Optimal Separation, Detection, and Analysis of FISH Images

Aparna Rajpurkar
Department of Genetics
arrajpur@stanford.edu

Maggie Engler
Department of Electrical Engineering
mengler@stanford.edu

Abstract—**Fluorescence In-Situ Hybridization (FISH) is a widely-used technology in biological research. FISH images show the locations of specific nucleotide-based species in the cell, such as cellular RNA transcripts, genomic regions of interest such as specific genes or non-coding regions, and chromosomal features such as telomeres among others. Though manual counting of spots remains the most common methods of processing FISH images by biological researchers, the advent of high throughput labeling technologies makes this extremely difficult, as high throughput single molecule FISH images may have on the order of thousands of individual spots per cell. Strategies exist to automatically analyze FISH data, but few are broadly applicable. We implemented and analyzed the performance of 4 algorithms for FISH spot counting, as well as 3 ensemble methods and an additional post-processing segmentation strategy to refine FISH spot detection. We found that K-means based segmentation performed best when measured by Precision Recall plots. We found that adding a further spot detection segmentation step after the original algorithm improved performance, but further optimization is required. Finally, we qualitatively assessed our algorithms on a new mouse placental cell dataset and found biologically meaningful results.**

I. Introduction

Fluorescence In-Situ Hybridization (FISH) is a widely-applied and extremely useful technology in biological research. FISH allows specific nucleotide-based species within cells to be marked by fluorescent oligo probes. Cells can then be imaged and the locations and count of all species of interest can be directly visualized. FISH has been applied to numerous studies, such as marking the location of every RNA in the cell, marking chromosomes for counting, and visualizing specific genes of interest in multiple cells, among many other applications.

Yet, most researchers manually count FISH spots in their images. This is possible when the number of expected FISH spots is very low, such as in chromosome counting or when visualizing the location of a specific RNA of interest. But recently, new technologies have emerged that allow high-throughput, specific labeling of thousands of individual nucleotide-based species, specifically RNA. In these experiments, it is not possible to manually count each FISH spot.

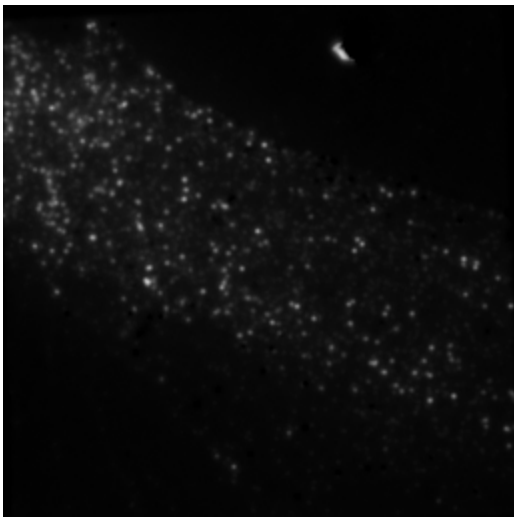
FISH images present unique challenges to automatic segmentation. The background level is often variable, high, and can contain many kinds of background noise (flecks, diffuse fluorescence, biased auto-fluorescence, bright artifacts, non-uniform illumination, low levels of random hybridization). In addition, FISH spots can vary in size due to the fluorophore intensity (a usually uninformative quality in single molecule resolution systems), have variable brightness, and vary slightly in shape. The resolution of FISH images falls over a wide range, depending on the microscope used and specific experiment [1].

Our goal is to test multiple strategies for FISH spot detection, quantify their performance on a published ground-truth MERFISH dataset [2], then apply them to a new dataset and examine the biological relevance of their output. Most biological images have no strong ground truth, given that the accuracy of manual counting varies between individuals and each system can behave very differently, making it difficult to apply general rules. Therefore, the usefulness of a new technology is frequently assessed not only against a quantitative dataset in which ground truth counts are known, but also against the biological relevance of its results. That is, for a biological system whose general trends are known but a quantitative metric in the same modality as the new technology may not yet exist, does the new technology recapitulate trends previously observed?

We will test our algorithms both against a dataset with quantitative FISH spot counts and locations, and also against a system with an expected biological trend.

II. Datasets

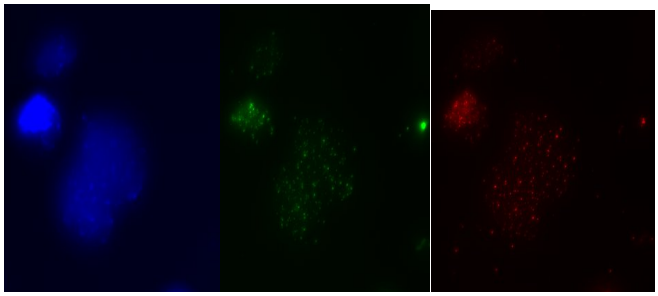
A. MERFISH



Multiplexed Error-Robust Fluorescence In-Situ Hybridization (MERFISH) is a high-throughput imaging technology that allows all the RNA transcripts in a cell to be simultaneously imaged at single molecule resolution using an encoding system for fluorescent oligo probes [2]. Unlike traditional FISH protocols, MERFISH can produce thousands of spots, each of which must be counted across 14-16 individual hybridization frames per cell. The images in our testing dataset have been previously analyzed by the authors in the original MERFISH publication and ground truth centroids of detected spots are available.

MERFISH images tend to have variable levels of background noise, much of which is produced by incompletely-quenched probes, yielding background spots that have a similar size and shape as real fluorescence spots. In addition, the FISH spots can be tightly clustered, which causes difficulty in segmentation as the images are also of low resolution.

B. Mouse Placental Cells



FISH-like protocols can also be used to individually image genomic regions of interest on the DNA, with different regions being marked by different fluorescent probes each of which is imaged in a separate fluorescence channel. This produces pseudo-colored images such as the above. The color is meaningless and only serves to make the gray-level image more visible. A DAPI image (blue) denotes nuclei of cells, which is important for segmentation of multiple cells in a single field of view. The dataset we used is new and unlabeled, therefore we were only able to qualitatively assess our algorithm performance on it. The two important regions imaged are the centromere (green) of individual chromosomes and the telomeres (red) of chromosomes. Using FISH against centromeres and telomeres allows chromosome numbers to be counted.

The dataset we use has two groups, an untreated sample of mouse placental stem cells and treated mouse placental stem cells, which are expected to contain a subset of cells that have differentiated. Differentiated placental stem cells are known to have extreme hyper-replication of the DNA, even up to 1000x [11]. However, it is unknown what happens to the DNA organization under such conditions. We used our algorithms to count the centromeres and telomeres of this dataset and compared untreated to treated samples.

This dataset has in general fewer real FISH spots of interest, but can vary between very low numbers (10-20 for cells in which not every chromosome was hybridized to a probe) to very high numbers (potentially up to 1000x as reported in the literature [11] for differentiated cells if chromosome count scales linearly with DNA content). It also has high levels of diffuse background noise.

III. Algorithms

The problem of extracting RNA spots from FISH images is essentially that of image segmentation, or separating the foregrounded spots from the background and noise. Based on a survey of object extraction methods [3], we identified four different approaches to implement and compare.

A. Locally Adaptive Threshold

The first and simplest method we consider is thresholding. Due to the presence of background noise, a global threshold derived by Otsu's method does not perform well on the cell images. Instead, we compute the optimal sensitivity factor and perform locally adaptive thresholding using that parameter. From MATLAB documentation, "A high sensitivity value leads to thresholding more pixels as foreground, at the risk of including some background pixels." [4] Therefore, we compute the output images from locally adaptive thresholding over the range of possible sensitivity factors (from zero to one), and count the number of spots identified at each value. Then, the sensitivity factor where the number of identified spots is least sensitive to changes in sensitivity should be the value at which we are including the lowest amount of noise. The binarized image corresponding to the minimum increase in number of connected components (RNA spots) as sensitivity increases is returned for each input image.

B. K-means Clustering

The second method we consider is region-based segmentation. In particular, k-means clustering is an unsupervised segmentation strategy that assumes pixels in the same region will have similar intensities. The algorithm iteratively clusters pixels into groups that minimize the difference in intensity between pixels of the same cluster. We implement k-means segmentation—separation of the input images into k clusters, with k as an input parameter—and return the processed image. The optimal output image will be the binarized image with the lowest overall sum of the squared "within-group" error, meaning the difference in intensity. We iterate over the parameter k and select the final value with a heuristic measure: we minimize the variance of the areas of the RNA spots detected, taking advantage of the fact that in our training data the spots we would like to extract are roughly the same size. The final output will be the segmented image with k clusters, binarized such that the foreground clusters, those assigned to the highest-intensity label, appear white and the k-1 background clusters are black.

C. Graph Cuts Segmentation

For a third segmentation method, we consider a relatively newer algorithm for image segmentation, from graph cuts based on energy minimization. The Graph Cut algorithm was introduced in [5] and expanded upon in [6] and [7]. We leveraged the MATLAB wrapper written by Shai Bagon and available at [8]. The two required parameters to Graph Cut energy minimization are an initial data cost matrix, which defines the cost of assigning a specific label (0 or 1) to a pixel, and a smoothness cost, which in the binary case is a 2-by-2 matrix s where $s_{i,j}$ describes describes the cost of labeling neighboring pixels with labels i and j. In object extraction tasks where the objects are larger, the smoothness cost might be more severe, but because the spots are small and disperse we selected a low smoothness cost for differing neighbor labels, and then a zero cost for matching neighbor labels. The data cost matrix is defined to be 1 if the intensity of a pixel is greater than the intensity threshold found by Otsu's Method and 0 otherwise. We also toyed around with a few other simple other data cost matrices, such as the distance from the pixel intensity to the average intensity, but they did not improve results. One may also specify a smoothness cost in the form of a sparse matrix, which allows for control over the smoothness costs of any two pixels in the image, rather than just neighbors, but this was ultimately not very relevant to our particular use case. The Graph Cut object uses these parameters to label each pixel, and their label becomes their binary value.

D. MSER Detection

The final method we consider is Maximally Stable Extremal Region (MSER) detection, an algorithm to isolate extremal regions that are "maximally stable," or do not change significantly in area as the threshold for extremity changes [9]. This idea is similar to the selection criterion for the locally adaptive thresholding—we are most interested in regions that are different enough in intensity from their surrounding pixels that they remain the same over a range of thresholds. Additionally, MSER has been found to do comparatively well with the detection of small regions [10], which is clearly the case with FISH images. We adjusted the parameters to the approximate range of the region sizes of the RNA spots in pixels. We examined different values of the parameter threshold delta, which controls the step size between intensity threshold levels, allowing more regions to be returned with a lower threshold delta. In the training set MSER detection tends to under-detect RNA spots, so we use a low threshold delta of 0.1.

IV. Accuracy and Performance

A. MERFISH Comparison to Ground Truth

Provided in the original MERFISH publication are coordinates of the centroids of the RNA-FISH spots found over 16 individual frames of a single cell. We compared the performance of our algorithms to the ground truth centroid coordinates. From the connected components output by our algorithms, we computed the centroids of each connected component and searched within a 5px window for a ground truth centroid within the window. We computed the precision and recall values for each of the 16 images and plotted the precision recall values for each of the 4 main algorithms we implemented (**Figure 1**).

K-Means Segmentation achieved the highest precision, with most images >0.5 precision and 8 images at 0.9 precision or better. Both K-Means Segmentation and Adaptive Thresholding had comparable recall that was higher than the other algorithms we tested. GraphCuts Segmentation and MSER had comparable precision and recall to Adaptive Thresholding on some images, but overall did not perform as well as K-Means Segmentation. All algorithms had low recall.

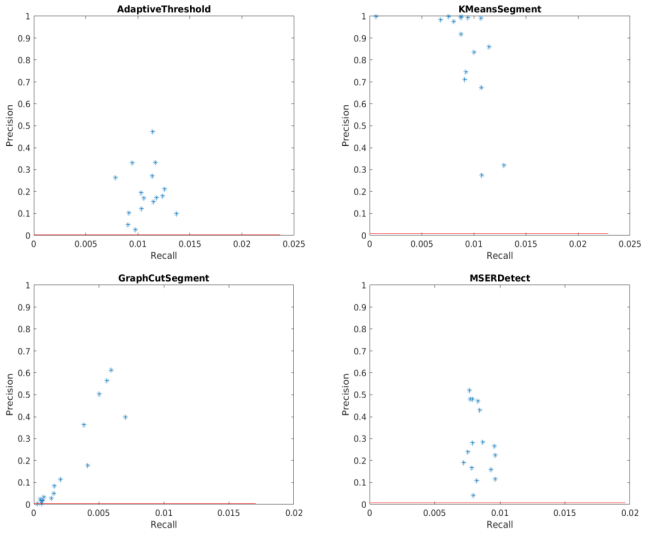


Figure 1: Precision recall graphs for each algorithm. *Each point represents the precision and recall value for a single image. The red horizontal line represents the threshold below which results are considered random, computed by P/(P+N).*

We also computed ensemble results for intersection, union, and voting (where minimum 2 of the algorithms were required to agree for a spot to be called) (**Figure 2**).

Here, VoteResults had the best performance of the ensemble algorithms, with a better balance of precision and recall than either Intersection or Union. The Intersection suffers very low recall as expected. Union achieves higher relative recall but the precision is reduced, while VoteResults achieves ~0.5 precision on some images and though the recall also suffers it performs better than the Intersection. However, it is clear that none of the ensemble algorithms perform as well as Adaptive Thresholding or K-Means Segmentation.

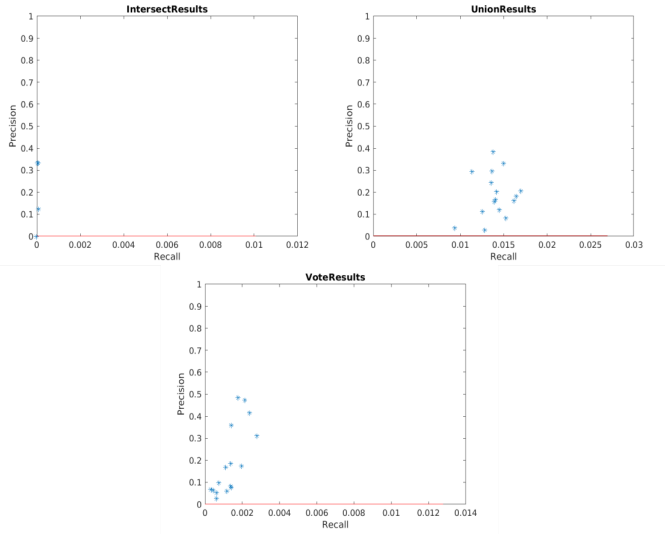


Figure 2: Precision Recall graphs for ensemble algorithm results.

B. Gaussian Mixture Model Segmentation

In order to improve the low recall we observed on all algorithms, we implemented a further post-processing

segmentation algorithm based on Gaussian Mixture Models (GMMs). We observed that the size of the connected components from the core algorithms were much larger than a typical FISH spot, and contained many such FISH spots. We decided to use a further segmentation method to refine the connected components by further segmenting them. We used the GMM-based method to model the gray image within a connected component as a mixture of Gaussian spots. FISH spots resemble small Gaussian points. Because the GMM method is very computationally expensive, this could only be run on a small area, therefore we focused the area containing FISH spots with a core algorithm and ran the GMM method as a post-processing step.

We implemented this as follows: first, taking a connected component generated by one of the core algorithms, find the gray pixel intensities for each pixel in the component, generate a histogram of pixels, then find a threshold using Otsu's method to discriminate between higher intensity pixels and lower intensity pixels. The coordinates of all high intensity pixels were then passed to a series of Gaussian Mixture Models. The initial centers were randomly placed. The GMMs were run for all values of k between 1 and the pixel area divided by 10, with a maximum of k = 75 to ensure it ran in reasonable time. We evaluated the performance of the GMMs for that connected component at different K values with the AIC measure, and chose the GMM with the lowest AIC. We extracted the positional clusters from the GMM, performed post-processing to ensure they were connected components, and added their centroids to the result.

We observed that the GMM Segmentation improved the recall of most of the algorithms and had no significant effect on precision for most algorithms except the GraphCut Segmentation (**Figure 3**). Adding the GMM segmentation had greatest effect on algorithms that produced the largest area connected components (especially GraphCuts Segmentation) and little to no effect on algorithms with good granularity already (K-means Segmentation).

However, recall was still very low for all algorithms following the GMM method. We hypothesize that the maximum limit of k=75 is the reason, it is possible that for some very large connected components there exist more than 75 potential FISH spots. This segmentation method could be improved by iteratively segmenting connected components optimally until they are small enough to run GMMs with k < 50.

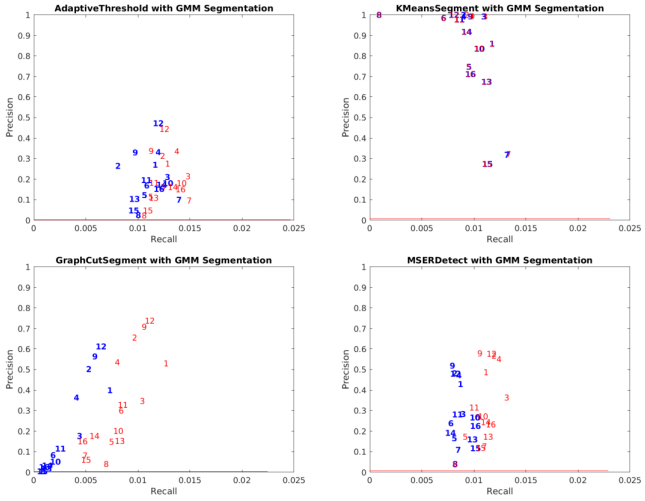


Figure 3: Precision Recall graphs for all core algorithm results with GMM Segmentation following the core algorithm. *Numbers represent the same image, blue is the original result without GMM Segmentation and red is the result with GMM segmentation.*

We also tested the effect of GMM Segmentation with the ensemble algorithms. We saw no strong improvement of the performance of ensemble algorithms with addition of the GMM (**Figure 4**).

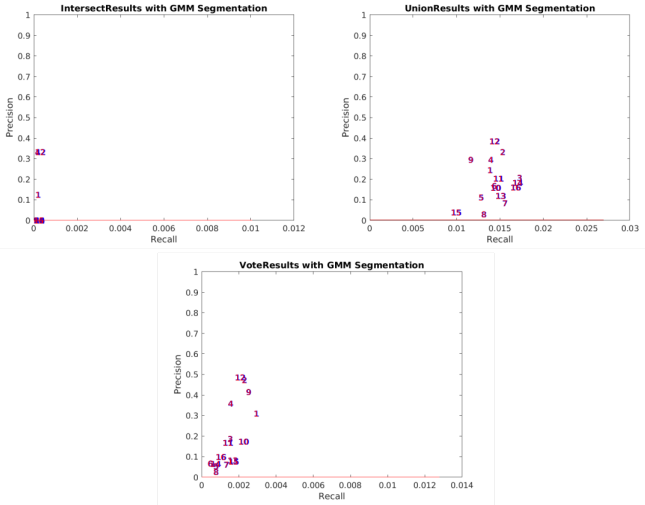


Figure 4: Precision Recall graphs for all ensemble algorithm results with GMM Segmentation following the core algorithm. *Numbers represent the same image, blue is the original result without GMM Segmentation and red is the result with GMM segmentation.*

V. Biological Relevance

A. Experimental Design

For qualitatively assessing the biological relevance of our FISH spot detection algorithms, we chose to use a biological question that centers around chromosome counting. Mouse placental stem cells have a normal number of chromosomes (40) before differentiation. Upon treatment with certain factors and induced differentiation, however, they hyper-replicate their DNA [11]. We chose to look at treated and untreated images from mouse placental cells, focusing on two marked regions: the centromere of the chromosome and the telomeres of the

chromosome. This will allow us to estimate relative chromosome counts. We expect to see a low number of both centromeric counts and telomeric counts for the untreated samples, and a high number in the treated samples. We ran our 4 core algorithms over both the treated and untreated samples, followed by GMM Segmentation. We did not run the ensemble algorithms, as they showed no improvement over the core algorithms.

B. Count Comparison of Treated vs. Untreated Samples

We assessed centromeric and telomeric spot counts by examining count boxplots (**Figure 5**). While it is clear that the core algorithms perform very differently on this dataset, there can be seen a clear trend in both centromere and telomere counts that the treated sample has cells with dramatically more detected chromosomes than the untreated samples. This trend is clear across all algorithms, though some are much higher than others. Additionally, in the treated cells, there are many cells which do not have high chromosome counts—this is expected because the treatment with differentiation factors had just begun when these images were taken, it is expected that only a few cells will have differentiated by this time, which is what we observe.

The total number of centromere and telomere counts is lower than expected in the untreated samples (mice have 40 chromosomes), but this is likely a result of not every chromosome being labeled combined with the low recall of all of our algorithms. Therefore, while these counts cannot be taken as an exact count of chromosomes in the cell, they still capture the key biological phenomenon in this experiment. These algorithms could be used, for example, to quickly determine whether differentiation is occurring in a sample without having to individually examine each cell's images by eye. This could be important in a time-dependent differentiation protocol, as the researcher could make a quick decision whether to modify a differentiation protocol or not based off of the estimate of differentiating cells given by our spot detection algorithms.

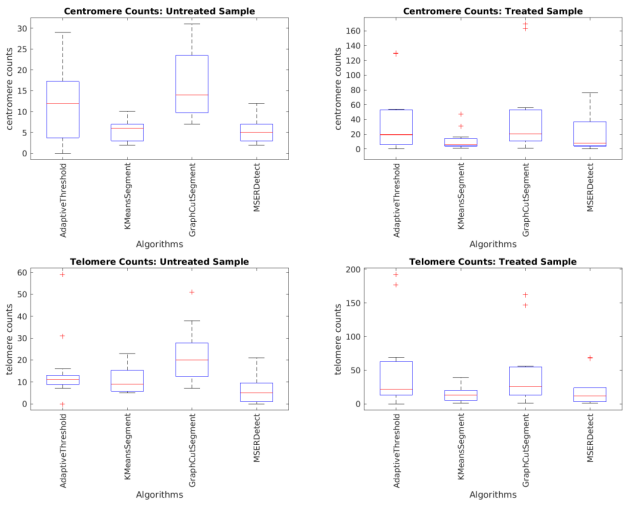


Figure 5: Boxplots of centromeric and telomeric counts in both treated and untreated mouse placental stem cells. *Each boxplot represents the counts for all cells in the sample, split by each algorithm used. These counts are post GMM segmentation.*

VI. Conclusion

We implemented four different strategies for FISH spot detection and three ensemble combinations of results, as well as a post-processing GMM segmentation method to refine the results. We assessed their performance quantitatively using a MERFISH dataset with ground truth, as well as qualitatively in a real experimental system. We found that K-means Segmentation had the best performance on the MERFISH dataset, but all algorithms were able to detect the expected trend in the mouse placental cell data. Future directions would be to optimize the recall of these algorithms, focusing on sensitivity, and modifying the GMM segmentation method to optimally segment large connected components into smaller pieces to remove the threshold on k <= 75. While there is much room for optimization and improvement in even the best performing methods, their results are not only quantitatively precise (though not sensitive), but also relevant in a real use case in the lab.

Acknowledgment

We thank Kay Kobak (Baker Lab) for providing the mouse placental cell images. The MERFISH data was downloaded from the examples page from the online MERFISH reference, associated with the original publication by Chen, Boettiger, and Moffit et al [2].

References

[1] Tsanov N, Samacoits A, Chouaib R, et al. smiFISH and FISH-quant – a flexible single RNA detection approach with super-resolution capability. Nucleic Acids Research. 2016;44(22):e165. doi:10.1093/nar/gkw784.

[2] Chen KH, Boettiger AN, Moffitt JR, Wang S, Zhuang X. Spatially resolved, highly multiplexed RNA profiling in single cells. Science (New York, NY). 2015;348(6233):aaa6090. doi:10.1126/science.aaa6090.

[3] Patel, Rajeshwary & Hinal Somani. (2016). A Survey on Object Extraction Using Image Segmentation and Adaptive Constraint Propagation. International Journal of Engineering Development and Research, 4(4). ISSN: 2321-9939. Retrieved from https://www.ijedr.org/papers/IJEDR1604102.pdf.

[4] The MathWorks, Inc. (2018). Binarize 2-D grayscale image or 3-D volume by thresholding - MATLAB imbinarize. Retrieved from https://www.mathworks.com/help/images/ref/imbinarize.html.

[5] Boykov, Yuri, Olga Veksler & Ramin Zabih. Fast Approximate Energy Minimization via Graph Cuts. Retrieved from http://www.cs.cornell.edu/rdz/Papers/BVZ-iccv99.pdf.

[6] Kolmogorov, Vladimir & Ramin Zabih. (2004). What Energy Functions Can Be Minimized via Graph Cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(2). Retrieved from http://www.cs.cornell.edu/~rdz/Papers/KZ-PAMI04.pdf.

[7] Boykov, Yuri, & Vladimir Kolmogorov. (2004). An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(9). Retrieved from http://www.csd.uwo.ca/~yuri/Papers/pami04.pdf.

[8] Bagon, Shai. (2006). Matlab Wrapper for Graph Cut. Retrieved from http://www.wisdom.weizmann.ac.il/~bagon.

[9] The MathWorks, Inc. (2018). Detect MSER features and return MSERRegions object - MATLAB detectMSERFeatures. Retrieved from https://www.mathworks.com/help/images/ref/imbinarize.html.

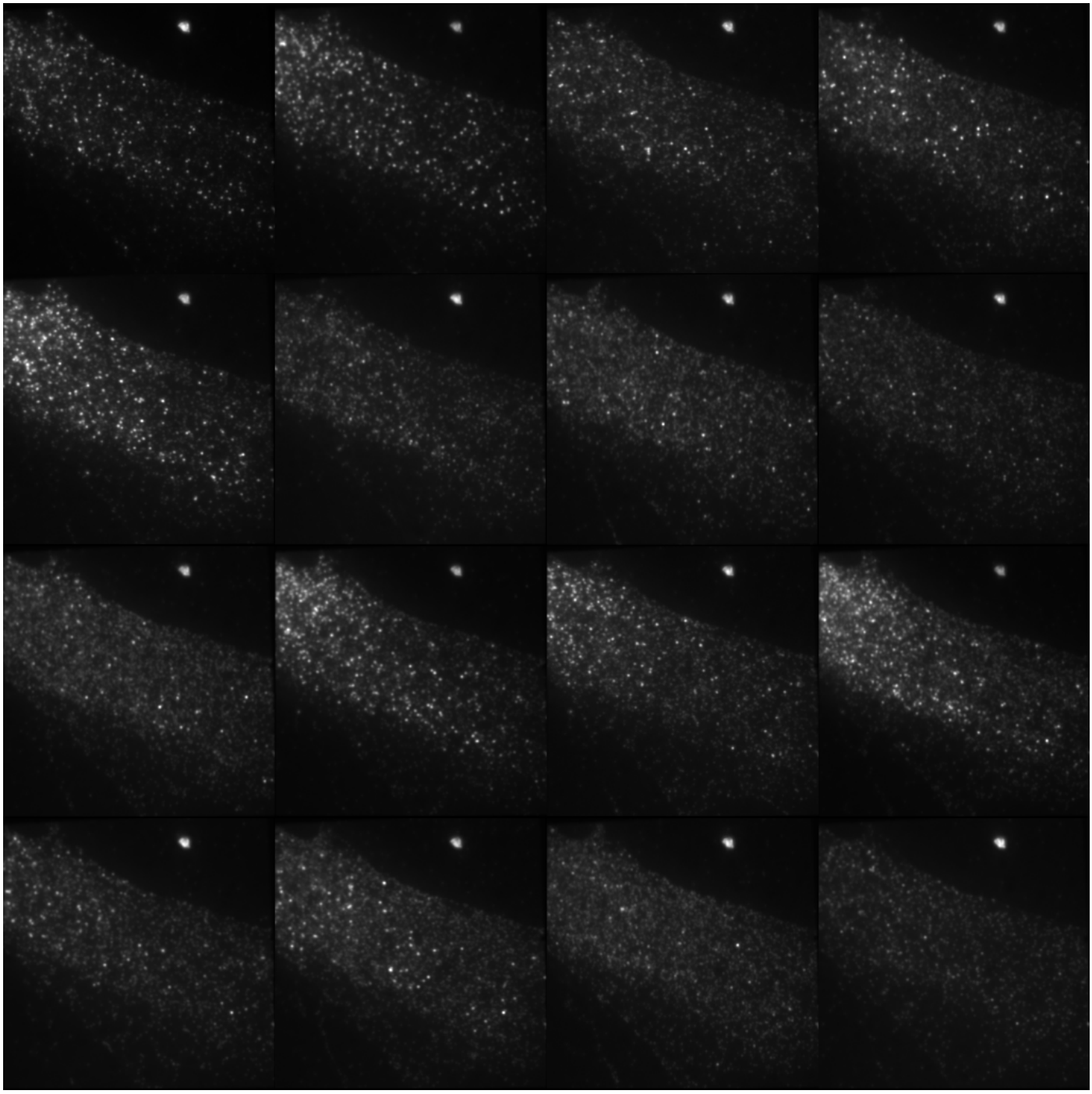
[10] Mikolajczyk, K & T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky & T. Kadir, & L. Van Gool. (2006). A Comparison of Affine Region Detectors. International Journal of Computer Vision. Retrieved from

http://www.robots.ox.ac.uk/~vgg/publications/papers/mikolajczyk05.pdf . DOI: 10.1007/s11263-005-3848-x.

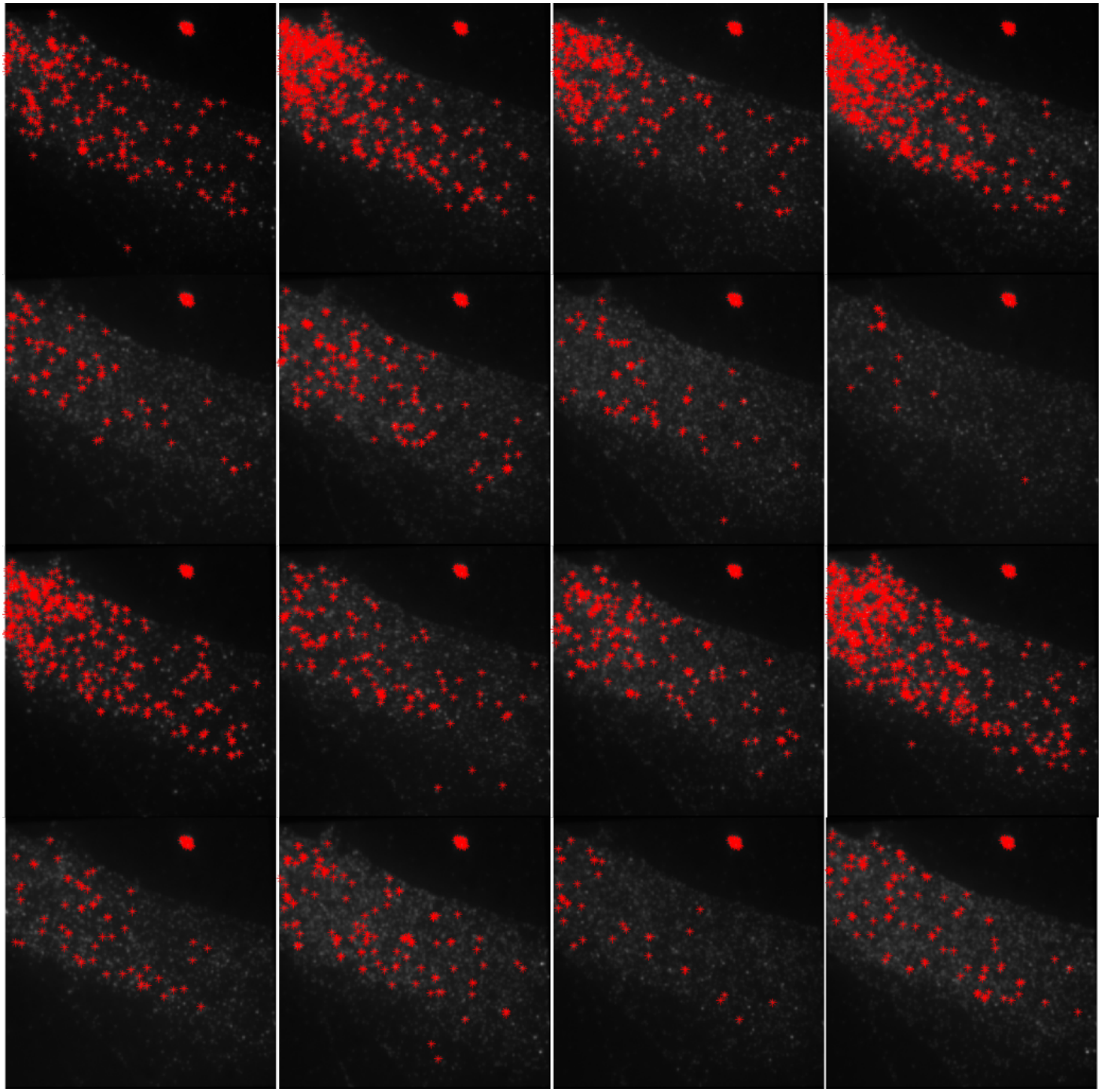
[11] Hannibal RL, Chuong EB, Rivera-Mulia JC, Gilbert DM, Valouev A, Baker JC. Copy Number Variation Is a Fundamental Aspect of the Placental Genome. Bartolomei MS, ed. PLoS Genetics. 2014;10(5):e1004290. doi:10.1371/journal.pgen.1004290.

Supplementary Images

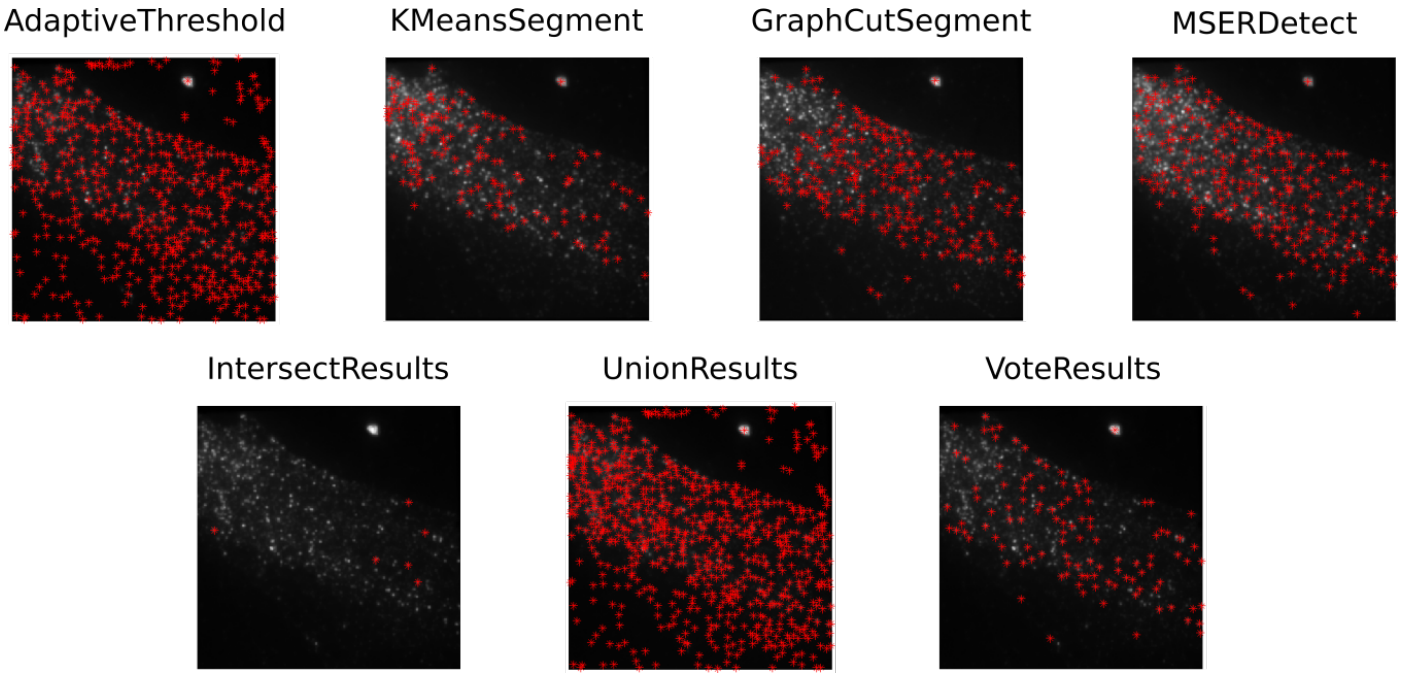
I. Raw MERFISH Images



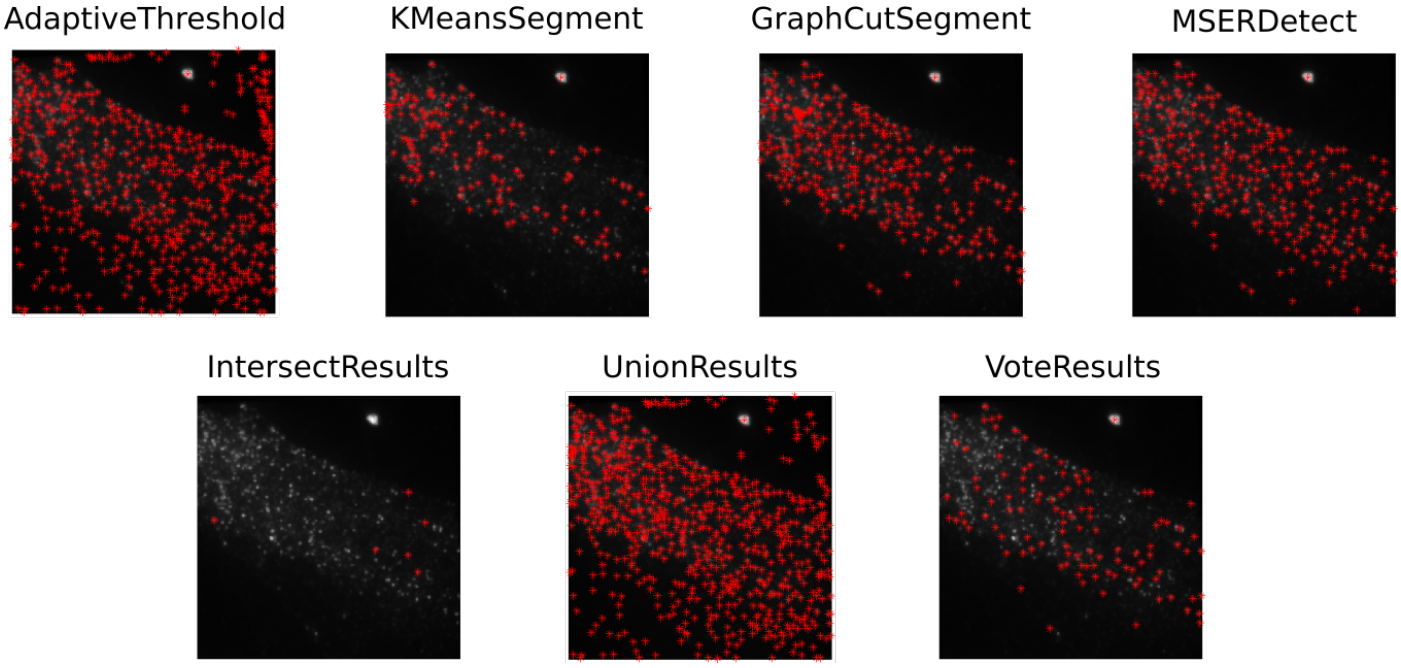
II. Ground Truth Centroid Labels



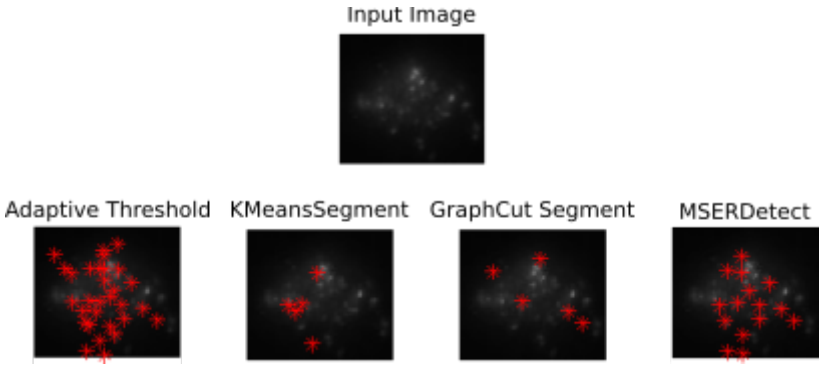
III. Centroid Labels Per Algorithm On MERFISH Image 1

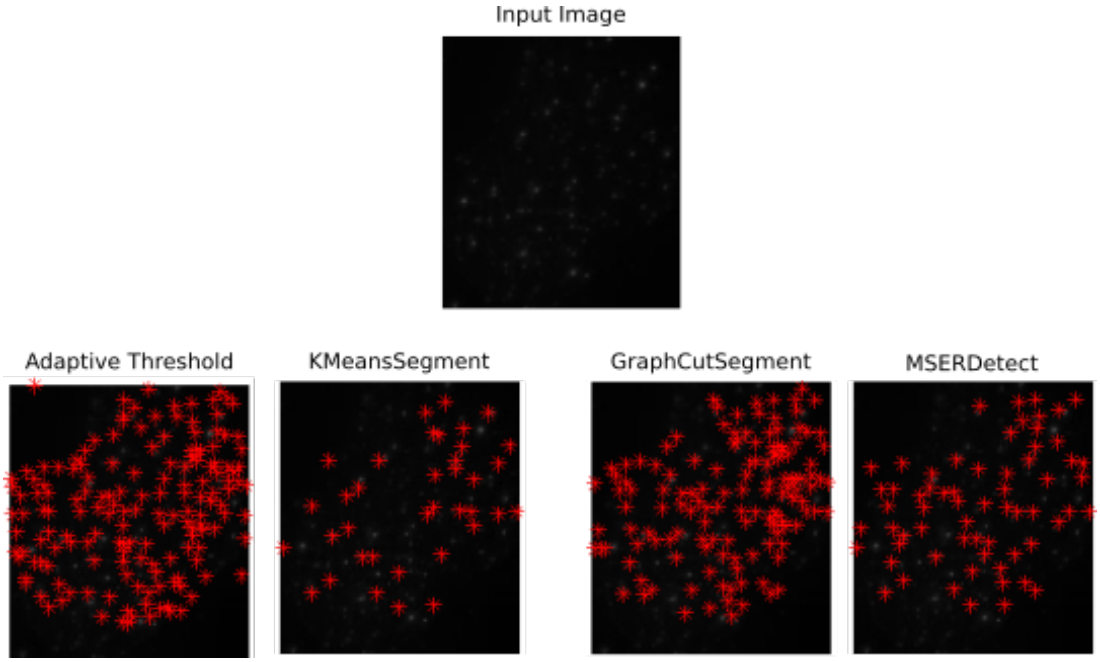


IV. Centroid Labels Per Algorithm On MERFISH Image 1 with GMM Segmentation



V. Example Results of Algorithms on Mouse Placental Cell Dataset





Appendix: Contributions

APARNA RAJPURKAR

Aparna designed and wrote the GMM Segmentation method and code (GMMSegment). She wrote the code for running the algorithms on both the MERFISH and mouse placental cell dataset, and designed and wrote the code for the quantitative and qualitative evaluations of the algorithms on the MERFISH and mouse placenta datasets. She wrote the short ensemble algorithms. She wrote the paper (with exception of the description of algorithms), the proposal, made all the figures, and acquired the datasets.

MAGGIE ENGLER

Maggie implemented the locally adaptive thresholding method and devised and added the selection criterion for parameter tuning. She wrote the K-means segmentation method and added the selection criterion for number of clusters after brainstorming possible criteria as a team. She also implemented the Graph Cuts segmentation method and the MSER detection method and documented the code for the four algorithms, as well as writing the algorithms section for the paper.