

# Spatial Transcriptomics: Mapping Visium and Xenium Data Automated Image Registration and Application of External Cell Segmentation Models

Claire Fucetola

Adviser: Ben Raphael

## Abstract

*Traditional spatial transcriptomics technologies providing whole transcriptome information, such as Visium, are hindered by low resolution and sparse data. Newer technology, such as Xenium, incorporates high resolution, sub cellular data for a select number of genes, introducing a solution to fill in the gaps produced by sparse data by mapping between the two datasets. Although current methodologies rely on manual feature selection on the provided H&E (Hematoxylin and Eosin) images to create the mapping between the technologies (by registering the images), this study presents an automated solution by utilizing intensity-based image registration and maximizing the mutual information between the two images. The intensity-based image registration method resulted in increased performance on image alignment. Additionally, the external cell segmentation model cellcutter was explored to create more accurate cell boundaries and aid in the process of binning transcripts to cells. Applying cellcutter to the Xenium dataset more accurately distinguished cell space from cell-free space, allowing the cell segmentation mask to recognize and separate the background of the image.*

## 1. Introduction

Spatial Transcriptomics (ST) is a groundbreaking molecular technology that allows scientists to spatially orient cells and cell-type compositions by measuring gene activity and expression across a tissue sample [6]. Furthermore, it is a technology used to derive an understanding of cell-type composition of tissues and the spatial organization of single-cell RNA sequences (scRNA-seq) across the tissue [20]. ST provides insight into three main biological domains: what types of cells

make up a tissue, the similarity and dissimilarity of the cells among one another, and understanding how cells interact and communicate with one another within a tissue [20]. This allows scientists to gain an understanding of tissue activity, gain insight into tissue structures and tumors, and define cell and tissue function [21]. These advancements greatly benefits the biological community and help provide solutions to complex problems within the medical community [21].

10x Genomics, a biotechnology company, has developed two technologies for extracting gene expression across a tissue sample while preserving spatial context: Visium and Xenium. The Visium technology uses spatially indexed pixels generated by a spot-based approach where spots, 50-200 microns in size, provide unique barcodes to describe known locations on the tissue sample [20]. From each spot, the Visium technology extracts mRNA sequences providing whole transcriptome information, high dimension gene expression data [17]. While Visium produces data for tens of thousands of different genes, this technology requires destruction of the tissue used and often results in sparse data [17]. Xenium, on the other hand, is a cutting edge technology released to provide high resolution cellular images with spatial profiling of genes and proteins [7]. Unlike Visium, Xenium does not require a spot based approach nor does it require destruction of the tissue sample. Xenium produces subcellular locations of transcripts across a tissue slice [17]. However, Xenium is low dimensional in that it only provides information for an upper bound of 500 genes, including 100 custom add-on RNA targets [17]. Additionally 10x Genomics performs its own “in house” cell segmentation to Xenium providing estimates of cell boundaries, assignment of transcripts to cells, and identification of cell types based on cluster assignment reflecting groups of cells with similar expression profiles [3].

The development of the Xenium technology provides a gateway for research focusing on mapping technologies and techniques for spatial representation to produce more robust, less sparse gene expression. Since the Xenium technology does not destroy the tissue sample it is analyzing, a tissue sample can first be analyzed by Xenium to produce high resolution, subcellular, information and then passed to Visium to extract whole transcriptome, high dimensional information [17]. By producing a mapping between the two datasets, the Xenium information allows for scientists to

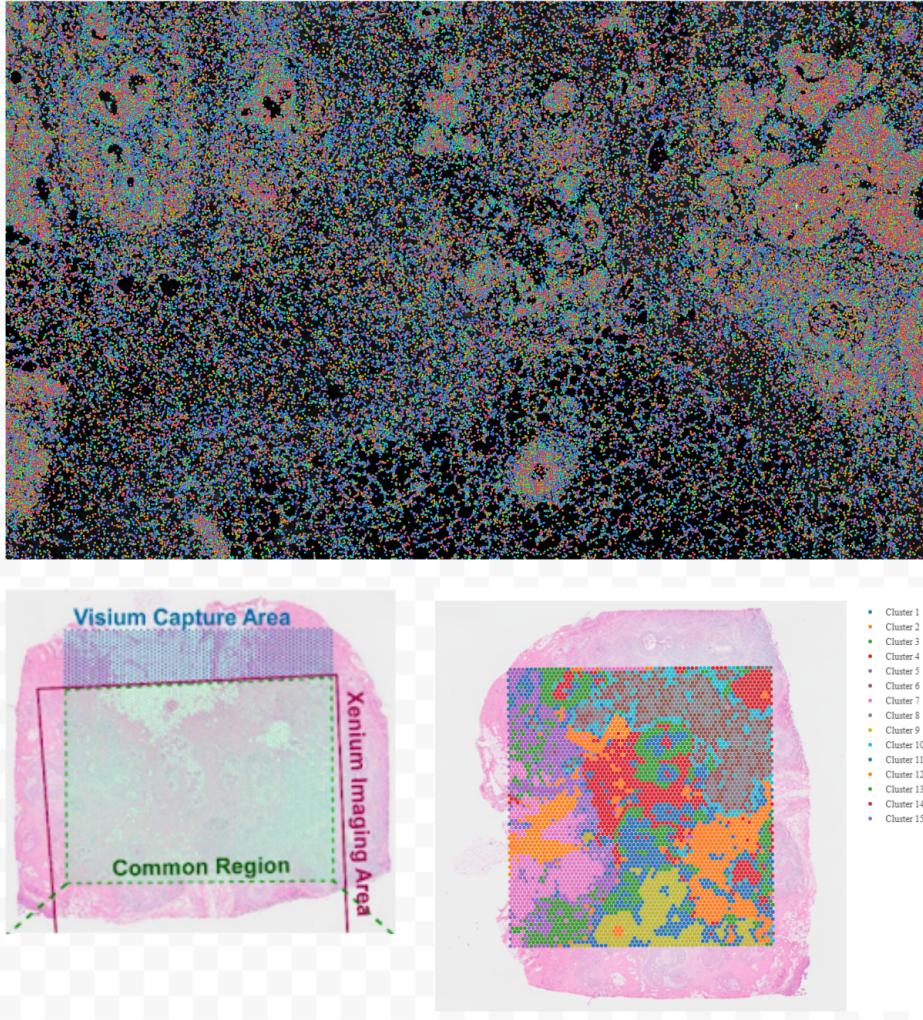
leverage the high resolution aspects of the Xenium expression to assist in filling in the sparse data provided by Visium. Both Xenium and Visium produce H&E (Hematoxylin and Eosin) images of the tissue slice, and one way a mapping can be acquired is through image registration, the process of creating one coordinate system describing two images by transforming one image to spatially align with another [10]. However, this process is a complex problem due to the large size of the images and varying resolution between images, leaving current registration to rely on manual techniques [17].

This project has two main objectives, one focusing on image registration and mapping and another focusing on cell segmentation. First, it aims to develop a more efficient and automated process of aligning the images provided by Xenium and Visium. Second, it will explore predefined cell segmentation models to implement on the Xenium Dataset with the purpose of improving the current cell segmentation from 10x Genomics and defining more robust cell boundaries in the Xenium images.

## 2. Background and Related Work

### 2.1. Image Registration

The image registration for creating a mapping between Xenium and Visium with the human breast tumor FFPE dataset is currently performed through a process of keypoint identification and RANSAC to produce a 3x3 Homography Matrix [17] (Figure 1). The keypoints here are defined via manually registering the keypoints between the two H&E images, where over 100 landmark features were identified [17]. This process is both inefficient and error prone. The Xenium image is 35416 x 25778 and the Visium Image is 21571 x 19505, therefore it requires someone to look through these large images, find features corresponding to the two images and accurately mark them in each image. Additionally, since the goal of the image registration process is to utilize the subcellular qualities of the Xenium image, it is critical that the defined features are highly accurate as discrepancies could lead to misalignment at a cellular level. Therefore, this project looks towards automated methods to conduct registration.

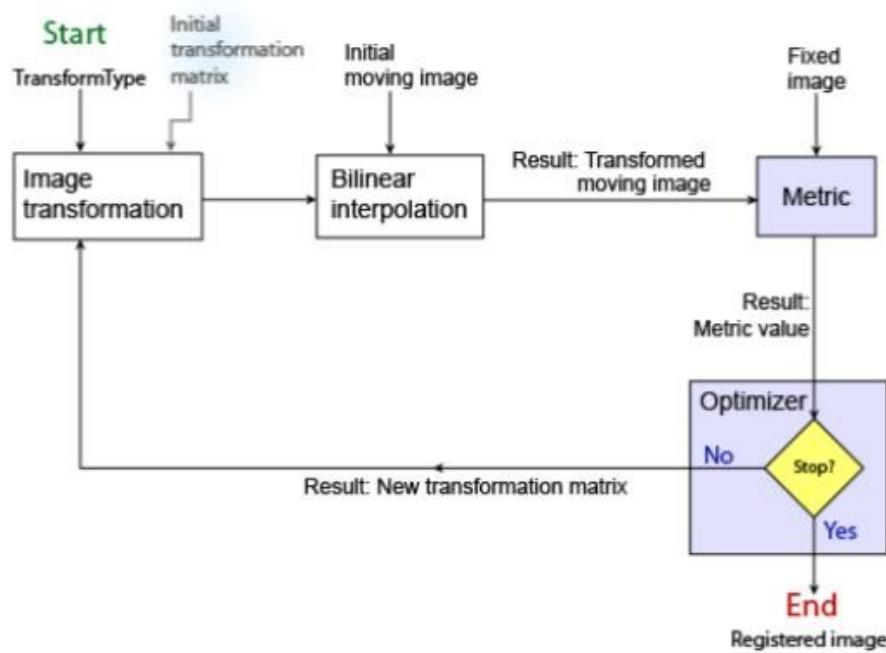


**Figure 1: The current registration of the Xenium image to the Visium Image (bottom-left). Xenium detected 313 Target Genes (top) whereas Visium detected 18,056 Genes over 4992 Spots (bottom-right)[17, 3]. Registration Diagram from [17]**

An automated image registration task has previously been dominated at an algorithmic level using methods such as SIFT (Scale Invariant Feature Transformation) [16]. However, with the emergence of deep learning this task has moved towards both supervised and unsupervised learning methods [16]. For the purposes of this project, unsupervised learning methods will be the focus as the H&E images from both Xenium and Visium are unlabeled. Unlike standard image registration tasks which have focused on producing homography matrices, many learning-based registration methods produce a deformation field.

Intensity-Based Image Registration (IBIR) is an iterative process of optimizing a specified transformation matrix over a similarity metric to produce an optimal transformation matrix to

register the provided moving image to the provided fixed image [2]. Rather than relying on feature extraction, this registers images through looking directly at intensity values and patterns to create a comparison. IBIR is commonly used across medical images such as MRI and CT scans to assist in registering multi-modal images [23]. This method aims at maximizing the similarity between the two images by adjusting the transformation matrix [2]. MATLAB currently has an implementation of IBIR with an architecture defined in Figure 2.



**Figure 2: Architecture of MATLAB’s Intensity Based Image Registration with user defined metric and optimizer. Diagram from [2]**

## 2.2. Mutual Information

Mutual Information is defined as the amount of information one random variable gives about another random variable, it measures the statistical dependence between two random variables [11]. Mutual Information is a widely used metric when working with multi-modal image registration, where images of the same tissue or area are taken using different technologies such as a CT scan and MRI [22]. However, it is also a useful metric for mono-modal images, images captured through the same

modality, when intensity values between the images differ due to lighting or contrast variation. The Mutual Information between two images A and B is formally defined as:

$$I(A, B) = \sum_{a,b} p(a,b) \log \frac{p(a,b)}{p(a)p(b)} [22] \quad (1)$$

The probabilities can be calculated for the images by computing a histogram of intensities over a bin size [2, 16]. Since we are treating images A and B as independent, due to either the degree of intensity difference or a multi-modal input, the images are aligned when the mutual information is maximized [16].

### 2.3. 1+1 Evolutionary

1+1 Evolutionary is a subset of the Evolutionary Algorithms which are a class of search algorithms used for optimization [13]. Evolutionary Algorithms is a repeated, iterative process of selection, mutation, and replacement to provide a solution(s) to an optimization task [13]. 1+1 Evolutionary is the most simple variant of this class and one of the few Evolutionary Algorithms to be behaviorally analyzed [13]. Furthermore, it has desirable properties in that it is efficient for many problems and cannot get stuck at local optima [8]. The Algorithm is formalized as the following (Figure 3):

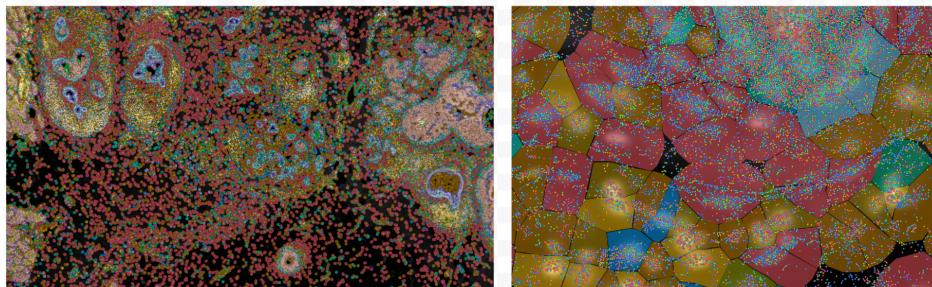
- 
- 1: Initialize  $\mathbf{x} \in \{0, 1\}^l$  uniformly at random.
  - 2: Create  $\mathbf{x}'$  by flipping one each bit in  $\mathbf{x}$  with probability  $p_m$ .
  - 3: Select if  $f(\mathbf{x}') \geq f(\mathbf{x})$  then  $\mathbf{x} := \mathbf{x}'$ .
  - 4: Go to 2 until a termination condition is fulfilled.
- 

**Figure 3: 1+1 Evolutionary Algorithm as defined in [8]**

This algorithm mutates the parameters from the previous iteration (parent) and observes how it affects the new parameters (child) [18]. If the new parameters are more optimal, with respect to the previous iteration, then the child becomes the new parent [18]. This process is repeated until the stopping condition is met. The stopping condition used in IBIR when MATLAB sets this optimizer is Maximum Iterations [18]. In the case for image registration,  $x$  is the transformation matrix and  $f(x)$  is the mutual information metric.

## 2.4. Cell Segmentation

10x Genomics defines cell boundaries across the Xenium image by performing nucleus segmentation across a DAPI (4',6-diamidino-2-phenylindole) image using a pre-trained Cellpose Model and then estimating cell boundaries relative to each nucleus [5] (Figure 4). However, to estimate the cell boundaries, cell boundaries are expanded out from the nucleus in each direction until it either reaches another, previously allocated, cell boundary or it has expanded 15 microns [5, 17]. While the nucleus representations are accurate, the current process of defining cell boundaries leads to a misrepresentation on a couple of fronts. First, cell boundaries are determined on a “first-come first-serve” basis where based on the order in which a nucleus is explored to define the cell boundary, one cell might be larger while a neighboring cell might be smaller when in reality it is reversed or vice versa. Furthermore, this allows for transcripts to be assigned to a cell in which they do not belong. Second, cells may be defined in “cell-free space” leading to a misrepresentation of the tissue sample itself.



**Figure 4: 10X Genomics Defined cell boundaries on whole sample (left) and a zoomed in version to more closely see the cell boundaries and transcript location within the cells (right). 10x Genomics detects 167,780 cells.**

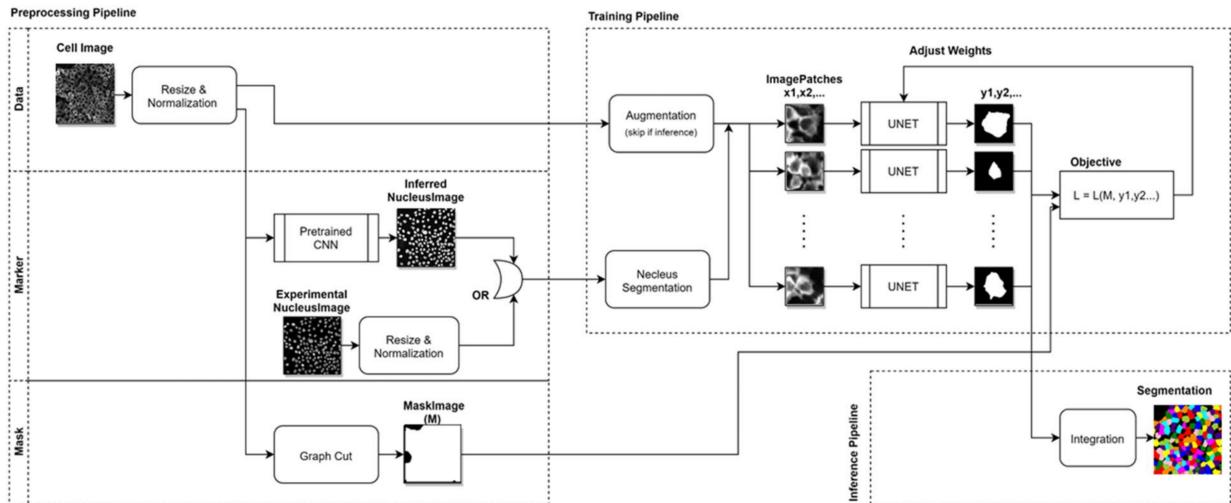
## 2.5. Cellpose

Cellpose is a cell segmentation model aimed towards developing a trained model to precisely identify cell boundaries and be generalizable across a variety of datasets without further need of post-training [19]. The underlying architecture is a modified UNet with encoder-decoder structures and skip connections [19]. Most notably, the training set consists of a large variety of labeled

data, primarily consisting of fluorescent, membrane, and bright field images and including smaller subsets of other microscopy images and non microscopy images resulting in high precision on both generalized and specialized datasets [19]. Additionally, cellpose was separately trained for nuclear segmentation [19]. Both methodologies resulted in two notable pretrained networks from cellpose which are generalizable across novel datasets, one for cytoplasmic cellular segmentation and the other for nuclear segmentation.

## 2.6. Cellcutter

Cell segmentation as a deep learning task has previously been reliant on supervised learning methods; however, cellcutter takes a step forward by creating an alternative strategy of self-supervised learning for single-cell segmentation. Cellcutter proposed a marker-controlled method to approximate locations of cells and avoid over segmentation with noise [12]. Additionally, this process was broken into pipelines for pre-processing, to identify approximate areas of where cells can be located based on cell nuclei marker locations and graph masks, and training, a collection of UNet encoder-decoder structures [12]. Cellcutter uniquely tackles the segmentation problem by breaking the input image down into single-cell chunks for training and analysis, aided by the marker location [12]. An overview of the architecture is shown in Figure 2.6.



**Figure 5: The Cellcutter algorithm architecture as displayed with the Pre-processing (left) and Training (right) pipelines. Diagram from [12]**

This showed high performance for high density cell areas with both fluorescence and bright-field microscopy images without the need for human labeled data [12]. Unlike using a pre-trained model this allows for the flexibility of unique datasets and further tailoring the model to the current data.

### **3. Approach**

This project was broken into two distinct parts, one focusing on Image Registration and another focusing on Cell Segmentation. Robust cell segmentation increases the accuracy at which Xenium cells are binned to Visium spots after image registration.

#### **3.1. Data**

The datasets that will be used to test these methods will be on the human breast cancer FFPE section, specifically In Situ Sample 1 and Visium Spatial.<sup>1</sup> Both image registration and cell segmentation has been previously performed on this dataset as a baseline in accordance with this task. The data needed can be found through both the input files (for the registration task) and the Xenium output files (for the cell segmentation task).

#### **3.2. Image Registration**

The Image Registration task was performed on the Xenium and Visium H&E images. For the Xenium image, the OME-TIFF image was downloaded and for the Visium image tissue image TIFF file was downloaded. Both files were read using opencv's `imread` function.

1. Feature Based Image Registration: SIFT was used to evaluate the performance of traditional algorithmic methods for automated image registration. Additionally, this was used to provide a baseline for the image registration task and help gain insight into the dataset. A feature based registration was not previously used for this task, although it was present in the registration of IF (immunofluorescence) images to the Xenium morphology images.
2. Intensity Based Image Registration: MATLAB's `imregconfig`, configured for multi-

---

<sup>1</sup><https://www.10xgenomics.com/products/xenium-in-situ/preview-dataset-human-breast>

modal, was used to find the optimized 3x3 Affine Transformation Matrix corresponding to the set of images. This sets the metric to be mattes mutual information, which is a variation of the mutual information metric described above, and the optimizer to be 1+1 Evolutionary.

### 3.3. Cell Segmentation

The Cell Segmentation task was performed using the Xenium output bundle which held information on cell segmentation masks, nucleus segmentation masks, nucleus centroids, and morphology images (DAPI). All files used were of the same resolution as the Xenium H&E OME-TIFF file.

CellCutter was determined to be the method and model used for this task due to two main features: masking and self supervision. Cellcutter's masking feature helps to pre-allocate space in the image that is “cell space”, the space where cells are located, and “cell-free space”, the space where there are no cells. This helps to establish a descriptor of where cells can and cannot be while training. Cellcutter is also self supervised which is novel, given that most segmentation methods require labeled data, and essential as the images in this dataset do not have cell boundary labels and it would be infeasible to add them for the sake of this project.

## 4. Implementation

All code was completed on Google Colab and MATLAB.<sup>2</sup>

### 4.1. Image Registration: SIFT

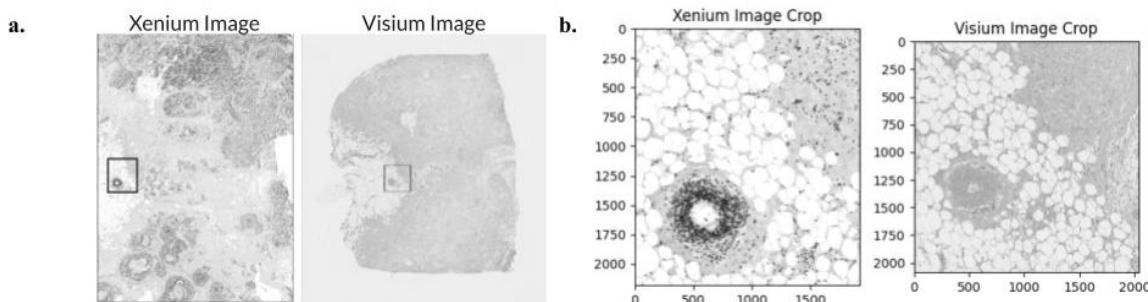
SIFT was performed using the following opencv functions to locate features and descriptors: `SIFT_create` and `sift.detectAndCompute`. An opencv `BFMatcher` object was used to match the descriptors with one another. RANSAC was completed to find the best fit transformation matrix given the matched descriptors using opencv's `findHomography` function. This was performed on the Xenium and Visium H&E images and first evaluated visually to see if the images adequately align.

---

<sup>2</sup>[https://github.com/clairaf/Fall\\_IW](https://github.com/clairaf/Fall_IW)

## 4.2. Image Registration: Data and Preprocessing

The Xenium Image used was a OME-TIFF file, which supplied a higher resolution image. To prepare the data for the MATLAB pipeline, the Xenium image first had to be rotated 90 degrees using `opencv rotate` function to position the image in the same orientation as the Visium image. Two corresponding crops were selected from each using an interactive tool in google colab to visualize the selection and help retrieve similar areas (Figure 6a). Crops were taken to send smaller, more computationally manageable images to the MATLAB pipeline. The Xenium Crop was then downsized by 50 percent using `opencv resize` function (Figure 6b) and both the Visium and Xenium crops were sent to the MATLAB Script. The Xenium Crop had to be downsized to retrieve a more comparable resolution to the Visium Crop to provide for an adequate amount of initial valuable mutual information between the two crops.



**Figure 6: Xenium and Visium Image with visualized section select (a) and the selected Xenium (downsized 50%) and Visium Crops (b).**

## 4.3. Image Registration: MATLAB Pipeline and Application

The MATLAB script consisted of a file that took in the saved Xenium Crop as the moving image and the saved Visium Crop as the fixed image. It used the following built-in MATLAB functions: `imregconfig` and `imregtform`. `imregconfig` was used to specify the configuration for the image registration, for the purposes of this study it was set to “multi-modal”. Setting the configuration to “multi-modal” within `imregconfig` automatically establishes the metric as Mattes Mutual Information and the optimizer as 1+1 Evolutionary. The default values within the metric are 50

Histogram bins and 500 Spatial Samples [2]. The optimizer defaults to 3 pyramid levels with 100 Maximum Iterations per pyramid [2]. The number of Maximum Iterations were tuned across three different values to find the range best fit for this dataset: 50, 300, and 500.

The Affine Transformation matrix returned by MATLAB first had to be transposed to put into standard form. Before the Affine Transformation could be applied to the whole Xenium Image, the Xenium Image and Affine Transformation had to go through the following postprocessing. The Xenium image was downsized by 50 percent to match the resolution of the crop. Additionally, since a crop was used to find the transformation matrix, the returned x and y-translation values are indicative of the translation within the crop rather than the translation across the whole image. Therefore, the x and y-translation values in the matrix needed to be updated to accommodate for the location of the crop. This was done using the following equation:

$$U = |(MS)T - F| \quad (2)$$

Where  $F$  is the top left coordinate of the crop in the fixed image ( $[x, y, 1]$ ),  $M$  is the top left coordinate of the crop in the moving image ( $[x, y, 1]$ ),  $S$  is the 3x3 scaling matrix applied to downsize the crop,  $T$  is the optimized Affine Transformation matrix from the pipeline sans translation values, and  $U$  are the updated translation values ( $[tx, ty, 0]$ ). The translation values were updated in the Affine Transformation matrix, and the matrix was applied to the whole Xenium Image.

#### 4.4. Cell Segmentation

The Cell Segmentation task focused on the Xenium Dataset from human breast tumor FFPE from the 10x Genomics site, which gave access to the “in-house” trained cell boundaries, cell-segmented masks, nucleus masks, and nucleus center points from the outs zip file within the `xenium_out` zip file. To read the data from within the outs file, code from the reading zarr files tutorial [4] was used, supplying additional information on how to properly access and index into the provided folders. The cell segmentation mask and nucleus segmentation mask were the same shape as the Xenium OME-TIFF Image (35416 x 25778) whereas the  $x, y$  nucleus center information was stored in 1xN

arrays with  $N$  as the number of detected nuclei. Given the large size of the data, smaller crops were taken to make intermediate training and testing stages more manageable.

CellCutter finds nucleus markers either through an input of a nucleus image, where it then runs blob detection to identify the nuclei or through an additional nucleus segmentation task which identifies the nuclei given the input image. Given that 10x Genomics generates robust nucleus segmentation using Cellpose and this information is accessible through the dataset, for the purposes of this project the 10x genomics nucleus segmentation was used in lieu of retraining through CellCutter. Specifically, this project relied on the 10x genomics nucleus segmentation masks and centers.

CellCutter uses blob detection to generate nucleus markers given a nucleus image. However, given the nucleus segmentation mask provided by 10x genomics, blob detection yielded inaccurate descriptors for the nuclei. Since 10x Genomics has nucleus  $x,y$  center-points for each identified nucleus in the image, these center-points were externally processed to be nucleus marker descriptors for the CellCutter pipeline.

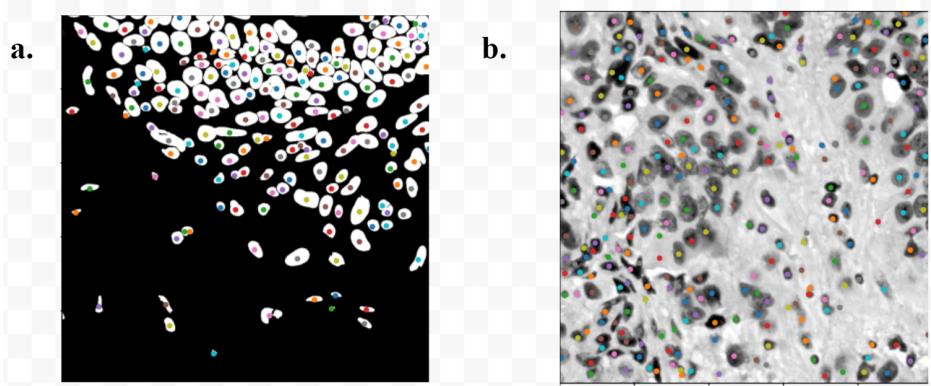
#### 4.5. Cell Segmentation: Generating Nucleus Markers

To compute nucleus markers given the 10x Genomics nucleus segmentation masks and  $x,y$  nucleus centers the following data processing had to be performed. The  $x,y$  nucleus centers were defined in physical space (microns) and needed to be converted to image space (pixels). 10x Genomics has a homogeneous transformation matrix as part of the `output/masks` files to convert information from physical space to image space. This matrix was applied to the  $x,y$  nucleus centers to convert the values into pixel coordinates to be usable markers for the cell segmentation task. Additionally, the  $x,y$  centers were filtered to include only those centers within the boundaries of the crop. Finally, the centers were mapped to the coordinates in the current image crop space. This process was completed using the projects defined `markers` function.<sup>3</sup>

To access that this process was completed correctly, the nucleus centers within a crop were plotted on top a corresponding crop of the nucleus segmentation mask and Xenium H&E image (Figure 7).

---

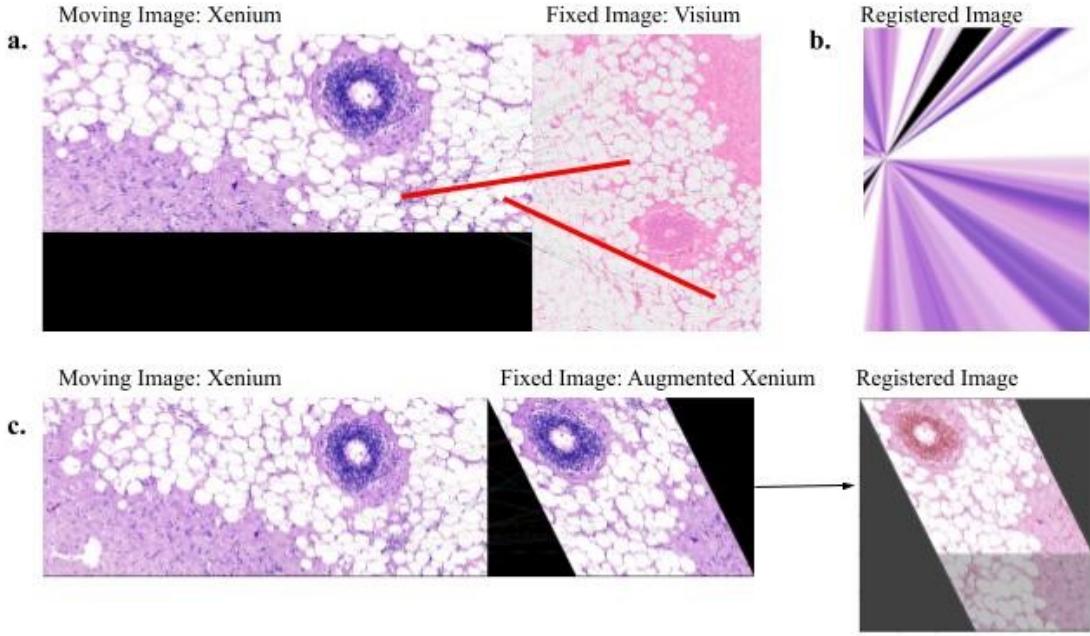
<sup>3</sup>See `markers` function from Cell Segmentation section in the `Cell_Segmentation` Notebook on GitLab



**Figure 7: Nucleus markers plotted on top of the 10x Genomics nucleus segmentation mask (a) and the H&E image provided for this tissue sample (b). Nucleus markers identified by colorful dots on top of the image and signify the nucleus centroid as found by the 10x Genomics nucleus segmentation task.**

#### 4.6. Cell Segmentation: CellCutter

Crops of the Xenium H&E image and IF image were sent through the CellCutter pipeline using provided demo [15] as a guideline. First the cell mask, distinguishing cell space from cell-free space, was obtained using cellcutter’s `graph_cut` function which takes the following parameters: `image`, `prior`, `max_weight`, `sigma`. Each of the parameters besides the image were tuned to find a mask that most accurately resembled the separation of the two spaces. The mask along with the processed markers, obtained by the previous step, were sent into cellcutter’s `Dataset` class, the model was established using CellCutters `UNet4`, and the model was trained on image crop sized 1000x1000 using cellcutter’s `train_self_supervised` function. The `train_self_supervised` function has `area_size`, `lam`, and `n_epochs` as hyper-parameters to fine-tune during the training process. The hyper-parameters were initialized with an area size of 320, `lam` of 3.5, and 60 epochs and the results of the self supervision using these parameters were recorded before tuning using grid search to find the most sensitive and effective values.



**Figure 8: The moving image and fixed image as defined as the Xenium Crop and Visium Crop respectively. Example of incorrectly matched descriptors identified by the solid red lines (a). From the matched descriptors, the product of the SIFT implementation is a highly distorted image (b). SIFT and RANSAC was attempted on the Xenium image and an augmented copy, which resulted in a successful registration (c).**

## 5. Evaluation

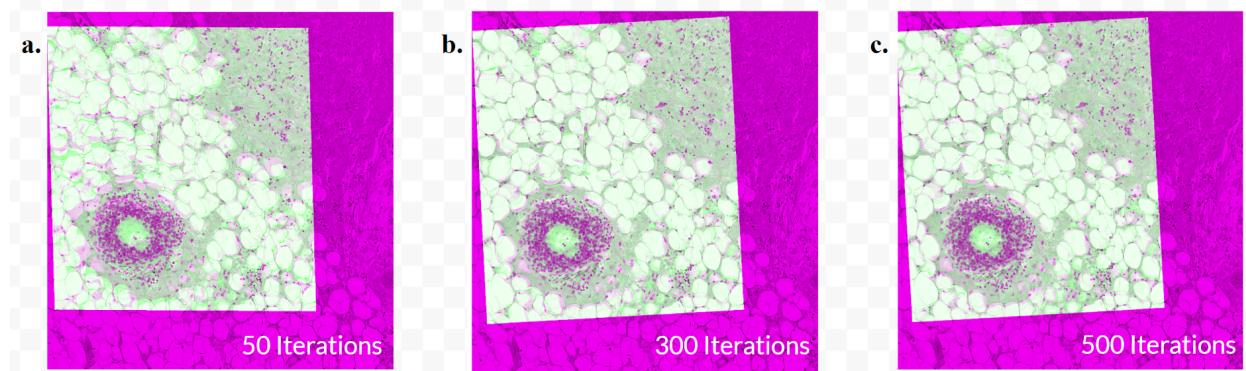
### 5.1. Image Registration: SIFT

Traditional feature-based image registration methods such as SIFT for finding and matching descriptors and RANSAC for best fit matrix failed to establish a mapping from the Xenium and Visium Datasets. Furthermore, SIFT failed to find valuable keypoints and properly match them as displayed in Figure 8a. This resulted in a highly morphed and distorted image upon registration (Figure 8b). To confirm that SIFT and RANSAC is applicable to an image of this type and solely fails on mapping the two datasets, an additional trial was conducted where a copy of the Xenium Crop was augmented and then image registration was performed between the Xenium crop and the augmented copy resulting in a successful registration (Figure 8c). Identical issues occurred when registering gray-scale Xenium and Visium crops. SIFT's inability to extract valuable keypoints between the two datasets suggests that although each image uses the same staining methods (H&E), the difference in

resolution and intensity values between Xenium and Visium increases the complexity of the task and renders SIFT ineffective.

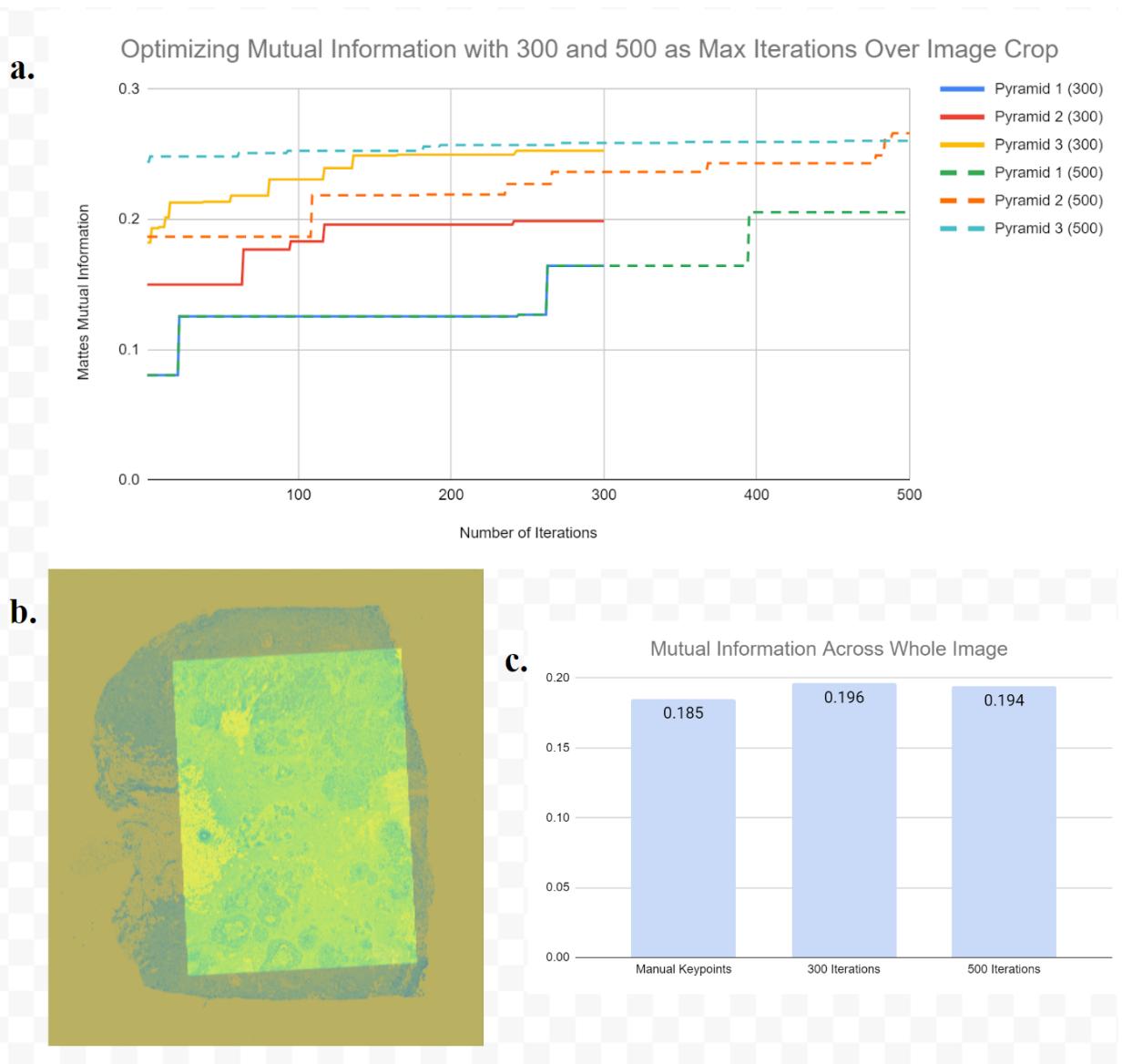
## 5.2. Image Registration: Intensity Based

The performance of the Intensity Based Image Registration on the image crops was tracked across the pyramid levels and iterations using MATLAB's Mattes Mutual Information Metric. Additionally, the performance on the crop level was tracked visually, noting how closely the cells aligned between the two images. From the visual analysis, it was immediately noticeable that 50 iterations was not sufficient in tuning the transformation matrix, as the cells are drastically misaligned (Figure 9a). However, with the increase of maximum iterations to 300 and 500, both parameters produce adequate alignments as there are few areas that have pink and green misaligned in the image (Figure 9b, Figure 9c). Additionally, both 300 and 500 maximum iterations produce similar visual results.



**Figure 9: Registration of the Xenium Crop (green) with the Visium Crop (pink) through the MATLAB script with Maximum Iterations in the Optimizer set to 50 (a), 300 (b), and 500 (c). Misalignment in (a) is seen through noticeable non-overlapping areas of green and purple as well as a less crisp picture as those seen in (b) and (c).**

To further distinguish the performance between 300 and 500 maximum iterations within the optimizer, Mattes Mutual Information was tracked across each iteration and pyramid level and is displayed in Figure 10a, where the dashed lines signify different pyramid levels for 500 maximum iterations and the solid lines for 300 maximum iterations. From the Mattes Mutual Information Metric, it is noticeable that 500 maximum iterations performs slightly better on the crops than 300 with finishing values of 0.2602 and 0.2527 respectively.



**Figure 10:** The results of the Transformation Matrix found via Intensity Based Image Registration through MATLAB. The tracking of the optimization for 300 and 500 Maximum Iterations where the solid lines denote the pyramid levels within 300 Maximum Iterations and the dashed lines denote the pyramid levels within 500 Maximum Iterations (a). The bottom half shows the resulting image registration (b), where the highlighted area is representative of the Xenium Image and the dull area representative of the Visium Image, as well as the mutual information metric across the whole image when registered with manual keypoints (left), 300 Maximum Iterations (center), and 500 Maximum Iterations (right) (c).

The registration of the whole Xenium and Visium images was evaluated using a Mutual Information Metric defined by 1. Applying the resulting transformation matrix to the entire Xenium Image and measuring the Mutual Information between the two whole images revealed a decrease in performance from 300 iterations to 500 iterations(Figure 10c), suggesting that increasing the

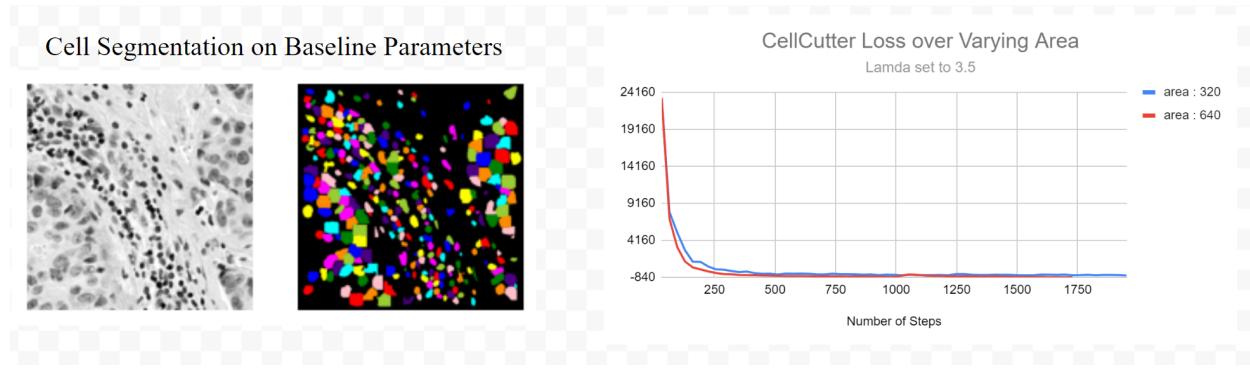
maximum iterations to 500 iterations caused the optimizer to over-fit on the crop. The mutual information across the whole image when using the resulting transformation matrix from 300 maximum iterations was 0.196 whereas the mutual information for 500 maximum iterations was 0.194, which is a 1% decrease. When referencing Figure 10a and looking at the solid lines, there is a substantial increase between the finishing stage of each pyramid level (0.1645, 0.1987, and 0.2527 respectively). However, when looking at the dashed lines, there is a substantial increase between the finishing stages of pyramids one (0.2054) and two (0.2661), but a decrease in finishing value from pyramid two (0.2661) to pyramid three (0.2602). Furthermore, there is little change throughout the whole duration of pyramid three, 0.2441 to 0.2602, and it performs only slightly better than the finishing stage of the solid line. Although there is an increase of performance in the optimizer for 500 maximum iterations with respect to 300 maximum iterations, it is both not substantial enough to warrant the more fine tuned transformation matrix and it results in a decreased performance when projected onto the whole dataset. Thus supporting the claim that 500 maximum iterations over-fit to the cropped dataset and was no longer generalizable across the entire image.

The Intensity Based Image Registration Method performed 5.6% better than the previous method when analyzing the mutual information. Given a matrix defined by the previous method of manual keypoint identification and RANSAC, the mutual information across the whole registered image was 0.185 whereas the mutual information across the whole image using Intensity Based Image Registration was 0.196 (Figure 10c).

### 5.3. Cell Segmentation

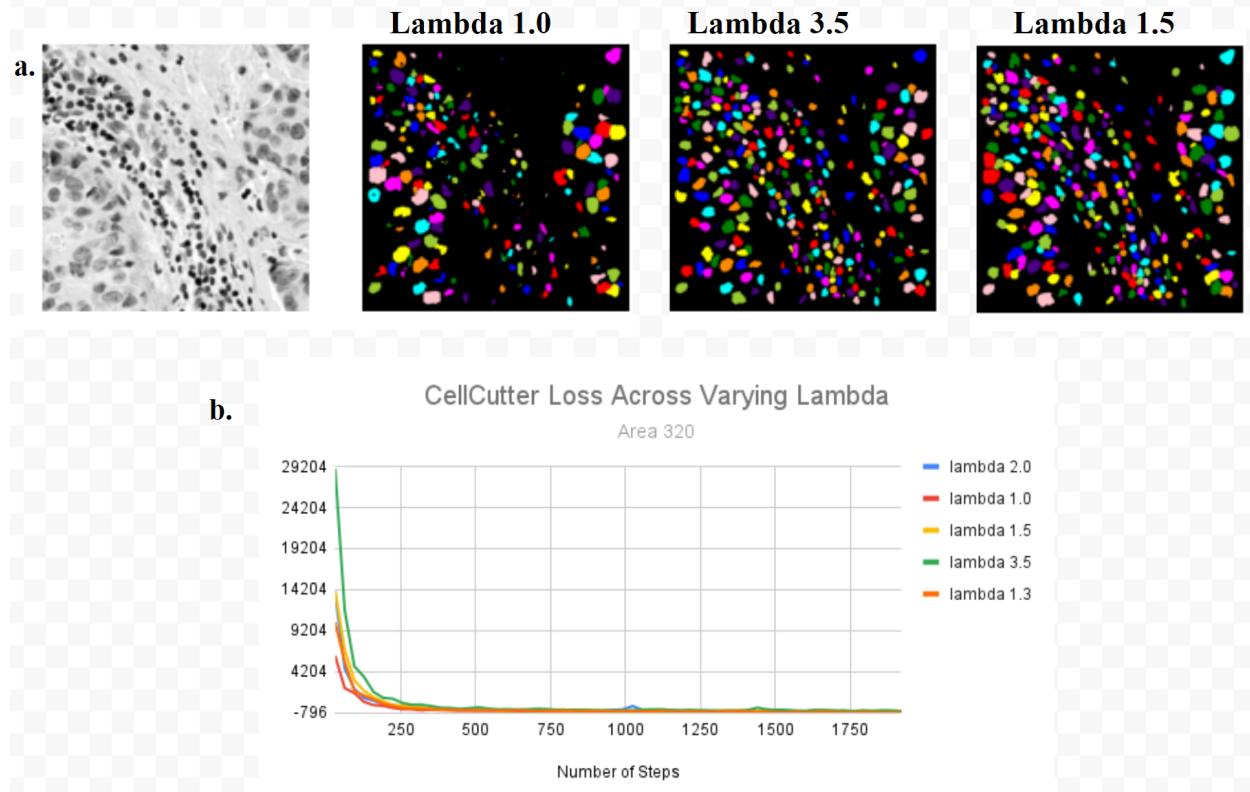
A baseline performance of Cell Segmentation was evaluated by setting the parameters as described above. The segmentation results given the baseline and the loss curve are displayed in Figure 11. The baseline parameters under-performed as cells showed a trend to be generalized with a rectangular structure.

To tune the model the area size and lambda parameters were modified. Cellcutter showed high performance with area size set to 640; however, upon initial runs this produced block-like structures



**Figure 11:** The baseline cell segmentation mask (left) showing less refined structures for larger cell areas and the loss curve over training for area size 320 and 640 (right) showing the stabilization of the loss.

for the cell segmentation for this dataset. Therefore for initial testing, area size was decreased to 320 and five different lambda values were experimented with: 1.0, 1.3, 1.5, 2.0, and 3.5. Figure 12 shows the training loss curve for each trial and the resulting segmentation mask.

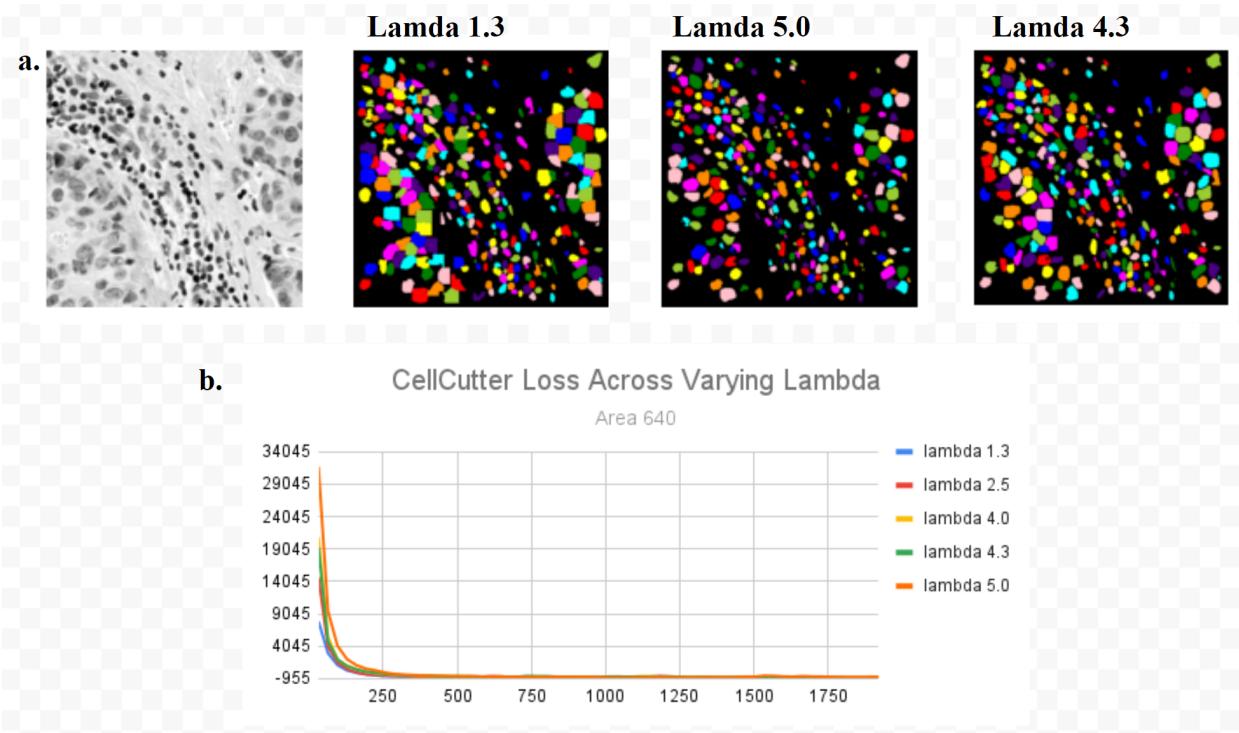


**Figure 12:** The cell segmentation across varying a constant area size of 320 and varying lambda values, displayed are the H&E image crop and the cell segmentation masks for lambda 1.0 (extreme low lambda case), 3.5 (high lambda case), and 1.5 (best performing range) (a). The training loss across each lambda value the model was trained at (b).

With area size set to 320, large lambda values produce a mask closely resembling the nucleus segmentation. That is, the model is able to pick up the dark colors, which indicate the nuclei, but is unable to expand outwards to find cell boundaries. As lambda is decreased to 1.0, large cells are represented by larger areas in the mask but there is a loss of representation for smaller cells in more congested areas. A lambda value between 1.5 and 2.0 yielded the best representation with the larger cells expanded past the nucleus segmentation mask while maintaining a representation for the smaller cells. Additionally, with lambda between 1.5 and 2.0 the cell space develops more as clusters and there is less of the mask lost to the background. The model was trained for each trial until the loss curve stabilized.

The second batch of trials were conducted with area size set to 640, the recommended value for the dataset initially trained by cellcutter. Figure 13a shows the cell segmentation masks across the lambda values for an area size of 640. This trial used lambda of 1.3, 2.5, 4.0, 4.3, and 5.0. Initially when running the trials, similar lambda values were used as with the area size of 320. However, it was quickly noticeable that with the higher area size the model malformed given small lambda values. Instead the model performed the best with higher values between 4.0 and 4.3. In low lambda value ranges the cells appear rectangular in structure with sharp edges. As lambda increases the edges of the cells smooth out and they start to take a more intentional form. However, similarly with the smaller area size, when lambda is increased too far the cells shrink and the mask becomes more representative of the nuclei instead of the cell boundaries. The model on all trials was trained until the loss curve stabilized as shown in Figure 13b.

To address efficiency, the baseline trial was first run on a standard Google Colab CPU, where the run time drastically increased between area size of 320 and 640. However, when running grid search the run time was switched to a Google Colab TPU, allowing the search to be more quickly performed. The trials for both area sizes were run for 60 epochs across the board, and the model was observed to take about double the time for area size of 640 as opposed to area size 320 (Figure 14).



**Figure 13:** The cell segmentation across varying a constant area size of 640 and varying lambda values, displayed are the H&E image crop and the cell segmentation masks for lambda 1.3 (extreme low lambda case), 5.0 (high lambda case), and 4.3 (best performing range) (a). The training loss across each lambda value the model was trained at (b).



**Figure 14:** The average runtime of training the model given an area size of 640 (620.47 seconds, 251.17 SD) and 320 (265.08 seconds, 73.62 SD)

## 6. Discussion and Future Work

Automating the Image Registration Task proved successful, as there was a significant increase in mutual information from the previous method of predefining features to the automated Intensity Based Image Registration. An increase in performance of mapping between the high dimension and high resolution datasets leads to more accurate representations of the gene sequences when attempting to fill in the sparse data. Both low resolution and sparse data are currently existing drawbacks and limitations of spatial transcriptomics, resulting in an ongoing research problem for scientists in this field [14, 24]. Through successfully creating a mapping, both issues are addressed via the introduction of subcellular resolution using Xenium. This allows scientists to gain a whole picture view of the benefits of spatial transcriptomics (understanding tissue function, cell to cell communication, tumor boundaries, etc.) without having to sift through incomplete data.

The cell segmentation task was informative, but needs additional training to be more effective. The largest limitation of this was the models tendency to create a boundary based on the nucleus of the image rather than the cell border. This could be due to using the H&E image, where the nuclei are well defined but the cell boundaries have a lower intensity difference from the cell membrane. However, the main purpose of using this model was the create distinct limits on distinguishing cell space from cell-free space. In regards to this aspect the task was successful, the cell segmentation mask showed cell boundaries as remaining in the defined cell space. Additionally, one metric for evaluating the performance of cell segmentation via a self-supervised method is to measure if the number of cells detected is equivalent to the number of nuclei present in the image. Cellcutters use of nucleus markers enforces that cells are found around the markers.

### 6.1. Limitations

Due to the nature of a 3x3 Affine Matrix as a uniform transformation across the whole image, the Affine Transformation was found on a small crop of the data being sent to MATLAB rather than the whole image. This was due to the large size of the image and the efficiency of MATLAB on a small crop rather than the whole image. While the transformation is uniform across the image

and an accurate alignment of a crop should be representative of an accurate alignment of the whole image when using uniform transformation, this disregarded consideration towards any deformation of the image in locations outside of the crop.

Cellcutter was trained and found effective on Bright Field images and IF images. However, this study aimed to apply cellcutter to the H&E images directly, as there was no access to bright field images and the available IF image was very dark even after normalization. This lead to less predictable results as the model was extended to a type of image it has never seen before.

Limitations for both tasks could be found in the dataset and the available computational power. In order to efficiently evaluate metrics and run tests, the majority of the computation had to be performed on crops of the data rather than the whole image itself. Furthermore, the large size of the images would cause the RAM space on Google Colab to reach maximum, effectively shutting down the session when processing the images as a whole. This had the most affect on the cell segmentation task, as all model training was limited to a 1000x1000 crop of the image.

## 6.2. Future Work

To further create a robust, automated image registration task for the Xenium and Visium H&E images, this task looks towards finding a displacement field. The displacement field would allow for a non uniform alignment and mapping to be found between the two datasets, thus, giving the ability to accurately represent unique parts of each image. MATLAB has an algorithmic approach to finding the displacement field when registering two images [1], similar to the Intensity-Based approach. However, this would require the whole image to be entered in, either as a combination of separate crops or as the whole image being passed at once. Therefore, since the whole image has to be sent through, a learning based approach such as VoxelMorph may be the most useful. VoxelMorph is a deep learning model that was designed for medical image registration and has unsupervised capability for learning the deformation field between two images [9]. Furthermore, training VoxelMorph with the H&E images of choice may be able to produce a model that is sensitive to the differences between the Xenium and Visium images and generalizeable across other

tissue sections. This would be time consuming on the training end, but develop an efficient pipeline for registering other Xenium and Visium images as this task becomes more popular.

Movement forward for Cell Segmentation would include the combination of a continued hyper-parameter search, the inclusion of the morphology images in the training data, to possibly be more representative of the images that cellcutter was previously trained on, and a more diverse training set of images with larger sizes. Sending larger images through cellcutter gives the model more information to train off of, which is crucial given that it is a self supervised method. Additionally, further work could be found in truly fine-tuning the masking feature to have very accurate boundaries of where cells can and cannot be.

## **7. Honor Code**

This paper is a representation of my own work in accordance with University Regulations - Claira Fucetola

## **8. Acknowledgements**

I would like to thank Professor Ben Raphael for advising me this semester, his feedback and help was invaluable. Additionally, I would like to thank Hongyu Zheng for meeting with me and helping me to understand the dataset, image registration task, and the current work that has been done for the registration. I would also like to thank my classmates in my seminar for providing feedback weekly at our meetings. Finally, I would like to thank my friend Maya for her countless support and willingness to help me and read over my work.

## References

- [1] (n.d.) Estimate displacement fields. MathWorks. [Online]. Available: <https://www.mathworks.com/help/images/ref/imregdemons.html>
- [2] (n.d.) Intensity-based automatic image registration - matlab & simulink. <https://www.mathworks.com/help/images/intensity-based-automatic-image-registration.html>. MathWorks.
- [3] 10x Genomics. (n.d.) Human breast dataset explorer - analysis. [Online]. Available: [https://cf.10xgenomics.com/samples/xenium/1.0.1/Xenium\\_FFPE\\_Human\\_Breast\\_Cancer\\_Rep1/Xenium\\_FFPE\\_Human\\_Breast\\_Cancer\\_Rep1\\_analysis\\_summary.html](https://cf.10xgenomics.com/samples/xenium/1.0.1/Xenium_FFPE_Human_Breast_Cancer_Rep1/Xenium_FFPE_Human_Breast_Cancer_Rep1_analysis_summary.html)
- [4] ——. (n.d.) Overview of xenium zarr output files - official 10x genomics support. [Online]. Available: <https://www.10xgenomics.com/support/in-situ-gene-expression/documentation/steps/onboard-analysis/xenium-outputs-zarr#transcripts>
- [5] ——. (n.d.) Performing 3d nucleus segmentation with cellpose and generating a feature-cell matrix. [Online]. Available: <https://www.10xgenomics.com/resources/analysis-guides/performing-3d-nucleus-segmentation-with-cellpose-and-generating-a-feature-cell-matrix>
- [6] ——. (n.d.) Spatial transcriptomics. [Online]. Available: <https://www.10xgenomics.com/spatial-transcriptomics>
- [7] I. 10x Genomics. (2022, December 8) 10x genomics commercially launches xenium platform for in situ analysis. [Online]. Available: <https://www.prnewswire.com/news-releases/10x-genomics-commercially-launches-xenium-platform-for-in-situ-analysis-301698748.html>
- [8] K. Aoki *et al.*, “Unified approach to (1 + 1) evolutionary algorithm on discrete linear functions,” *Journal of Robotics, Networking and Artificial Life*, vol. 6, no. 1, p. 56–59, 2019.
- [9] G. Balakrishnan *et al.*, “Voxelmorph: A learning framework for deformable medical image registration,” *IEEE Transactions on Medical Imaging*, vol. 38, no. 8, pp. 1788–1800, 2019. Available: <https://doi.org/10.1109/tmi.2019.2897538>
- [10] M. Chen *et al.*, “Image registration: Fundamentals and recent advances based on deep learning,” in *Machine Learning for Brain Disorders*, O. Colliot, Ed. New York, NY: Humana, July 23 2023, ch. 14. Available: <https://www.ncbi.nlm.nih.gov/books/NBK597490/>
- [11] M. Delgado. (n.d.) Shannon information and the mutual information of two random variables. Available: <https://www.math.uchicago.edu/~may/VIGRE/VIGRE2008/REUPapers/Delgado.pdf>
- [12] N. U. Din and J. Yu, “Training a deep learning model for single-cell segmentation without manual annotation,” *Nature News*, December 14 2021. Available: <https://www.nature.com/articles/s41598-021-03299-4>
- [13] S. Drosté, T. Jansen, and I. Wegener, “On the analysis of the (1+1) evolutionary algorithm,” *Theoretical Computer Science*, vol. 273, no. 1-2, pp. 51–81, April 2002. Available: <https://www.sciencedirect.com/science/article/pii/S0304397501001827>
- [14] J. Du *et al.*, “Advances in spatial transcriptomics and related data analysis strategies,” *Journal of Translational Medicine*, vol. 21, no. 1, 2023.
- [15] Google. (n.d.) Cellcutter demo. Google Colab. Available: <https://colab.research.google.com/github/jiyuuchc/cellcutter/blob/main/notebooks/demo.ipynb>
- [16] C. Guo. (2018) Multi-modal image registration with unsupervised deep learning. <https://dspace.mit.edu/bitstream/handle/1721.1/123142/1128823285-MIT.pdf>.
- [17] A. Janesick *et al.*, “High resolution mapping of the breast cancer tumor microenvironment using integrated single cell, spatial and in situ analysis of ffpe tissue,” 2022.
- [18] OnePlusOneEvolutionary. (n.d.) One-plus-one evolutionary optimizer configuration - matlab. Available: <https://www.mathworks.com/help/images/ref/registration.optimizer.oneplusoneevolutionary.html>
- [19] C. Stringer *et al.*, “Cellpose: A generalist algorithm for cellular segmentation,” *Nature Methods*, vol. 18, no. 1, p. 100–106, 2020.
- [20] L. Tian, F. Chen, and E. Z. Macosko, “The expanding vistas of spatial transcriptomics,” *Nature Biotechnology*, vol. 41, no. 6, p. 773, 2022.
- [21] C. Williams *et al.*, “An introduction to spatial transcriptomics for biomedical research,” *Genome Med*, vol. 14, 2022. Available: <https://doi.org/10.1186/s13073-022-01075-1>
- [22] J. Woo, M. Stone, and J. L. Prince, “Multimodal registration via mutual information incorporating geometric and spatial context,” *IEEE Transactions on Image Processing*, vol. 24, no. 2, p. 757–769, 2015.
- [23] C. Xing and P. Qiu, “Intensity-based image registration by nonparametric local smoothing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 2081–2092, 2011. Available: <https://doi.org/10.1109/tpami.2011.26>
- [24] P. Zhao *et al.*, “Modeling zero inflation is not necessary for spatial transcriptomics,” *Genome Biology*, vol. 23, no. 1, 2022.