centroid of the FIRST
time point for all so that as sph invades the dist stays constant
   Ytk = CellTracks(i).XYZ(k,2) - MeanSpheroidCentroid(1,2);
   Ztk = CellTracks(i).XYZ(k,3) - MeanSpheroidCentroid(1,3);
   CellTracks(i).CtdXYZ(k,:) = [Xtk,Ytk,Ztk]; %Centered XYZ
   [azimuth,elevation,radius] = cart2sph(Xtk,Ytk,Ztk);
   CellTracks(i).SphCoords(k,:) = [azimuth,elevation,radius]; %theta reported in radians
   %also calculate the distance from the spheroid centroid for each cell at every timestep
   %calculate distance using adjusted XYZ coords that centers each cell making the spheroid centroid the origin of
the reference frame, so dist is meas from sph centroid also the origin
     distk = sqrt(Xtk^2+Ytk^2+Ztk^2);
   CellTracks(i).DistFromSphCentroid(k,1) = distk;
   end
```

```
    % calculate average cell props
    CellTracks(i).TotAngDisp =
abs(rad2deg(angdiff(CellTracks(i).SphCoords(end,1),CellTracks(i).SphCoords(1,1)))); %total ang disp in degrees
    % calculate velocity of polar coords
    for j = 1:size(CellTracks(i).XYZ,1)-1
    tdiff = CellTracks(i).Time(j+1,1) - CellTracks(i).Time(j,1); %time diff btw points - either 1 or 2 frames (no more
than 2 tho)
    CellTracks(i).Azimuth_Vel(j,1) =
rad2deg(angdiff(CellTracks(i).SphCoords(j,1),CellTracks(i).SphCoords(j+1,1)))/(t_step*tdiff); %degrees per hour
    CellTracks(i).Elevation_Vel(j,1) =
rad2deg(angdiff(CellTracks(i).SphCoords(j,2),CellTracks(i).SphCoords(j+1,2)))/(t_step*tdiff);  %microns per hour
    CellTracks(i).Radius_Vel(j,1) =  (CellTracks(i).SphCoords(j+1,3)-CellTracks(i).SphCoords(j,3))/(t_step*tdiff);
%microns per hour
    end
end
for i = 1:MaxTime


% need to get all the XYZ points for cells at a given time point
    iXYZ = [];
    indices = [];
    for j = 1:length(CellTracks)
    times = CellTracks(j).Time;
    if sum(ismember(times,i)) == 1
    [ridx,~] = find(ismember(times,i)==1);
    xyz = CellTracks(j).CtdXYZ(ridx,:);
    %also store the cell track index for the centroid
    iXYZ = [iXYZ;xyz]; %compile the CtdXYZ points for each time
    indices = [indices;j];
    end
    end
%store a list of all the XYZ coords for all cells at each timepoint to use later
SpheroidData(i).AllCtdXYZ = iXYZ;
SpheroidData(i).AllCtdIndices = indices;
[TriIdx1, V] = convhull(iXYZ(:,1),iXYZ(:,2),iXYZ(:,3));
%figure
%trisurf(TriIdx1, iXYZ(:,1),iXYZ(:,2),iXYZ(:,3))
% now find XYZ points within +/- 5 um from z=0 centroid
[rc,~] = find(iXYZ(:,3) > -5 & iXYZ(:,3) < 5);
[TriIdx, CSa] = convhull(iXYZ(rc,1),iXYZ(rc,2));
%[TriIdx2, V2] = convhull(iXYZ(rc,1),iXYZ(rc,2),iXYZ(rc,3));
%figure
%trisurf(TriIdx2, iXYZ(rc,1),iXYZ(rc,2),iXYZ(rc,3))
out = CircleFitByPratt(iXYZ(rc,1:2)); % out [a b R] is the fitting circle: center (a,b) and radius R
SpheroidData(i).Volume = V;
SpheroidData(i).CSArea = CSa;
SpheroidData(i).EqRadius = out(3);
% use convhull at the centroid - find all points with Z coords of +/- 5um from z = 0, then only use those XY coords
to get conv hull CS area
% use this to try to calculate an equivalent radius at each timestep from a circular fit of the boundary
end
%% how much does the spheroid centroid position change over time?
% we calculate the relative XYZ coords relative to centroid at each timestep - I use these to calculate the polar
coordinates so that angles are relative to the centroid
% should we keep the spheroid centroid position fixed across all time of change it at each timestep?
% plot change in overall XYZ distance over time
```

```
dist = [];
for i = 1:length(SpheroidData)-1
diff = SpheroidData(i+1).MeanSpheroidCentroid - SpheroidData(i).MeanSpheroidCentroid;
disti = sqrt(sum(diff.^2));
dist = [dist;disti];
end
figure
plot(1:length(SpheroidData)-1,dist,'LineWidth',2)
%% calculate a mean spheroid centroid across all time, then plot dist to mean centroid at each time
% mean centroid
mat = [];
for i = 1:length(SpheroidData)
mati = [SpheroidData(i).MeanSpheroidCentroid];
mat = [mat;mati];
end

MeanCentroid = mean(mat,1);
%%
dist = [];
for i = 1:length(SpheroidData)-1
diff = SpheroidData(i).MeanSpheroidCentroid - MeanCentroid;
disti = sqrt(sum(diff.^2));
dist = [dist;disti];
end
figure
plot(1:length(SpheroidData)-1,dist,'LineWidth',2)
%% get the number of cells across the equator of the sph centroid
addpath('point_to_line_distance/')
% get coords of a 2D line for a given angle, then sweep angle from 0 to 90
% make the line go from the centroid to the EqRadius
for t = 2:length(SpheroidData) %for each timestep with a velocity associated
CtdList = SpheroidData(t).AllCtdXYZ;
Rad = SpheroidData(t).EqRadius;
a = 10:10:360;
clear angs
angs = cell(length(a),4); %stores the indexes of cells that are within the line for each angle
% c1 = cell index from CELL TRACKS at each rotation angle
% c2 = NumCellsAlongLine for each rotation angle
% c3 = Relative angle for each cell intersecting the line at each rotation angle
% c4 = radius of each cell that a rel angle was calculated for
   cnt = 0;
   for ang = 10:10:360 %each angle from 0 to 90
   cnt = cnt + 1;
   v1 = [0,0,0]; %line is at z = 0 - the spheroid centroid
   v2 = [2*Rad*cosd(ang),2*Rad*sind(ang),0];
   distance_3D = point_to_line_distance(CtdList, v1, v2);
   [ridx,~] = find(distance_3D < 10);
   NumCellsAlongLine = length(ridx);
   angs{cnt,1} = [SpheroidData(t).AllCtdIndices(ridx,1)]'; %this gets the cell tracks index
   angs{cnt,2} = NumCellsAlongLine;
   % get the indexes of the cells that are along the line
   % with these cells, can get the NUMBER of cells for the simulation
   % can also calcualte the RELATIVE ANGLE = angular velocity (azimuth) orientation (diff of angles?)  - current
azimuth angle
      %relative angle caclated for each cell that intesectes each line
      %cellpos = [];
```

```
        %velang = [];
        relangs = [];
        cellrad = [];
        for r = 1:size(ridx,1); %for each cell that the line intersected
         %need to get the coordinate value with the right time index
        cellidx = SpheroidData(t).AllCtdIndices(ridx(r),1);
        [tidx1,~] = find(CellTracks(cellidx).Time == t-1);
        [tidx2,~] = find(CellTracks(cellidx).Time == t);
        CellAngt1 = CellTracks(cellidx).SphCoords(tidx1,1); %col 1 is azimuth
        CellAngt2 = CellTracks(cellidx).SphCoords(tidx2,1);
        CellPos1 = CellTracks(cellidx).CtdXYZ(tidx1,:);
        CellPos2 = CellTracks(cellidx).CtdXYZ(tidx2,:);

        if ~isempty(CellAngt1) & ~isempty(CellAngt2)

        %VelAngle = angdiff(CellAngt1,CellAngt2); %velocity is calculated from the previous time step
        dx = CellPos2(1,1)-CellPos1(1,1);
        dy = CellPos2(1,2)-CellPos1(1,2);
        %dz = CellPos2(1,3)-CellPos1(1,3); %NOT USING Z component here
        VelAngle = atan2(dy,dx);
        %VelAngle
        RelAngle = rad2deg(angdiff(CellAngt2,VelAngle)); %VelAngle - Current Angle,
        CellRadius = CellTracks(cellidx).SphCoords(tidx2,3),
        else
        RelAngle = NaN; %means for the given time step, the cell had no previous time point to calc a vel vec from
        end
        relangs = [relangs;RelAngle];
        cellrad = [cellrad;CellRadius];
        end
    angs{cnt,3} = relangs;
    angs{cnt,4} = cellrad;
    end
    SpheroidData(t).CellsAlongLine = angs;
end
%% now plot the values:
% c1 = cell index from CELL TRACKS at each rotation angle
% c2 = NumCellsAlongLine for each rotation angle
% c3 = Relative angle for each cell intersecting the line at each rotation angle
% c4 = radius of each cell that a rel angle was calculated for
angs = 10:10:360;
count = 0;
figure('Renderer', 'painters', 'Position', [200 200 2000 2000])
%2 - 69
%70 - 137
for time = 2:69
count = count+1;
subplot(8,9,count)
for i = 1:length(angs)
x = SpheroidData(time).CellsAlongLine{i,4}; %radius values
y = SpheroidData(time).CellsAlongLine{i,3}; %relative angles
if sum(isnan(y)) >= 1
[nidx,~] = find(isnan(y)==1);
y(nidx) = [];
x(nidx) = [];
end
hold on
```

```
scatter(x,y,30,''blue'','filled')
%histogram(y)
hold off
%title(strcat('angle =',num2str(angs(i))))
end
xlabel('Radius (microns)')
ylabel('Relative Angle (degrees)')
title(strcat('Time == ',num2str(time)))
xlim([0 100])
ylim([-180 180])
end


%calculate relative angle for each cell
% for each time point, want to find all cells with a
MaxTime = length(SpheroidData);
RelAng = cell(MaxTime-1,1); %Time-1 pts bc getting rel ang from velocity
% at each time want to find all cells that have a position at t=2 and t=1
for i = 1:length(CellTracks) %check each cell
   RelAngMat = []; %first column is relative angle, second column is the radius
   for t = 2:MaxTime
   % check if the cell has a time = t and time = t-1, if so proceed and use the indices to get the cell position data too
   [t1idx,~] = find(CellTracks(i).Time == t-1);
   [t2idx,~] = find(CellTracks(i).Time == t);
   if ~isempty(t1idx) & ~isempty(t2idx) %if the cell has a position for both times, continue
      CellAngt1 = CellTracks(i).SphCoords(t1idx,1); %col 1 is azimuth
      CellAngt2 = CellTracks(i).SphCoords(t2idx,1);
      CellPos1 = CellTracks(i).CtdXYZ(t1idx,:);
      CellPos2 = CellTracks(i).CtdXYZ(t2idx,:);
      dx = CellPos2(1,1)-CellPos1(1,1);
      dy = CellPos2(1,2)-CellPos1(1,2);
      VelAngle = atan2(dy,dx);
      RelAngle = rad2deg(angdiff(CellAngt2,VelAngle)); %VelAngle - Current Angle,
      CellRadius = CellTracks(i).SphCoords(t2idx,3);
      RelAngMat = [RelAngMat;RelAngle,CellRadius];
   end
   end
%store for each cell
CellTracks(i).RelAng = RelAngMat;
end
%% make a cell for all the relative angles and radii at each time step
RelAngCell = cell(MaxTime-1,1);
for t = 2:MaxTime
tvec = []; %collect data for each time point
   for i = 1:length(CellTracks)
   [t1idx,~] = find(CellTracks(i).Time == t-1);
   [t2idx,~] = find(CellTracks(i).Time == t);
   if ~isempty(t1idx) & ~isempty(t2idx) %if the cell has a pos for both times, continue (means it has a rel ang
calculated)
   relang = CellTracks(i).RelAng(t2idx-1,1);
   cellradius = CellTracks(i).RelAng(t2idx-1,2);
   tvec = [tvec;relang,cellradius];
   end
   end
   %store for each time after we have looped thru all cells
   RelAngCell{t-1,1} = tvec;
```

```
end
%% now plot all cells rel angles
figure('Renderer', 'painters', 'Position', [200 200 2000 2000])
% 2 - 46
% 47 - 93
% 94 - 138
count = 0;
for t = 94:138
count = count+1;

x = RelAngCell{t-1,1}(:,1); %rel angle vec
y = RelAngCell{t-1,1}(:,2); %radii vec
subplot(7,7,count)
%subplot(1,1,count)
scatter(y,x,30,"blue",'filled')
xlabel('Radius (\mum)')
ylabel('Rel Ang (deg)')
title(strcat('Time == ',num2str(t)))
xlim([0 100])
ylim([-180 180])
end
%% quiver plot sanity check
% get all cells that have a velocity vector at time = 2 --> so make sure it has time==2 and that t==2 is not the first index (so there is a time before it, and a vel vec)
%store the X and Y coordinate as well as the deltaX and deltaY from the vel vec (just get from the previous position)
time = 3;
figure
cellcount = 0;
for i = 1:length(CellTracks)
tvec = CellTracks(i).Time;
[ridx,~] = find(tvec == time);
if ~isempty(ridx) & ridx ~= 1
cellcount = cellcount + 1;
%store the coords
pos1 = CellTracks(i).CtdXYZ(ridx-1,:);
pos2 = CellTracks(i).CtdXYZ(ridx,:); %plot pos 2
delx = pos2(1,1) - pos1(1,1);
dely = pos2(1,2) - pos1(1,2);
hold on
quiver(pos2(1,1),pos2(1,2),delx,dely,3)
hold off
end
end
%quiver(X,Y,U,V) %plots arrows with directional components U and V at the Cartesian coordinates specified by X and Y.
%% we want to color the cell nuclei in an image (scatter) by their velocity vector
%red for +90, blue for -90
% find all cells at timepoint 2 that have a velocity vector
% plot scatter plot of nuclei XY positions (can filter for Z=0 or not the entire spheroid)
addpath('customcolormap')
mycolormap = customcolormap([0 1], [1 0 0; 0 0 1]); % red --> blue
mycolormap2 = [mycolormap;flip(mycolormap)];
mycolormap2(end,:) = [];
anglevec1 = linspace(-180,0,size(mycolormap,1))';
anglevec2 = linspace(0,180,size(mycolormap,1))';
```

```
anglevec3 = [flip(anglevec1);flip(anglevec2)];
anglevec3(end,:) = [];
% actually 90 to 180 to -180 to -90 to be
%goes from -180 to + 180 in matlab coordinates
% colormap goes for
time = 4;
figure('Renderer', 'painters', 'Position', [20 20 1500 1400])

count = 0;
for i = 1:length(CellTracks)
tvec = CellTracks(i).Time;
[ridx,~] = find(tvec == time);
if ~isempty(ridx) & ridx ~= 1
count = count + 1;
%store the coords
pos1 = CellTracks(i).CtdXYZ(ridx-1,:);
pos2 = CellTracks(i).CtdXYZ(ridx,:); %plot pos 2
delx = pos2(1,1) - pos1(1,1);
dely = pos2(1,2) - pos1(1,2);
velang = atan2d(dely,delx);
v = abs(anglevec3 - velang); %look for the minimum here
[~,ridx] = min(v);
hold on
scatter(pos2(1,1),pos2(1,2),60,mycolormap2(ridx,:),'filled')
quiver(pos2(1,1),pos2(1,2),delx,dely,3,'linewidth',3,'color',mycolormap2(ridx,:))
hold off
%axis equal
end
end
set(gca, 'YDir','reverse')
title(strcat('Time = ',num2str(time)))
xlim([-125 125])
ylim([-125 125])
xlabel('X Position (\mum)')
ylabel('Y Position (\mum)')
set(gca,'FontSize',15)
%depath = '/Users/Alexhruska1/Desktop/PhD/Research/Vimentin Project 2021/2023.02 spheroid cell tracking
analysis/IanFiles/';
%% WITH Ian's DATA: we want to color the cell nuclei in an image (scatter) by their velocity vector
% Using Ian's data: PosX, PosY,  CellAngle (389x101 mat)
%red for +90, blue for -90
% find all cells at timepoint 2 that have a velocity vector
% plot scatter plot of nuclei XY positions (can filter for Z=0 or not the entire spheroid)
addpath('customcolormap')
% angles go from 0 to 2 pi
mycolormap = customcolormap([0 1], [1 0 0; 0 0 1]); % red --> blue
mycolormap2 = [mycolormap;flip(mycolormap)];
mycolormap2(end,:) = [];
anglevec1 = linspace(-180,0,size(mycolormap,1))';
anglevec2 = linspace(0,180,size(mycolormap,1))';
anglevec3 = [flip(anglevec1);flip(anglevec2)];
anglevec3(end,:) = [];
% actually 90 to 180 to -180 to -90 to be
%goes from -180 to + 180 in matlab coordinates
% colormap goes for
time = 30;
```

```
figure('Renderer', 'painters', 'Position', [20 20 1500 1400])
% PLOT FIRST using the polarity angle supplied, THEN plot using the velocity angle calculated
for i = 1:size(PosX,1) % plot cell by cell
pos1 = [PosX(i,time-1),PosY(i,time-1)]; %position at t-1 = time - 1 , %x,y
pos2 = [PosX(i,time),PosY(i,time)]; %position at t = time,  %x,y

delx = pos2(1,1) - pos1(1,1);
dely = pos2(1,2) - pos1(1,2);
%velang = atan2d(dely,delx);
velang = wrapTo180(rad2deg(CellAngle(i,time))); %need to convert radians to degrees, wrap to 180
v = abs(anglevec3 - velang); %look for the minimum here
[~,ridx] = min(v);
hold on
scatter(pos2(1,1),pos2(1,2),60,mycolormap2(ridx,:),'filled')
quiver(pos2(1,1),pos2(1,2),delx,dely,3,'linewidth',3,'color',mycolormap2(ridx,:))
hold off
end
title('Using polarity angle for quiver')
%% sample color data points
colorMap = flip(parula(300));
[row,col] = find(BWRotVimentin==1); %only fill pixels in the BW image
for i = 1:length(row)
 % first match the value of the pixel (angle) to a colormap idx
ang = OimgAF(row(i),col(i));
[~,idx] = min(abs(vec-ang));
BWcol(row(i),col(i),1) = colorMap(idx,1);
BWcol(row(i),col(i),2) = colorMap(idx,2);
BWcol(row(i),col(i),3) = colorMap(idx,3);
end
figure,imshow(BWcol)
colormap(colorMap)
h = colorbar
caxis([0 90])
set( h, 'YDir', 'reverse' );
%% first see how many cells have tracks with and without any gaps just to eval how many "quality" cell tracks we
have - also should only use tracks that are at least 10 frames long
% all cells have at least 1 gap of tdiff = 2 , but none have tdiff>2.
CellsNoBreak = [];
CellsBreak = [];
for i = 1:length(CellTracks)
gaps = CellTracks(32).Time(2:end,1)-CellTracks(32).Time(1:end-1,1);
[r,~] = find(gaps>2);
if isempty(r) %no gaps >1 detected
CellsNoBreak = [CellsNoBreak;i];
elseif ~isempty(r) %gaps detected
CellsBreak = [CellsBreak;i];
end
end
NCells = length(CellsNoBreak)+length(CellsBreak); %should be equal to total num cells
fracNoBreak = length(CellsNoBreak)/NCells
fracBreak = length(CellsBreak)/NCells
%% calcualte vel autocorrealtion for cartesian and polar coords: angular,radial,Z velocity autocorrelation from polar
coords
%ONLY calculate this quantity for cells with a track length of 10 or more timepoints
%for each cell the max delta t that can be acheived should be calculated - for each cell need a delta t column and
then the associated autocorr
```

```
tstep = 0.5; %in hours
MinNumFrames = 10;
for i = 1:length(CellTracks)
if size(CellTracks(i).Time,1) >= MinNumFrames

% for each cell define a max delta t that we can evaluate over:
MaxDelT = max(CellTracks(i).Time) - min(CellTracks(i).Time) - 1; %-1 bc velocities have t-1 time points
assocaited with them
%now make a vector that represents each time step, delt = 1,2, up to MaxDelT
tdiff = 1:MaxDelT;
   clear AutoCorri
   AutoCorri = cell(size(tdiff,2),10);
   %c1 = del t
   %c2 = time pairs to eval v corr over
   %c3 = vcorr for each pair (cartesian)
   %c4 = mean vcorr
   %c5 = angular vcorr for each pair
   %c6 = mean angular vcorr
   %c7 = radial vcorr for each pair
   %c8 = mean radial vcorr
   %c9 = z vcorr for each pair
   %c10 = mean z vcorr
   %fill in first column with tdiffs
   for n = 1:size(tdiff,2)
   AutoCorri{n,1} = tdiff(1,n);
   end
% for each delta t, need to get cell list of all possible timepoint pairs (not double counting any)
   tveci = CellTracks(i).Time(2:end,1); %for every time point that has a velocity with it, so every one except first
time point
   for j = 1:size(tdiff,2)-1 %for the number of diff time step
   tdiffj = tveci(j+1:end,1) - tveci(1:end-j,1)
      for k = 1:size(tdiffj,1) %store each value of tdiff
      if isempty(AutoCorri{tdiffj(k,1),2})
      AutoCorri{tdiffj(k,1),2} = [tveci(k,1),tveci(k+j,1)]; %list two time point pairs, add to list that's already there
      else
      AutoCorri{tdiffj(k,1),2} = [AutoCorri{tdiffj(k,1),2};tveci(k,1),tveci(k+j,1)];
      end
      end
   end
% now that we have all the correct time point pairs in the second column of the AutoCorri cell, we can evaluate the
vel corrs btw these 2 time points
   for n = 1:size(tdiff,2) %for each
      for k = 1:size(AutoCorri{n,2}) %for the number of pairs at each time point
      % now we need to get the v components (x,y,z) associated with each time point
      [rt1,~] = find(CellTracks(i).Time==AutoCorri{n,2}(k,1)); %get the index for the right time point
      [rt2,~] = find(CellTracks(i).Time==AutoCorri{n,2}(k,2));
      dt1 =  CellTracks(i).Time(rt1,1)-CellTracks(i).Time(rt1-1,1);
      v1x = (CellTracks(i).CtdXYZ(rt1,1)-CellTracks(i).CtdXYZ(rt1-1,1))/(dt1*tstep);
      v1y = (CellTracks(i).CtdXYZ(rt1,2)-CellTracks(i).CtdXYZ(rt1-1,2))/(dt1*tstep);
      v1z = (CellTracks(i).CtdXYZ(rt1,3)-CellTracks(i).CtdXYZ(rt1-1,3))/(dt1*tstep);
      dt2 =  CellTracks(i).Time(rt2,1)-CellTracks(i).Time(rt2-1,1);
      v2x = (CellTracks(i).CtdXYZ(rt2,1)-CellTracks(i).CtdXYZ(rt2-1,1))/(dt2*tstep);
      v2y = (CellTracks(i).CtdXYZ(rt2,2)-CellTracks(i).CtdXYZ(rt2-1,2))/(dt2*tstep);
      v2z = (CellTracks(i).CtdXYZ(rt2,3)-CellTracks(i).CtdXYZ(rt2-1,3))/(dt2*tstep);
      %make vel vecs into vel unit vecs by normalizing by mean squared average
      v1N = [v1x,v1y,v1z]/sqrt(v1x^2+v1y^2+v1z^2);
```

```
    v2N = [v2x,v2y,v2z]/sqrt(v2x^2+v2y^2+v2z^2);
    vcorr = dot(v1N,v2N);

    if isempty(AutoCorri{n,3})
    AutoCorri{n,3} = vcorr;
    else
    AutoCorri{n,3} = [AutoCorri{n,3};vcorr];
    end
    %also get angular vcorr
    v1a = angdiff(CellTracks(i).SphCoords(rt1-1,1),CellTracks(i).SphCoords(rt1,1))/(dt1*tstep);
    v2a = angdiff(CellTracks(i).SphCoords(rt2-1,1),CellTracks(i).SphCoords(rt2,1))/(dt2*tstep);
    vacorr = cos(angdiff(v1a,v2a));
    %vacorr = v1a*v2a;
    if isempty(AutoCorri{n,5}) %5,6=mean
    AutoCorri{n,5} = vacorr;
    else
    AutoCorri{n,5} = [AutoCorri{n,5};vacorr];
    end
    v1r = (CellTracks(i).SphCoords(rt1,2)-CellTracks(i).SphCoords(rt1-1,2))/(dt1*tstep);
    v2r = (CellTracks(i).SphCoords(rt2,2)-CellTracks(i).SphCoords(rt2-1,2))/(dt2*tstep);
    vrcorr = v1r*v2r;
    if isempty(AutoCorri{n,7}) %7,8=mean
    AutoCorri{n,7} = vrcorr;
    else
    AutoCorri{n,7} = [AutoCorri{n,7};vrcorr];
    end
    v1z = (CellTracks(i).SphCoords(rt1,3)-CellTracks(i).SphCoords(rt1-1,3))/(dt1*tstep);
    v2z = (CellTracks(i).SphCoords(rt2,3)-CellTracks(i).SphCoords(rt2-1,3))/(dt2*tstep);
    vzcorr = v1z*v2z;
    if isempty(AutoCorri{n,9}) %9,10=mean
    AutoCorri{n,9} = vzcorr;
    else
    AutoCorri{n,9} = [AutoCorri{n,9};vzcorr];
    end
    end
  %once we get all the vcorr's for each delt, average them
  AutoCorri{n,4} = mean(AutoCorri{n,3}); %should we use circ_mean since this is the product of two angular
velocities?
  AutoCorri{n,6} = mean(AutoCorri{n,5});
  AutoCorri{n,8} = mean(AutoCorri{n,7});
  AutoCorri{n,10} = mean(AutoCorri{n,9});
  end
% save value to CellTracks
% do not save cell values that are blank for a certain t step
%standard v corr
ensemble_vc = [];
for n = 1:size(tdiff,2)
if ~isempty(AutoCorri{n,4})
ensemble_vc = [ensemble_vc;AutoCorri{n,1}*tstep,AutoCorri{n,4}];
end
end
CellTracks(i).VAutoCorr = ensemble_vc;
%angular v corr
ensemble_va = [];
for n = 1:size(tdiff,2)
if ~isempty(AutoCorri{n,6})
```

```
ensemble_va = [ensemble_va;AutoCorri{n,1}*tstep,AutoCorri{n,6}];
end
end
CellTracks(i).VAngAutoCorr = ensemble_va;
% radial vel corr
ensemble_vr = [];
for n = 1:size(tdiff,2)
if ~isempty(AutoCorri{n,8})
ensemble_vr = [ensemble_vr;AutoCorri{n,1}*tstep,AutoCorri{n,8}];
end
end
CellTracks(i).VRAutoCorr = ensemble_vr;
% z vel corr
ensemble_vz = [];
for n = 1:size(tdiff,2)
if ~isempty(AutoCorri{n,10})
ensemble_vz = [ensemble_vz;AutoCorri{n,1}*tstep,AutoCorri{n,10}];
end
end
CellTracks(i).VZAutoCorr = ensemble_vz;
end
end
%% look at mean dist from sph centroid to get an idea of the cutoffs we should use
d = [];
for i = 1:length(CellTracks)
d = [d;mean(CellTracks(i).DistFromSphCentroid)];
end
figure
histogram(d)
%% now plot autocorrelations for cells , ensemble average
% maybe bin cell groups by their mean dist from centroid?
t = 0.5:0.5:18;
conv = 1/tstep;
prop = 'VAngAutoCorr';
%prop = 'VZAutoCorr';
ensemble1 = cell(size(t,2),2);
for i = 1:size(t,2)
ensemble1{i,1}=t(i);
end
count1 = 0;
for i = 1:length(CellTracks)
if ~isempty(CellTracks(i).VAutoCorr) & CellTracks(i).TotAngDisp > 60 &
mean(CellTracks(i).DistFromSphCentroid) > 70
count1 = count1 + 1;
%need to plot the ensemble average - for each vec, make a
   for j = 1:size(CellTracks(i).(prop),1)
   if isempty(ensemble1{CellTracks(i).(prop)(j,1)*conv,2})
   ensemble1{CellTracks(i).(prop)(j,1)*conv,2} = CellTracks(i).(prop)(j,2);
   else
   ensemble1{CellTracks(i).(prop)(j,1)*conv,2} =
[ensemble1{CellTracks(i).(prop)(j,1)*conv,2};CellTracks(i).(prop)(j,2)];
   end
   end
end
end
count1
```

```
emean1 = NaN(size(t,2),1);
for i = 1:size(t,2)
emean1(i,1) = nanmean(ensemble1{i,2});
end
emean1 = [1;emean1];
% next ensemble
ensemble2 = cell(size(t,2),2);
for i = 1:size(t,2)
ensemble2{i,1}=t(i);
end
count2 = 0;
for i = 1:length(CellTracks)
if ~isempty(CellTracks(i).VAutoCorr) & CellTracks(i).TotAngDisp < 30 &
mean(CellTracks(i).DistFromSphCentroid) < 40
count2 = count2 + 1;
%need to plot the ensemble average - for each vec, make a
   for j = 1:size(CellTracks(i).(prop),1)
   if isempty(ensemble2{CellTracks(i).(prop)(j,1)*conv,2})
   ensemble2{CellTracks(i).(prop)(j,1)*conv,2} = CellTracks(i).(prop)(j,2);
   else
   ensemble2{CellTracks(i).(prop)(j,1)*conv,2} =
[ensemble2{CellTracks(i).(prop)(j,1)*conv,2};CellTracks(i).(prop)(j,2)];
   end
   end
end
end
count2
emean2 = NaN(size(t,2),1);
for i = 1:size(t,2)
emean2(i,1) = nanmean(ensemble2{i,2});
end
emean2 = [1;emean2];
t = [0,t];
figure('Renderer', 'painters', 'Position', [500 500 1200 800]) %figure size
hold on
plot(t',emean1,'LineWidth',3,'color','r','DisplayName','Cells at spheroid periphery, with high angular displacment')
plot(t',emean2,'LineWidth',3,'color','b','DisplayName','Cells at spheroid core, with low angular displacment')
hold off
xlabel('Time Delay (hours)');
ylabel('Velocity Autocorrelation (hours)');
%xlim([0 8])
set(gca,'fontsize',16)
legend
%figure
%plot(t',emean,'LineWidth',3)
%% look at angular vel and radial vel over time
% angular vel should be high at the beg and fade over time, whereas radial vel should increase over time
figure
for i = 1:30%length(CellTracks)
subplot(2,1,1)
hold on
plot(CellTracks(i).Time(2:end),CellTracks(i).Rho_Vel,'LineWidth',2)
hold off
title('Radial Veloicty')
subplot(2,1,2)
hold on
```

```
plot(CellTracks(i).Time(2:end),CellTracks(i).Theta_Vel,'LineWidth',2)
hold off
ylim([-20 20])
title('Angular Velocity (degrees/hour)')
end
figure
data = [];
for i = 1:length(CellTracks)
data = [data;CellTracks(i).DistFromSphCentroid(:)];
end
histogram(data)
%% plot velocity at each time step with distance from centroid at each time step
%even for OHT it doesn't seem like there's a trend (maybe very slight) - but this makes me question how good the cell tracking is...
% is there ANY way to validate 3D cell tracking?
% SEEMS like most cells maintain their dist from centroid thru time - maybe 10% of cells significantly increase or decrease their position thru time
% for the cells who experience a net increase towards edge - do they inc their vel? Do ones that experience neg movement decrease their velocity
% maybe compare vel of cells on the EDGE of spheroid with those very close to sph center
   % 2 subplots - one with close to spheroid, one far away, then plot
   % maybe need to look at cells on the edge that have larger ang disp? idk
% DOESNT APPEAR that there is a trend between cells distance from the spheroid centroid and the cells velocity (which is a bit surprising)
%pair the first velocity with the second dist value
%will also be able to see if cells are moving away or closer to cell centroid over time
% plot only cells with at least 10 frames
%NumTracksToPlot = 46;
NumFramesPerCell = [];
MeanDist = [];
for i = 1:length(CellTracks)
NumFramesPerCell = [NumFramesPerCell;size(CellTracks(i).XYZ,1)];
MeanDist = [MeanDist;mean(CellTracks(i).DistFromSphCentroid)];
end
[r1,~] = find(NumFramesPerCell >= 10);
[r2,~] = find(MeanDist >= 80); %mean dist from sph centroid
r = intersect(r1,r2);
[r3,~] = find(MeanDist <= 40); %mean dist from sph centroid
rb = intersect(r1,r3);
NumTracksToPlot = min(length(r),length(rb));
rtoplot1 = randperm(length(r),NumTracksToPlot);
rtoplot2 = randperm(length(rb),NumTracksToPlot);
clrs = distinguishable_colors(NumTracksToPlot);
figure
for i = 1:NumTracksToPlot
x1 = rad2deg(circ_mean(deg2rad(CellTracks(r(rtoplot1(i))).Theta_Vel(:,1))));
y1 = mean(CellTracks(r(rtoplot1(i))).Rho_Vel(:,1));
%z = mean(CellTracks(r(rtoplot(i))).DistFromSphCentroid(:,1));
%x = CellTracks(r(rtoplot(i))).Time(:,1)*0.5;
%y = CellTracks(r(rtoplot(i))).DistFromSphCentroid(:,1);
x2 = rad2deg(circ_mean(deg2rad(CellTracks(rb(rtoplot2(i))).Theta_Vel(:,1))));
y2 = mean(CellTracks(rb(rtoplot2(i))).Rho_Vel(:,1));
hold on
%scatter(x,y,30,'filled');
scatter(x1,y1,30,'r','filled');
scatter(x2,y2,30,'b','filled');
```

```
hold off
end
%xlabel('Distance from Spheroid Centroid (microns)')
%ylabel('Z Velocity (microns/hour)')
xlabel('Angular Velocity (degrees/hour)')
ylabel('Radial Velocity (microns/hour)')
%zlabel('Distance from Spheroid Centroid (microns)')
%zlabel('')
set(gca,'fontsize',15)
legend('Cells at periphery > 80um from sph center','Cells in core < 40um from sph center')
NetChange = [];
MeanV = [];
for i = 1:length(CellTracks)
   if size(CellTracks(i).XYZ,1) >= 15
   chngi = CellTracks(i).DistFromSphCentroid(end,1) - CellTracks(i).DistFromSphCentroid(1,1);
   Vi = mean(CellTracks(i).DistFromSphCentroid(:,1));
   MeanV = [MeanV;Vi];
   NetChange = [NetChange;chngi];
   end
end
figure
scatter(MeanV,NetChange,30,'filled')
ylabel('Net Change in Distance from Spheroid Centroid (microns)')
xlabel('Mean Distance from Spheroid Centroid (microns)')
%histogram(NetChange)
%xlabel('Counts')
%ylabel('Net Change in Distance from Spheroid Centroid (microns)')
set(gca,'fontsize',15)
%% plot mean ang vel, radial vel, vs dist from  sph
%% plot spheroid volume over time
t = (1:length(SpheroidData))*0.5
figure
yyaxis left
plot(t,[SpheroidData.Volume],'LineWidth',3)
ylabel('Spheroid volume (microns^3)')
yyaxis right
plot(t,[SpheroidData.CSArea],'LineWidth',3)
ylabel('Spheroid cross sectional area (microns^2)')
xlabel('Time (hours)')
set(gca,'fontsize',15)
% for each time point, calculate the velocity of cells as a fn of their dist from centroid
% every point is the same color for same time point
% could try to filter cells and not count any cell that contains a vel >40 because this most likely isnt real and means
the cell wasn't tracked properly.
%% calculate velocity distribution
Vel = [];
for i = 1:length(CellTracks)
Veli = CellTracks(i).Velocity(:);
Vel = [Vel;Veli];
end
figure,histogram(Vel,'BinWidth',5)
xlabel('Velocity (\mum)','FontSize',16)
ylabel('Probability','FontSize',16)
title('OHT 3hr D4C Velocity Distribution')
set(gca,'FontSize',14)
%% plot all 3D tracks
```

```
numTracks = length(CellTracks);
colorsidx = randi(256,numTracks);
cmap = jet(256);
figure
for i = 1:numTracks
X = CellTracks(i).XYZ(:,1);
Y = CellTracks(i).XYZ(:,2);
Z = CellTracks(i).XYZ(:,3);
hold on
plot3(X,Y,Z,'-o','MarkerFaceColor',cmap(colorsidx(i),:),'MarkerSize',10)
end
hold off
%% plot the boundary of the spheroid
k = boundary(Data(:,1),Data(:,2),Data(:,3));
figure
hold on;
%plot3(Data(:,1),Data(:,2),Data(:,3));
trisurf(k,Data(:,1),Data(:,2),Data(:,3),'Facecolor','red','FaceAlpha',0.01,'EdgeColor',[0.9 0.9
0.9],'LineWidth',0.25,'LineStyle','--')
%% plot 1 track with spheroid surface as reference
k = boundary(Data(:,1),Data(:,2),Data(:,3));
cnum = 1; %20,
X = CellTracks(cnum).XYZ(:,1);
Y = CellTracks(cnum).XYZ(:,2);
Z = CellTracks(cnum).XYZ(:,3);
figure, plot3(X,Y,Z,'-o','Color','b','MarkerSize',10,'MarkerFaceColor','#D9FFFF')
hold on
%plot3(MeanCentroid(1,1),MeanCentroid(1,2),MeanCentroid(1,3),'-o','MarkerFaceColor','r','MarkerSize',10) %plot
spheroid centtroid as well
trisurf(k,Data(:,1),Data(:,2),Data(:,3),'Facecolor','red','FaceAlpha',0.01,'EdgeColor',[0.9 0.9
0.9],'LineWidth',0.25,'LineStyle','--')
%% check the transformation worked, check angular displacement of example cell
Datat = [(Data(:,1)-MeanCentroid(:,1)),(Data(:,2)-MeanCentroid(:,2)),(Data(:,3)-MeanCentroid(:,3))];
k = boundary(Datat(:,1),Datat(:,2),Datat(:,3));
cnum = 693; %20, %55 vs 60
X = CellTracks(cnum).XYZt(:,1);
Y = CellTracks(cnum).XYZt(:,2);
Z = CellTracks(cnum).XYZt(:,3);
figure, plot3(X,Y,Z,'-o','Color','b','MarkerSize',10,'MarkerFaceColor','#D9FFFF')
hold on
plot3(0,0,0,'-o','MarkerFaceColor','r','MarkerSize',10) %plot spheroid centtroid as well
trisurf(k,Datat(:,1),Datat(:,2),Datat(:,3),'Facecolor','red','FaceAlpha',0.01,'EdgeColor',[0.9 0.9
0.9],'LineWidth',0.25,'LineStyle','--')
%% get cells on edge of spheroid say top 75% percentile, then from those rank for max ang displacement top 90%
percentile
figure,histogram([CellTracks.MeanDistFromCentroid])
%top 10% = 90th percentile
distdata = [CellTracks.MeanDistFromCentroid]';
P = prctile(distdata,75);
[ridx,~] = find(distdata>=P);
angdata = [CellTracks.TotAngDisp]';
P2 = prctile(angdata,75);
[ridx2,~] = find(angdata>=P2);
%find values common to both datasets - intersect
ridxu = intersect(ridx,ridx2);
k = boundary(Data(:,1),Data(:,2),Data(:,3));
```

```
colorsidx = randi(256,length(ridxu));
cmap = jet(256);
figure,hold on
plot3(MeanCentroid(1,1),MeanCentroid(1,2),MeanCentroid(1,3),'-o','MarkerFaceColor','r','MarkerSize',10) %plot
spheroid centtroid as well
trisurf(k,Data(:,1),Data(:,2),Data(:,3),'Facecolor','red','FaceAlpha',0.01,'EdgeColor',[0.9 0.9
0.9],'LineWidth',0.25,'LineStyle','--')
nplot = length(ridxu);
clrs = distinguishable_colors(nplot);
for i = 1:nplot
cnum = ridxu(i);
X = CellTracks(cnum).XYZ(:,1);
Y = CellTracks(cnum).XYZ(:,2);
Z = CellTracks(cnum).XYZ(:,3);
plot3(X,Y,Z,'-o','Color','b','MarkerSize',10,'MarkerFaceColor',clrs(i,:))
end
hold off
%% plot velocities of polar coordinates
% what is a reasonable mean dist from centroid cutoff? plot all cell's mean dist to centroid
%figure
%histogram([CellTracks.MeanDistFromCentroid])
%do 60 or less vs > 80
distdata = [CellTracks.MeanDistFromCentroid]';
[ridx1,~] = find(distdata < 50);
[ridx2,~] = find(distdata > 90);
NumCells = 40;
ridx1pts = randperm(length(ridx1),NumCells);
ridx2pts = randperm(length(ridx2),NumCells);
clrs1 = distinguishable_colors(NumCells);
clrs2 = distinguishable_colors(NumCells);
figure
for i = 1:NumCells
subplot(2,3,1)
data1 = [CellTracks(ridx1(ridx1pts(i))).Theta_Vel(:,1)];
hold on
plot([CellTracks(ridx1(ridx1pts(i))).Time(2:end,1)],data1,'Color',clrs1(i,:));
title('theta displacement') %theta,rho,z
ylim([-50 50])
hold off
subplot(2,3,2)
data2 = [CellTracks(ridx1(ridx1pts(i))).Rho_Vel(:,1)];
hold on
plot([CellTracks(ridx1(ridx1pts(i))).Time(2:end,1)],data2,'Color',clrs1(i,:));
ylim([-40 40])
title('rho displacement') %theta,rho,z
hold off
subplot(2,3,3)
data3 = [CellTracks(ridx1(ridx1pts(i))).Z_Vel(:,1)];
hold on
plot([CellTracks(ridx1(ridx1pts(i))).Time(2:end,1)],data3,'Color',clrs1(i,:));
ylim([-40 40])
title('z displacement') %theta,rho,z
hold off
subplot(2,3,4)
data4 = [CellTracks(ridx2(ridx2pts(i))).Theta_Vel(:,1)];
hold on
```

```
plot([CellTracks(ridx2(ridx2pts(i))).Time(2:end,1)],data4,'Color',clrs2(i,:));
ylim([-50 50])
title('theta displacement') %theta,rho,z
hold off
subplot(2,3,5)
data5 = [CellTracks(ridx2(ridx2pts(i))).Rho_Vel(:,1)];
hold on
plot([CellTracks(ridx2(ridx2pts(i))).Time(2:end,1)],data5,'Color',clrs2(i,:));
ylim([-40 40])
title('rho displacement') %theta,rho,z
hold off
subplot(2,3,6)
data6 = [CellTracks(ridx2(ridx2pts(i))).Z_Vel(:,1)];
hold on
plot([CellTracks(ridx2(ridx2pts(i))).Time(2:end,1)],data6,'Color',clrs2(i,:));
ylim([-40 40])
title('z displacement') %theta,rho,z
hold off
end
%% plot mean velocity vs average dist from centroid
% recalculate the mean spheroid position for each timepoint :
% try plotting individual time points not averages
MeanV = [];
MeanD = [];
for i = 1:length(CellTracks)
MeanD = [MeanD;CellTracks(i).MeanDistFromCentroid];
MeanV = [MeanV;mean([CellTracks(i).Velocity(:)])];
end
figure
scatter(MeanD,MeanV)
%% plot radial disp vs ang disp at diff times for cells in a spheroid (color by average radial dist from spheroid)
distdata = [CellTracks.MeanDistFromCentroid]';
P = prctile(distdata,75);
[ridx,~] = find(distdata>=P);
angdata = [CellTracks.TotAngDisp]';
P2 = prctile(angdata,75);
[ridx2,~] = find(angdata>=P2);
%find values common to both datasets - intersect
ridxu = intersect(ridx,ridx2);
colorsidx = randi(256,length(ridxu));
cmap = jet(256);
figure

nplot = length(ridxu);
clrs = distinguishable_colors(nplot);
for i = 1:nplot
cnum = ridxu(i);
subplot(3,1,1)
data1 = rad2deg([CellTracks(cnum).polar(:,1)]);
hold on
plot(1:length(data1),data1,'Color',clrs(i,:));
title('theta displacement') %theta,rho,z
hold off
subplot(3,1,2)
data2 = [CellTracks(cnum).polar(:,2)];
hold on
```

```
plot(1:length(data2),data2,'Color',clrs(i,:));
title('rho displacement') %theta,rho,z
hold off
subplot(3,1,3)
data3 = [CellTracks(cnum).polar(:,3)];
hold on
plot(1:length(data3),data3,'Color',clrs(i,:));
title('z displacement') %theta,rho,z
hold off
end
%% MSD analysis for cells to compute velocity autocorrelations
SPACE_UNITS = 'µm';
TIME_UNITS = 's';
NumTracks = size(CellTracks,2);
tracks2 = cell(NumTracks,1);
for i = 1:NumTracks
    % convert frame number to time in seconds
    tvec = CellTracks(i).Time.*(30*60); %30 min per frame, 60 sec per minute
    XYZvec = CellTracks(i).XYZ;
    tracks2{i} = [tvec,XYZvec];
end
ma = msdanalyzer(3, SPACE_UNITS, TIME_UNITS);
ma = ma.addAll(tracks2);
%ma2 = msdanalyzer(3, SPACE_UNITS, TIME_UNITS);
%ma2 = ma.addAll(tracks2);
figure
ma.plotTracks
ma.labelPlotTracks
%%
v = ma.getVelocities
V = vertcat( v{:} );
figure
hist(V(:, 2:end), 50) % we don't want to include the time in the histogram
box off
xlabel([ 'Velocity (' SPACE_UNITS '/' TIME_UNITS ')' ])
ylabel('#')
%%
ma = ma.computeVCorr;
%ma2 = ma2.computeVCorr;
figure
ha = gca;
ma.plotMeanVCorr(ha,0,1:3:10)
%hold on
%ma2.plotMeanVCorr(ha,0,2:2:10)
%hold off
% 'XTick' Values
%xlabel('Delay (hours)')
%xt = get(gca, 'XTick');
%set(gca, 'XTick', xt, 'XTickLabel', round(xt/3600))
h = get(gca, 'Children');
set(h(1), 'Color', 'r');
%%
distdata = [CellTracks.MeanDistFromCentroid]';
P = prctile(distdata,75);
[ridx,~] = find(distdata>=P);
angdata = [CellTracks.TotAngDisp]';
```

```
P2 = prctile(angdata,75);
[ridx2,~] = find(angdata>=P2);
%find values common to both datasets - intersect
ridxu = intersect(ridx,ridx2);
figure
ha = gca;
ma.plotMeanVCorr(ha, 0,ridxu) %idxs
hold on
ma.plotMeanVCorr(ha, 0) %idxs
h = get(gca, 'Children');
set(h(1), 'Color', 'k');
set(h(2), 'Color', 'r');
xlim([0 10*3600])
xlabel('Delay (hours)')
xt = get(gca, 'XTick');                    % 'XTick' Values
set(gca, 'XTick', xt, 'XTickLabel', round(xt/3600))
%% plot autocorrelations for all cells, and then by radius
%sort cells by their total autocorrelation over time
%%
  y=randn(30,80); % 30 by 80 , 30 observations
  x=1:size(y,2); % 1 by 80 time vec
figure,shadedErrorBar(x,y,{@median,@std},'lineProps',{'r-o','markerfacecolor','r'});
%%
NPts = 10; %size(ma.vcorr,1);
NtimePts = size(ma.vcorr{1, 1},1);
clrs = distinguishable_colors(NPts);
Y_all = NaN(NPts,NtimePts);
figure
for i = 1:NPts %size(ma.vcorr,1)
X = ma.vcorr{i, 1}(:,1)./60; %convert seconds to minutes
Y = ma.vcorr{i, 1}(:,2);
hold on
plot(X,Y,'LineWidth', 2,'Color',clrs(i,:))
xlabel('Delay(min)')
ylabel('Normalized Velocity Autocorrelation')
yline(0,'--','LineWidth',2,'Color',[0.5 0.5 0.5])
   for j = 1:size(Y,1)
   tidx =  CellTracks

   Y_all(j,:) = Y';
   end
end
shadedErrorBar(X', Y_all, {@mean,@std},'lineprops',{'Color','k','LineWidth',3,'LineStyle','-'}) %[map(1,:)] for
color
hold off
%%
%% also compute spatial autocorrelations
% from a distance of seperation of 1 cell radii to 2 speroid radii , r --> R
% for each distance, compute unique pairs of cells for each seperation distance
% to get the spatial autocorrelation get the normalized vector then dot product
p1 = [1,4;2,5];
dx_1 = p1(2,1)-p1(1,1);
dy_1 = p1(2,2)-p1(1,2);
d1 = [dx_1,dy_1];
%normalize the velocity vector
d1norm = d1./sqrt(dx_1^2+dy_1^2);
```

```
p2 = [1,1;4,1];
dx_2 = p2(2,1)-p2(1,1);
dy_2 = p2(2,2)-p2(1,2);
d2 = [dx_2,dy_2];
d2norm = d2./sqrt(dx_2^2+dy_2^2);
corr = dot(d1norm,d2norm);
%% compute velocity cross-correlations
% for each value of radius we choose to calculate pairs of spatial autocorrelations
% for each velocity timepoint, for each seperation radius we  will calculate the pairs of cells that are within
int = 5;
SDist = 0:int:200; % in microns
SC = NaN(37,size(SDist,2)); %then average over the columns
SCcell = cell(37,size(SDist,2));
SCv = NaN(1,size(SDist,2));
radc = 25; %50 microns
for i = 1:37 % number of velocity time pts = (tot frames - 1)
for j = 1:size(SDist,2) %num of pts in resolution of spatial velocity correlations
   % we need to find all centroid pairs that are within SD+/- 5um of eachother so if SD is 20, 15 to 25 um would
   % for each pair of timepoints, find cells that have a pair of centroids for those times e.g. frame 3 and 4
   % make a list of cells that has centroids for time 1 and 2
      % get a list of cells that have indices at t == 1
      f1 = i; f2 = i+1;
      %make list of cells that have centroids at both f1 and f2
      clist = [];
      for ii = 1:length(CellTracks)
      if ismember(f1,CellTracks(ii).Time) & ismember(f2,CellTracks(ii).Time)
         if CellTracks(ii).MeanDistFromCentroid >= radc
         clist = [clist;ii];
         end
      end
      end
      % generate matrix :
      % ave pos, del_pos,
      %a
      mat = NaN(size(clist,1),6);
      for m = 1:size(clist,1)
      idx1 = find([CellTracks(clist(m)).Time] == f1);
      idx2 = find([CellTracks(clist(m)).Time] == f2);
      pos1 = CellTracks(clist(m)).XYZ(idx1,:);
      pos2 = CellTracks(clist(m)).XYZ(idx2,:);
      mat(m,1:3) = [(pos1(1)+pos2(1))/2,(pos1(2)+pos2(2))/2,(pos1(3)+pos2(3))/2];
      mat(m,4:6) = [(pos2(1)-pos1(1)),(pos2(2)-pos1(2)),(pos2(3)-pos1(3))];
      end
      % now we can calculate the distance for each cell and make it a list,
      %remember all these indices are going to reference the row index of matrix 'mat'
      pairlist = [];
         %calculate distances between each pair of points, start with
         for m1 = 1:size(clist,1)
            dist = zeros(size(clist,1),1);
            for m2 = 1:size(clist,1)
            dist(m2,1) = sqrt(((mat(m2,1)-mat(m1,1))^2)+ ((mat(m2,2)-mat(m1,2))^2)+((mat(m2,3)-mat(m1,3))^2));
            end
            %for m1, find all pairs by subtracting SDist incriment j
            idxs1 = find(dist > SDist(j)-int/2);
            idxs2 = find(dist < SDist(j)+int/2);
            idxs = intersect(idxs1,idxs2);
```

```
            for x = 1:length(idxs)
            pairlist =  [pairlist;m1,idxs(x)];
            end
          end
      % now we can pair down this list
          [idx,idx]=unique(sort(pairlist')','rows','stable');
          B=pairlist(idx,:); %only unique pairs are here
          %now we can assess the spatial velocity correlation for each pair of points aka each row in mat
          vcorrs = NaN(size(B,1),1);
          for jj = 1:size(B,1)
          vec1 = [mat(B(jj,1),4),mat(B(jj,1),5),mat(B(jj,1),6)];
          vec2 = [mat(B(jj,2),4),mat(B(jj,2),5),mat(B(jj,2),6)];
          vec1N = vec1./sqrt((vec1(1)^2)+(vec1(2)^2)+(vec1(3)^2));
          vec2N = vec2./sqrt((vec2(1)^2)+(vec2(2)^2)+(vec2(3)^2));
          vcorrs(jj,1) = dot(vec1N,vec2N);
          end
          SCcell{i,j} = vcorrs;
          SC(i,j) = mean(vcorrs);
          %B are the unique pairs that we should sample for the
      % for each entry, calculate pairs that satify criteria
   % calculate the x,y,z displacements, along with the average position ()
      % for each cell, search the entire list for cells within 15 to 25um of the current cell, and add it to a list pairs
with accumulates over every loop iteration
end
   % make a list of the cell centroids for each time point, put original index in first column
   % for these centroids, get a list of pairs that are within SD+/-5um of eachother
end
for xx = 1:size(SC,2)
SCv(1,xx) = mean(SC(:,xx));
end
%%
figure,shadedErrorBar(SDist,SC,{@mean,@std},'lineProps',{'r-o','markerfacecolor','r'});
hold on
plot(SDist,exp(-SDist./10),'LineWidth',2)
yline(0,'--','LineWidth',2,'Color',[0.5 0.5 0.5])
hold off
%% error fcn to determine length correlation,
% for 10A = 9 microns
% for OHT = 3 microns
% plot an error function
error = [];
lcorr = 20; %max value
for lc = 1:lcorr
pts = exp(-SDist./lc);
ee = (pts - SCv).^2;
eesum = sum(ee);
error = [error;eesum];
end
figure,plot(1:lcorr,error)
%%
A = [1 2 3; 2 4 6; 3 1 2; 1 1 1]
[idx,idx]=unique(sort(A')','rows','stable');
B=A(idx,:) %only unique pairs are here
%then for each pair, calculate a velocity correlation, average the correlations for a given time point and seperation
dist, then average all times to get 1 point for each sep dist
%% calculate path straightness for 3D trajectories
```

```
%% seperate trajectories that have a time gap into seperate segments without the gap, get rid of any traj 1-4 in
length?
%%
%% calculate distance of point to triangular mesh that defines the spheroid exterior
[distances,surface_points] =
point2trimesh('Faces',k,'Vertices',Data(:,1:3),'QueryPoints',MeanCentroid,'UseSubSurface',false);
%patch(FV,'FaceAlpha',.5); xlabel('x'); ylabel('y'); zlabel('z'); axis equal; hold on
figure
plot3M = @(XYZ,varargin) plot3(XYZ(:,1),XYZ(:,2),XYZ(:,3),varargin{:});
plot3M(points,'*r')
plot3M(surface_points,'*k')
plot3M(reshape([shiftdim(points,-1);shiftdim(surface_points,-1);shiftdim(points,-1)*NaN],[],3),'k')
%%
Faces = k;
Vertices = Data(:,1:3);
points = [MeanCentroid(1,1)+15,MeanCentroid(1,2)+15,MeanCentroid(1,3)];
%points = MeanCentroid;
% Find vertices without face
NVert = size(Vertices,1);
has_faces = ismember(1:NVert, Faces);
% Remove vertices without face
FV2.vertices = Vertices(has_faces,:);
new_ind = cumsum(has_faces);
FV2.faces = Faces;
for iVert = 1:NVert
FV2.faces(FV2.faces==iVert) = new_ind(iVert);
end
[distances,surface_points] = point2trimesh(FV2, 'QueryPoints', points);
figure, hold on
patch(FV2,'FaceAlpha',.5); xlabel('x'); ylabel('y'); zlabel('z'); axis equal; hold on
plot3M = @(XYZ,varargin) plot3(XYZ(:,1),XYZ(:,2),XYZ(:,3),varargin{:});
plot3M(points,'*r')
plot3M(surface_points,'*k')
plot3M(reshape([shiftdim(points,-1);shiftdim(surface_points,-1);shiftdim(points,-1)*NaN],[],3),'k')
%%
figure
trisurf(Faces,FV2.vertices(:,1),FV2.vertices(:,2),FV2.vertices(:,3))
```