# Automated seqFISH data analysis : Mathematical Appendix

Pierre Bost, Uria Mor, Florian Mueller

December 23, 2019

## Contents

# 1 SeqFISH data : properties and challenges

## 1.1 Introduction

SeqFISH [1, 2], and other similar techniques such as osmFISH [3], and MERFISH [4] generate large amount of raw data in the form of high resolution (100x) multi-stacks (20-50) microscopy images. While whole organs are usually not imaged, large regions of several mm2 are typically studied generating up to Terabytes of data. Those data usually consist in .Tiff files that cannot be directly used and have to be pre-processed similarly to the raw .fastq files that have to be heavily processed to generate gene count tables.

While the protocol is supposed to generate clean image with high Signal to Noise Ratio (SNR), we observed in practice significant background signal, as well as non specific binding of the probes making the analysis of such data extremely complex. In the following part of the manuscript we will describe what are the different steps of the pipelines as well as the mathematical foundation on which our pipeline relies on.

## 1.2 Automated processing of seqFISH data is required

The most common use of seqFISH raw data consist in converting them into tables containing the location of each identified cell together with their gene expression level. This requires two key steps : spot detection and cell segmentation. While both tasks can be performed manually for small datasets generated by smFISH technology, this is not feasible for seqFISH data primarily due to the large number of genes studied (up to hundreds in some experiments). Moreover as the data need to be analyzed in a robust and reproducible manner, human intervention need to be as limited as possible. Indeed manual image analysis methods tends to :

- Be poorly reproducible.
- Be tedious and time-consuming while finding an appropriate cut-off values.
- Exhibit high intra- and inter-user variability.

Therefore fully automated seqFISH raw data processing is required.

## 1.3 Current state of the art

To our knowledge no freely available pipeline dedicated to seqFISH has been developed : indeed each paper using seqFISH-like data tend to use a specific script. While those scripts are freely available, they lack clear documentation and robust mathematical foundations that would make them broadly

usable and applicable. In addition, many scripts do not address the issue of efficient and robust cell segmentation as well as non specific probe binding.

On the other hand, efficient tools have been developed for smFISH data such as FISH-quant [5] but are relying on a user defined threshold and are not designed to deal with large datasets. Furthermore, cell segmentation is also performed manually or is only automated in case of in-vitro data. Therefore there is a need for a simple, robust and yet computationally efficient method for seqFISH data pre-processing.

# 2 Detection and counting of RNA molecules

## 2.1 Choice of the method

Spot detection is the first and most important step of seqFISH data processing. As we did not aim to create a new spot detection algorithm but rather identify the most suited approach, we screened recent review[6, 7, 8] and looked for a method with little to no false positive, robust to background noise and with easily understandable parameters. We identified the H-Dome (HD) and the Multiscale methods as the most efficient and adapted method [7, 8], each of them having specific advantages and drawbacks.

The following part of the subsection will be devoted to the description of the mathematical background of the HD and Multiscale method as well their respective software implementation. If not stated otherwise, all functions used are Matlab® functions.

## 2.2 H-dome based spot detection

We selected the HD method because :

1. It has the highest accuracy (F-score) for small object detection.
2. Overall it has the best performance independently of the object shape when the hypothesis made (given size and circular shape) are met.
3. Key parameters of the HD method correspond to precise features of the RNA spots (minimal and maximal size, estimated SNR).

However HD suffers from high computational cost and strong sensitivity to the parameters, therefore requiring careful parameter tuning so that it will perform reasonably across all channels and images.

### 2.2.1 LoG filter

As all spot/small object detection methods, HD method relies on three successive steps : noise reduction, signal enhancement and signal thresholding. In our case, noise reduction and signal enhancement is performed simultaneously by applying Laplacian of Gaussian (LoG) filter on the image [9]. The LoG filter consists in first applying a Gaussian convolution filter G and then computing the Laplacian value at each pixel [9]. Such filter is commonly used in many image analysis cases, such as edge and blob/spot detection and has already been used for RNA spot detection for smFISH data [5].

More formaly, lets $I(x, y)$ be the intensity of the Image of interest at location $(x, y)$. Applying Gaussian kernel consist in computing the convolution product of the Image and and a function $G_\sigma$ :

$$G_\sigma(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

The resulting image $g(x, y)$ is therefore :

$$g(x, y) = G_\sigma(x, y) * I(x, y) \tag{2}$$

where $*$ corresponds to the convolution product. The final image $L(x, y)$ is obtained by applying Laplacian operator :

$$L(x, y) = \nabla^2(G_\sigma(x, y) * I(x, y)) \tag{3}$$

This can be simplified as :

$$\begin{aligned} L(x, y) &= (\nabla^2 G_\sigma(x, y)) * I(x, y) \\ &= LoG_\sigma * I(x, y) \end{aligned} \tag{4}$$

It is therefore enough to convolute the original image with the $LoG_\sigma$ function :

$$LoG_\sigma(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{5}$$

Such function is often called the 'Mexican hat' due to it shape (ref Figure plot Log function) and is implemented using the imfilter and fspecial functions.

In practice we add to small steps :

- Taking the opposite of the transformed image so that the highest values correspond to local maximum and not local minimum.
- Sizing the values of transformed image between 0 and 1 (imadjust function).

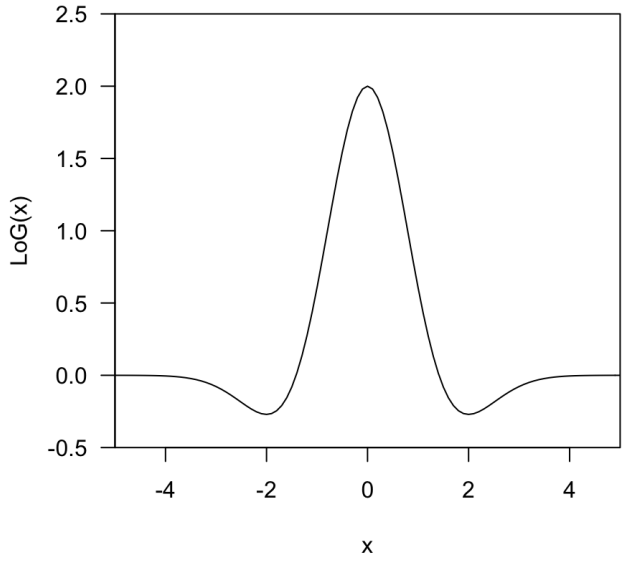


Figure 1: Laplacian of Gaussian (LoG) function value

The choice of the parameter $\sigma$ is essential for the whole spot detection analysis : if it too small the image will not be smoothed enough and noise will be considered as spot while if it is too large spots will be considered as noise and removed. Typically $\sigma$ value will correspond to the radius of a spot (one or two pixels). In the following part of the manuscript this parameter will be name $\sigma_{small}$.

### 2.2.2 H-dome transformation

The next step consist in identifying and extracting the local maxima of a given height. To do so we use a method based on morphological grayscale reconstruction : the h-dome transform. As we do not aim to describe the whole field of morphological grayscale reconstruction we recommend to the reader the field founding paper by Luc Vincent [10].

The h-dome transform aims to identify the so called h-dome structure which are defined as follow. Lets $D$ be a set of connected/neighbor pixels and $h$ a strictly positive real number. Then it is an h-dome if and only if :

- Every pixel $p$ neighbor of $D$ satisfies $I(p) < \min\{I(q)|q \in D\}$, i.e it is a local maximum.
- $\max\{I(q)|q \in D\} - \min\{I(q)|q \in D\} < h$, i.e the difference of intensity between the brightest and dimmest pixel is lower than $h$.

where $I(p)$ corresponds to the pixel intensity of the pixel $p$.

Unlike classical top-hat transformation, the h-dome transformations isolates local maxima without involving any shape size or shape but only height criterion. Interestingly resulting image only

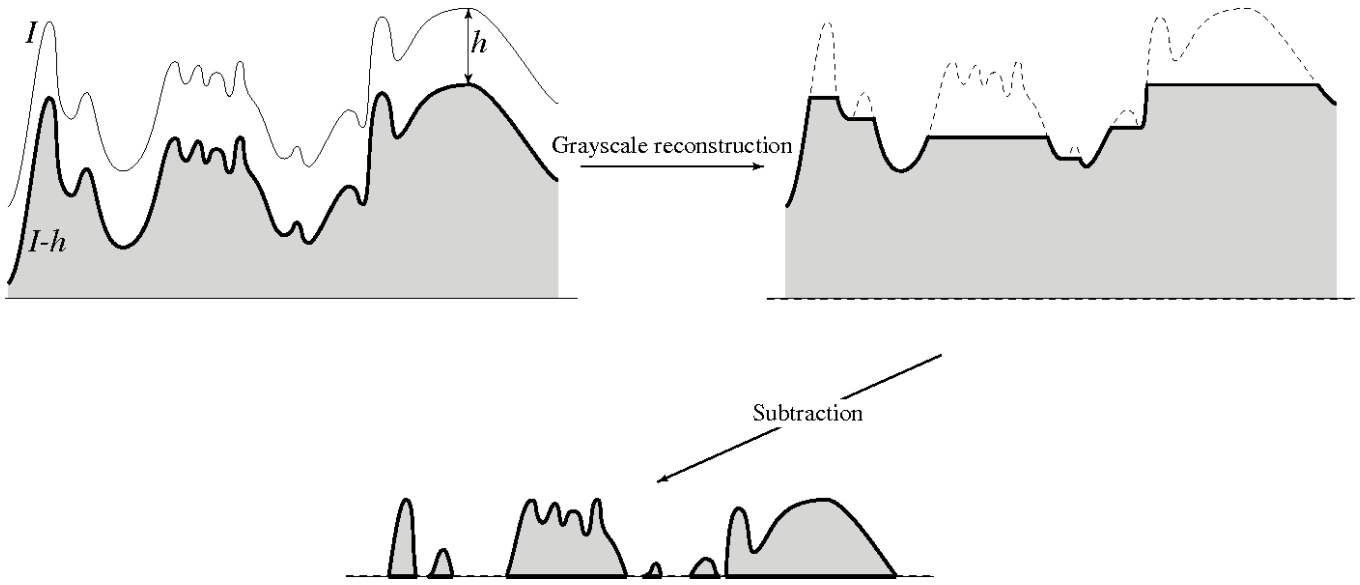


Figure 2: Example of the H-dome transform (from [10]).

contains peaks of maximum height $h$. The h-dome transformed image is generated using the imhmax function and will be called $M$ in the manuscript.

The $h$ parameter need to be carefully chosen as a variation of it can dramatically change the result of the h-dome transform and hence of the whole spot detection analysis. To make the analysis more robust we used the following strategy : for each channel/gene we compute the mean and max value of the LoG transformed image $L(x, y)$ and use them to compute the $h$ parameter :

$$h = Std(L(x, y)) \times h_{Meta} \tag{6}$$

where $h_{Meta}$ is a positive real number and $Std(L(x, y))$ the standard deviation of the $L(x, y)$ image. The $h_{Meta}$ parameter need to be chosen : if the Signal to Noise Ratio (SNR) is expected to be high then $h$ has to be high and $h_{Meta}$ also so that only the tip of the peak is selected while if the SNR is low then $h$ need to high enough to select real peaks.

### 2.2.3 Pixel sampling and aggregation

The filtered image is first transformed by raising the image values to a power $S$ and then by dividing each element by the total image intensity.

$$M_{normalised}(x, y) = \frac{M(x, y)^S}{\sum_{x=1}^{x_{max}} \sum_{y=1}^{y_{max}} M(x, y)^S} \tag{7}$$

where $x_{max}$ and $y_{max}$ correspond to the width and height of the image. Such transformation will favor strong signal and remove low intensity peaks.

It is obvious that :

$$\sum_{x=1}^{x_{max}} \sum_{y=1}^{y_{max}} M_{normalised}(x, y) = 1 \tag{8}$$

Therefore the transformed image can be considered as a probability vector and we can sample pixels according to it. We therefore sample $N_{Sampled\,pixels}$ pixels according to this probability .

The last step consist in aggregating the sampled pixels into spots. To do so the Mean-Shift (MS) algorithm is used. Briefly the MS algorithm identifies clusters as high density regions through an efficient hill-climbing algorithm based on kernel density estimator. In opposition to many other clustering techniques, the number of clusters does not need to be provided, making it particularly suited for our use case as we ignore the number of spots. For more details we recommend the reader to look at the exhaustive and excellent review from Carreira-Perpinan [11].

While efficient and well-suited , the MS algorithm suffers from two main issues :

1. It relies on a key parameter $w$ that controls the width of the kernel window used for density estimation.

2. It requires large CPU and memory for big datasets (the kernel density estimator uses all the points at each iteration in its naive implementation)

While the first problem can be easily solved as we expect the spot to have a precise size (empirically we used $w = 2\sigma_{small}$) the second problem requires to implement a modified version of the algorithm where at each iteration only the neighboring points that are close enough are used for the mean-shift computation. To do so, only points for which the corresponding weight after applying gaussian kernel is bigger than 0.001 are used. As this correspond to a Fixed-radius near neighbors finding, this can be efficiently performed through the use of K-D Tree and hence considerably reduces computational time.

At the end of the MS algorithm we got a list of cluster centroids/positions as well as the pixels associated with it.

### 2.2.4 Pixel clusters filtering

The MS algorithm can produce spurious pixel clusters that do not correspond to real spots. Those pseudo-spots will tend to have few pixels or to be highly spread. Such criteria can be used to remove them. Let $C_i$ be the set of pixels associated with cluster i and $N_{Clusters}$ be the total number of clusters. Then :

$$\forall i \in [[1, N_{Clusters}]] \quad C_i = \{P_j | Cluster(P_j) = i\} \tag{9}$$

where $P_j$ corresponds to the (x,y) position of pixel $j$.

We can them compute the number of pixels $Nc_i$ and the dispersion $Disp_i$ of each pixel cluster $C_i$:

$$Nc_i = |C_i| \tag{10}$$

$$Disp_i = det(cov(C_i)) \tag{11}$$

where $cov(C_i)$ corresponds to the empirical covariance matrix of pixel location belonging to cluster $i$.

We only kept clusters $i$ such that $Nc_i > 5$ and $Disp_i < \sigma_{Max}^4$ where $\sigma_{Max}$ correspond to maximal size of the spot. This last filtering can be interpreted as follow : the size of a spot will be proportional to the determinant of the co-variance spot matrix. Therefore in the case of a purely spherical spot of characteristic size $\sigma$, the determinant of the matrix is therefore equal to $\sigma^2 \times \sigma^2 + 0 \times 0 = \sigma^4$. The pixel clusters that pass those filtering are considered as real spots and are then kept for further analysis.

### 2.2.5 Parameters of the HD approach

The HD relies on many different parameters :

- The minimal spot size ($\sigma_{small}$)
- The maximal spot size ($\sigma_{max}$)
- The height of the extracted h-domes ($h_{Meta}$)
- The power to which the H-dome matrix is raised ($S$)
- The number of pixels sampled ($N_{Sampled\,pixels}$)

While the HD method is highly efficient, it requires a fine tuning to be efficient in the case of low SNR. While $\sigma_{small}$ and $\sigma_{max}$ are easy to interpret and to tune, the $h_{Meta}$ and $S$ are way harsher to deal with, as both parameters strongly interact together. Lastly, the $N_{Sampled\,pixels}$ parameter need to high enough to catch all the spots. However, in the case of highly genes (thousands of spots per image), we observed that the maximal number of pixels sampled (more than hundred thousands) can saturate the working memory, making the HD method not usable in that case.

## 2.3 Multiscale spot detection

While in case of simulated data, HD method performs best, it may suffer from a significant loss of accuracy in case of real data. Conversely, the Multiscale method exhibits highly robust performance and is easy to tune with lower performance in the case of low quality signal. We present here a modification of the original method published in [12]. If the reader is interested by a more

comprehensive description of mathematical background on which the Multiscale approach is founded on, we recommend to read [13].

### 2.3.1 Multiscale Hessian matrix determinant computation

The first step of the Multiscale method consists in smoothing the original image with gaussian kernels of various size. Let $\sigma_0$ be the smallest $\sigma$ gaussian kernel parameter used and $N$ the number of different gaussian kernels used, we defined $\theta = \{\sigma_i\})_{k=0}^{N-1}$ where $\sigma_k = 2^k\sigma_0$. For each different $\sigma_k$ we can therefore compute :

$$g_k(x, y) = G_{\sigma_k} * I(x, y) \tag{12}$$

For each voxel of the smoothed image $g_k$, the Hessian matrix is then computed :

$$H_k(x_0, y_0) = \begin{bmatrix} \frac{\partial^2 g_k}{\partial x^2} & \frac{\partial^2 g_k}{\partial x \partial y} \\ \frac{\partial^2 g_k}{\partial y \partial x} & \frac{\partial^2 g_k}{\partial y^2} \end{bmatrix}_{(x_0,y_0)} \tag{13}$$

In practice, the Hessian matrix components are computed using the Sobel operator. The determinant of the Hessian matrix is then used to estimate the total signal variation. Therefore a new image can be computed for each $sigma_k$ value :

$$V_k(x, y) = det(H_k(x, y)) \tag{14}$$

Finally by taking for each pixel the maximal value across all $V_k$ images, we obtained a new image $M$ that highlights local variations at several scales, making the spot detection step more robust.

$$M(x, y) = \max_k(V_k(x, y)) \tag{15}$$

### 2.3.2 Automated thresholding and spot extraction

The obtained $M$ matrix is then automatically thresholded using the triangle's method. Briefly the triangle method consists in selecting the point in the image intensity histogram that maximize the distance with the diagonal line of the height and range and then adding a fixed offset $T_{offset}$ [14].

The connected components of the thresholded images are then extracted and their centroid used as spot location. Connected components with more than 5 pixels are kept and considered as spots.

### 2.3.3 Parameters of the Multiscale approach

In conclusion, the Multiscale approach relies on three different parameters : $\sigma_0$ (minimal smoothing parameter), $N$ (number of scales) and $T_{offset}$ (offset to threshold used in the binarisation step). The

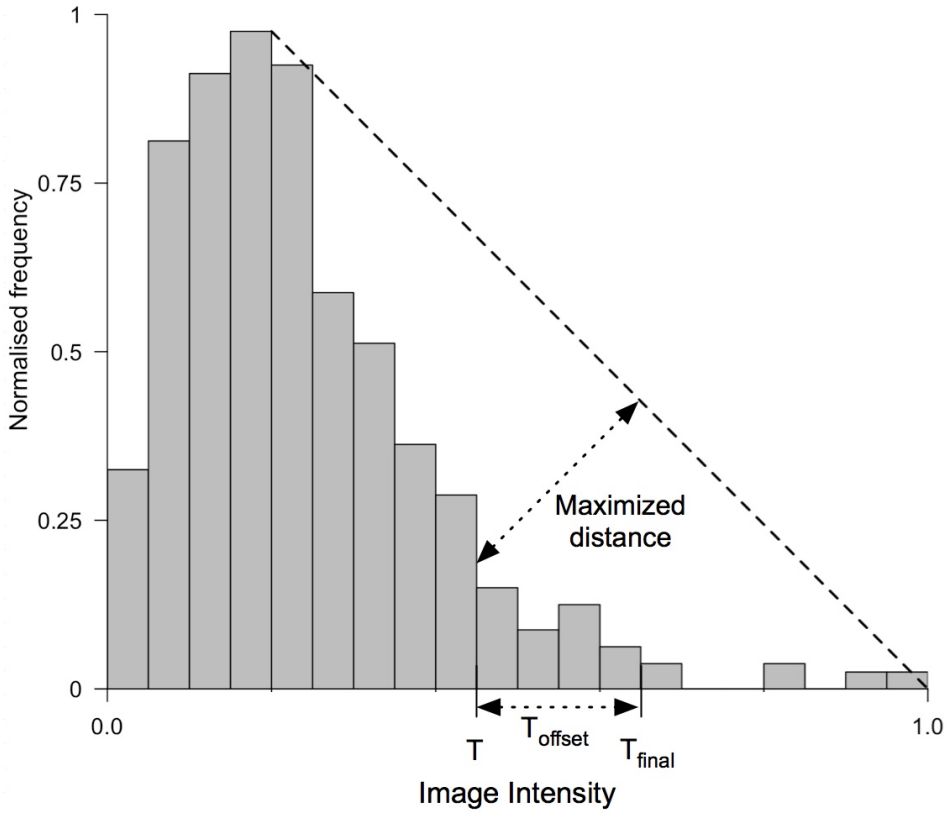


Figure 3: Description of the triangle binarization method

first two parameters are easy to tune as $\sigma_0$ can be set to the minimal size of a spot (at least one pixel) and and $N$ should simply be above a given value to avoid missing larger objects. We however recommend not to use a too high value of $N$ that would significantly increases computation time. Lastly the $T_{offset}$ parameter can be used to fine tune the thresholding step but is by default set to 0 to avoid over-fitting and increase robustness of the analysis.

## 2.4 Removing non specific spots

While the HD and Multiscale method are extremely efficient at identifying spots, they can still be fooled by a specific type of technical artifact : the non-specific binding of probes and of HCR amplifiers to sub-cellular structures. This can generate false positive signal and artificial spots that have no biological meanings. To remove such artifacts we used the following empirical observations :

- Those spots are dimmer than the real RNA spots.
- They are spread uniformly across the tissue while the real spots are highly clustered.

We therefore used a spatial clustering based approach :

1. First the image intensity at each spot location is extracted.

2. The local spot density is computed using the density.ppp function from the R package spatstat.

3. For 30 different intensity threshold, the spots with an intensity below the value are removed and a Kulldorf's spatial scan test is computed [15]. The corresponding Likelyhood Ratio ($LR$) is then extracted.

4. The intensity with the highest $LR$ is selected and all spots with an intensity below this threshold are considered as noise and removed.

The Kulldorf's spatial test consists in checking if there is at least one zone of the image where the spot density is not coherent with a background spatial distribution, in our case an non-homogeneous Poisson Point Process (PPP) derived from the initial spot distribution. More details can be found in Kulldorf's paper [15].

### 2.5 Quality Control of the spots

Now that the points have been identified and filtered, the Quality Control of the spot analysis can be performed. To provide as much information as possible to the user the following features are displayed for each position and each channel/gene :

- The number of spots identified before and after filtering.
- The ratio of spots that pass or not the filtering.
- The mean Signal to Noise Ratio (SNR) of the spot.

For each spot $i$, its SNR ($SNR_i$) is computed by extracting the neighbor pixel intensity (a square box of seven pixel width and height is used) and estimating the corresponding standard variation $\sigma_i$. If the itensity of the spot is $\mu_i$, then its SNR is simply :

$$SNR_i = \frac{\mu_i}{\sigma_i} \tag{16}$$

We consider that below a mean SNR of 2, a channel should not be used and that the identified spots are mostly noise. Similarly, a mean SNR above 5 correspond to a high quality channel and can be used without any risk for further analysis.

## 3 Cell segmentation

### 3.1 Limitations of current cell segmentation methods

Cell segmentation is an essential step for automated microscopy image processing that allows to quantify a given measurement/feature (RNA or protein expression) at a single-cell resolution. While

fully automated cell segmentation is easily performed on in-vitro cell images, doing it on in-vivo cell images remain challenging. Numerous experimental and computational strategies have already been developed but all of present significant drawbacks that limit their use in seqFISH technology :

- DAPI/nuclear based techniques can not be used if the cells are not rounds or if the tissue is too crowded such as in lymphoid tissues.
- Membrane staining through antibody or Wheat Germ Agglutinin (WGA) is not compatible with some seqFISH protocols that use SDS for tissue cleaning.
- Cytoplasm staining, such as the Nissl staining, has already been used with seqFISH [1] but requires manual analysis due to high cell crowding.
- Background signal based segmentation, a powerful strategy already used for in-vitro smFISH image analysis can not be used on tissues samples as the background can be generated by non cellular structures [5].

There is therefore a need for an automated cell segmentation method that could be apply on any seqFISH dataset, independently of the imaged tissue structure and underlying cells shape.

## 3.2 RNA spots based segmentation

### 3.2.1 Principles

Our segmentation method is based on the following empiric observation : genes that are highly expressed but only in a limited set of cells can be used ton infer the shape the cells. Indeed if a gene is highly expressed by a cell, its RNA molecules will usually spread across its whole cytoplasm, revealing the cell shape. Moreover, if only a minor fraction of the cells express that gene, the probability that two touching cells express that gene is low and therefore the observed structures will not originate from two neighbor cells. We thus developed an automated cell segmentation method based on this observation. This method relies on the following steps :

1. For each gene, the spots are clustered using a spectral based approach, inspired from the improved NJW clustering algorithm and from the Diffusion Map dimensionality reduction method[16, 17, 18].
2. The obtained clusters are filtered out to remove spurious clusters. Clusters obtained across all genes are then compared and overlapping clusters are aggregated
3. For each cluster aggregate, the associated spots are used to compute the final shape of the cell.

### 3.2.2 Spectral clustering of the RNA spots

To cluster the RNA spots in a shape and cluster number agnostic way, we decided to use a method inspired from the NJW clustering approach [16, 17] and from the diffusion map dimensionality reduction tool. Both methods are highly linked and an extensive comparison can be read in [18].

Our clustering approach starts by computing the similarity between all $n$ RNA spots of a given gene at a given position. Let $x_i$ and $x_j$ be the location of points $i$ and $j$ respectively. We define the similarity $s(i,j)$ between those two points as :

$$s(i,j) = \exp(\frac{-d(x_i,x_j)^2}{2\sigma_i\sigma_j}) \tag{17}$$

where $d(x_i,x_j)$ corresponds to the euclidean distance between points $i$ and $j$, while $\sigma_i$ and $\sigma_j$ correspond to the distance of point $i$ (and $j$ respectively) to their $K$ nearest neighbor. Typically $K$ value will be around 10/20. Such approach avoid to manually define a global $\sigma$ value and to deal with varying point density in the sample space.

The similarity matrix $S$, defined by $S_{i,j} = s(i,j)$, is then normalized to produce a stochastic matrix :

$$M = D^{-1}S \tag{18}$$

where $D$ is a diagonal normalization matrix with $D_{i,j} = \sum_j S_{ij}$ . We now can utilize the fact that by construction M is stochastic matrix with rows summing to one and can be thus interpreted as a transition matrix and defines a random walk. Under this view $M_{i,j}$ defines the transition probability between points $i$ an $j$ in one step.

We can therefore define $p(t,j|i)$ as the probability distribution of a random walk landing at point $j$ at time $t$ with starting point $i$. A diffusion distance distance can then be defined for a given $t$ :

$$D_t^2(i,j) = \sum_k (p(t,k|i) - p(t,k|j))^2 w(k) \tag{19}$$

where $w(k) = \frac{\sum_i D_{ii}}{D_{kk}}$ corresponds to the inverse of the empirical local density of the points. This distance can be heuristically understood as an exhaustive comparison of random walks starting at point $i$ and $j$ ability to reach points in the graph. Such distance is especially useful in our case to perform clustering : indeed performing clustering using 'classical' euclidean distance usually results in poor results as the euclidean distance between two neighboring cells can frequently be smaller than the distance between spots inside a given cell.

As $M$ is adjoint to a symmetric matrix, it is diagonizable and has a set of $n$ ordered real eigenvalues $\{\lambda_i\}_{j=1}^n$ with corresponding eigenvector $\{\psi_i\}_{j=1}^n$ . Using the $k$ first eigenvectors, a new

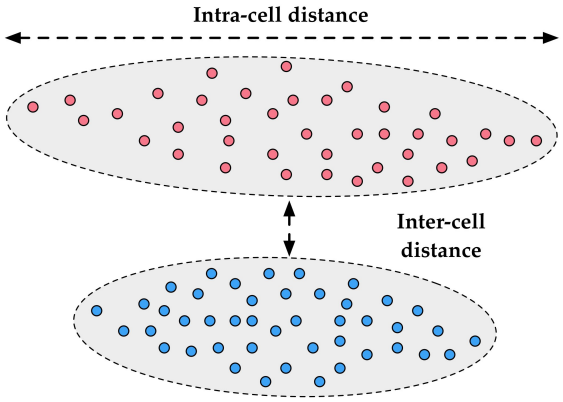


Figure 4: Example of the inefficiency of euclidean distance.

space can be defined for each point $i$, called the diffusion map at a given time $t$, :

$$\Psi_t(i) = (\lambda_2^t\psi_2(i), \lambda_3^t\psi_3(i), .., \lambda_k^t\psi_k(i)); \tag{20}$$

It is possible to show that the euclidean distance in this space is a correct approximation of the diffusion distance in the initial space [18]. We can therefore compute the location of the points in the diffusion map space and perform clustering in it using a simple k-means algorithm similarly to the NJW algorithm [16, 17]. In practice the choice of the number of eigenvalues used ($k$ parameter) as well as the number of clusters is made by looking for the spectral gap, i.e a significant decrease in the eigenvalues. We therefore define $k$ as being :

$$k = \underset{i}{\text{argmax}}(\{\frac{\lambda_{i+1}}{\lambda_i}\}_{i=2}^{N_{comp}}) \tag{21}$$

Here $N_{comp}$ corresponds to the maximal number of eigenvalues computed, which is equivalent to the maximal number of cells expected.

The resulting clusters are then filtered using two different criteria :

1. The number of spots belonging to the cluster : clusters with less than $N_{\text{Min spots}}$ are removed. $N_{\text{Min spots}}$ is usually set to 15 to get significant clusters.

2. The spot density of the clustering : for each cluster the two-dimensional convex hull area $A$ is computed. If the global point density is equal to $\lambda$, then by random the number of points in the convex hull will follow a Poisson law of parameter $\lambda A$. We only select clusters where the probability of getting the observed number of spots according to the Poisson model is lower than 0.0001.

Our clustering procedure therefore results in a list of high quality clusters for each gene at each position.

### 3.2.3 Aggregation of spot clusters

The obtained clusters are then gathered across the different measured genes and their potential overlap computed. To do so, a two-dimensional histogram density estimate of each cluster points is performed with .... x ... grid matrix. For each pair of clusters $i$ and $j$ with associated density estimates $p_i(x, y)$ and $p_j(x, y)$, we compute the Bhattacharyya similarity coefficient :

$$D_B(p_i, p_j) = \sum_x \sum_y \sqrt{p_i(x, y) p_j(x, y)} \tag{22}$$

As this coefficient is bounded between 0 (no overlap) and 1 (identical distributions), it is easy to define a global similarity threshold $\alpha$, above which two spot clusters are considered as similar. We therefore defined a squared matrix $G$, of size $|N_{cluster} N_{cluster}|$ where :

$$G_{ij} = \begin{cases} 1 & \text{if } D_B(p_i, p_j) \geq \alpha \\ 0 & \text{if } D_B(p_i, p_j) < \alpha \end{cases} \tag{23}$$

A graph is then constructed using $G$ as an adjacency matrix. The connected components, of the graph are then extracted and are considered as corresponding to a unique cell. While the choice of the $\alpha$ parameter is left to the user, we observed in practice that a threshold value of 0.3 was stringent enough to avoid the merge of spot clusters coming from different cells.

### 3.2.4 Final cell shape estimation

The last step of our method consists in estimating the shape of the cell. To do so, for each cluster aggregate previously identified, we collect all the RNA spots across all genes. The two-dimensional density distribution of those spots is then estimated using a gaussian kernel density estimator with a bandwidth parameter $\sigma_{Spread}$. The resulting density estimate matrix is then binarized using Otsu's method and the biggest connected component is considered as the final shape of the cell.

### 3.2.5 Removing cell overlaps

While our segmentation strategy produces a good quality cell segmentation, it may generates overlapping cell areas. To remove those overlaps, we developed a heuristic algorithm that starts from a list of cell areas which partially overlap and returns a list of corrected cell areas that do not overlap. Let be $L_{\text{overlap}}$ the list of overlaps, i.e the list of pairs of cells sharing at least one pixel, $N_{\text{overlaps}}$ the number of overlaps, $N_{cells}$ the number of cells, and for each cell $i$, $C_i$ the list of pixels within its

borders. Then we apply the following algorithm :

**Algorithm 1:** Cell overlap cleaning algorithm

**Input :** List of overlapping cell areas $\{C_i\}_{i=1}^{N_{cells}}$

**Output:** List of updated overlapping cell areas $\{C_i\}_{i=1}^{N_{cells}}$

**for** $k = 1 \rightarrow N_{overlaps}$ **do**
  $i = L_k(1)$
  $j = L_k(2)$
  **if** $C_i \cap C_j \neq \emptyset$ **then**
    $D_i = \frac{\text{Dist}_{\text{transform}}(C_i)}{|C_i|}$
    $D_j = \frac{\text{Dist}_{\text{transform}}(C_j)}{|C_j|}$
    $I = C_i \cap C_j$
    $N_{\text{shared pixels}} = |I|$
    **for** $l = 1 \rightarrow N_{shared\ pixels}$ **do**
      $s = \text{argmin}(D_i(l), D_j(l))$
      $C_s = C_s - \{I(l)\}$
    **end**
  **else**
  **end**
**end**

# 4 Image alignment and stitching

## 4.1 Requirement of Image alignment and stitching for seqFISH data processing and analysis

During seqFISH data generation, the same positions are imaged several times to increase the number of observed genes in a linear (serial hybridization) or exponential manner (barcoding hybridization) [4]. While in theory the images coming from the same position should be perfectly aligned, a significant drift can be observed primarily due to temperature variations and external vibrations. It is therefore necessary to computationally re-align the pictures before running any analysis, especially in the case of barcoding hybridization where the same RNA spot has to be identified across the different rounds to perform demultiplexing.

A similar procedure has to be applied to perform image stitching, that is to say to combine multiple images with overlapping fields of view to produce a panoramic picture. Indeed it is possible to automatically image a whole tissue section by generating partially overlapping images (usually

around 10 to 20 % of overlap). While the theoretical overlap between pictures is known, it can vary in practice and need to be precisely estimated in an automated manner.

Both problems are linked and are part of what is known as Image Registration. The following subsections will describe the computational method used to solve both problems as well as its implementation and practical details.

## 4.2 Image registration using phase correlation method

Over the last decades several methods have been developed to align or stitch related images. As we do not aim to provide an exhaustive review of the field, we recommend the interested readers to have a look at the excellent review by Szeliski [19].

We decided to use the Fourrier transform based Phase correlation method, a method commonly used for microscopy image stitching [19, 20] due to its high efficiency and low computational cost thanks to Fast Fourrier Transform (FFT). While this image registration method is only able to deal with translation-based alignment, we considered that rotation and scaling transformation where not likely to happen in our experimental setting.

Let $g_A$ and $g_B$ be two different images of same size $(N \times M)$. Let $G_a = F(g_A)$ and $G_b = F(g_B)$ their respective discrete Fourrier transform signal. We can compute the cross-power spectrum R matrix :

$$R = \frac{G_a \odot \overline{G_b}}{|G_a \odot \overline{G_b}|} \tag{24}$$

where $\odot$ corresponds to the Hadamard product (element wise matrix multiplication) and $\overline{G_b}$ the complex conjugate of $G_b$. The inverse Fourrier transform of R can then be obtained and used to identify the translation vector $v$ :

$$r = F^{-1}(R) \tag{25}$$

$$(v_x, v_y) = \text{argmax}(r) \tag{26}$$

This technique is motivated by the shifting property of Fourrier transform. Indeed if image $g_B$ is generated from perfect (cyclic) shift of image $g_A$, i.e $g_B(x, y) = g_A(x + v_x, y + v_y)$, then :

$$G_b(w, z) = e^{-2\pi j(\frac{wv_x}{N} + \frac{zv_y}{M})} G_a(w, z) \tag{27}$$

We can then simplify equation 24.

$$R(w, z) = e^{-2\pi j(\frac{wv_x}{N} + \frac{zv_y}{M})} \tag{28}$$

The inverse Fourrier transform of R will therefore result in a highly localized peak corresponding to the shift vector.

In practice, to increase the performance of the phase correlation method, we applied a filter to increase the signal from the borders in the case of image stitching . This two dimensional filter, $H$, correspond to one minus the Hamming filter :

$$H(x,y) = (1 - \frac{1}{2}(1 - cos(2\pi\frac{x}{N_x})))(1 - \frac{1}{2}(1 - cos(2\pi\frac{y}{N_y}))) \tag{29}$$

where $N_x$ and $N_y$ to the $x$ and $y$ size values.

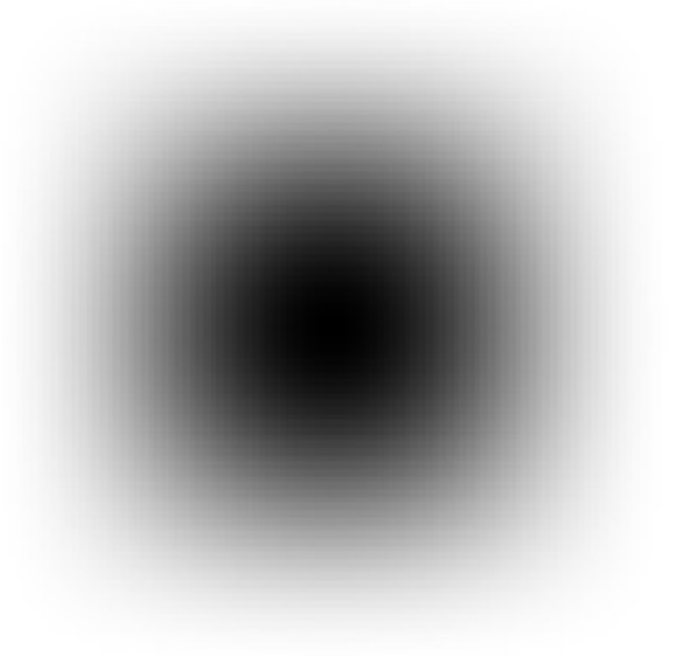

Figure 5: Inverse two-dimensional Hanning filter function.

Conversely, in the case of image alignment, a classical Hanning filter is used.

## 4.3 Computation of the cumulative stitching vectors

Stitching more than two images together is a challenging task, as independent pairwise transforms may not agree when the image share several neighbors. To overcome this limitation we developed a graph based method that blindly identifies and favors robust stitching vectors.

Lets consider a set of $N_{\text{Image}}$ overlapping images. We assume that each image overlaps with at least one other image. First, the phase correlation method is applied for each possible pair of non identical images. To quantify the likelihood that the two images, $i$ and $j$, are indeed overlapping, the ratio between the maximal value of the phase correlation matrix $r_{ij}$ (correlation peak height) and the standard variation of $r_{ij}$ values is computed :

$$S_{ij} = \begin{cases} \frac{\max(r_{i,j})}{\sigma(r_{i,j})} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \tag{30}$$

We therefore end up with a symmetric matrix of size $|N_{\text{Image}}|$. This matrix value must then be thresholded to identify significant and non significant stitching vectors. To do so we applied Otsu's thresholding on the non null elements of the matrix to obtain a threshold $T$. A thresholded version of $S$, $S^{\text{threshold}}$ is then computed :

$$S_{ij}^{\text{Threshold}} = \begin{cases} 1 & \text{if } S_{ij} \geq T \\ 0 & \text{if } S_{ij} < T \end{cases} \tag{31}$$

An undirected graph $G$ is then build using $S^{\text{Threshold}}$ as adjacency matrix.

While automated thresholding is usually performing well, we established a list of criterion to control for the value of T based on the obtained graph $G$. Indeed, if the graph is interpreted as a representation of the overlaps between pictures (nodes correspond to images and edges to significant overlap), then the graph should respect the following rules :

- Only one connected component of size strictly bigger than one should be observed in $G$.
- There must be at least $N_{\text{Image}} - 1$ edges in $G$.
- If we consider that the images are spread on a regular grid then the maximal number of edges in $G$ is equal to $2(N_{\text{Image}} - \sqrt{N_{\text{Image}}})$.

Therefore an upper ($T_{max}$) and lower bounds ($T_{min}$) of $T$ can computed with $T_{max}$ being equal to the $N_{\text{Image}} - 1$ highest value of $S$ and $T_{min}$ to the $2(N_{\text{Image}} - \sqrt{N_{\text{Image}}})$ highest value of $S$. Thus, even before $G$ is computed, $T$ is updated :

$$T_{\text{updated}} = \begin{cases} T_{max} & \text{if } T \geq T_{max} \\ T_{min} & \text{if } T \leq T_{min} \\ T & \text{if } T_{min} < T < T_{max} \end{cases} \tag{32}$$

The graph $G$ is then computed and evaluated : if only one connected component with strictly more than one node is observed then the graph is used as is. However if multiple connected components are observed then the threshold value is progressively lowered and the graph build until a unique connected component is observed. If no connected component is observed while T has reached $T_{min}$, then the stitching procedure is stopped and an error message provided.

Once the graph is build, a reference image is chosen and the shortest graph path between each node and the node corresponding to the reference image are computed. These paths are then used

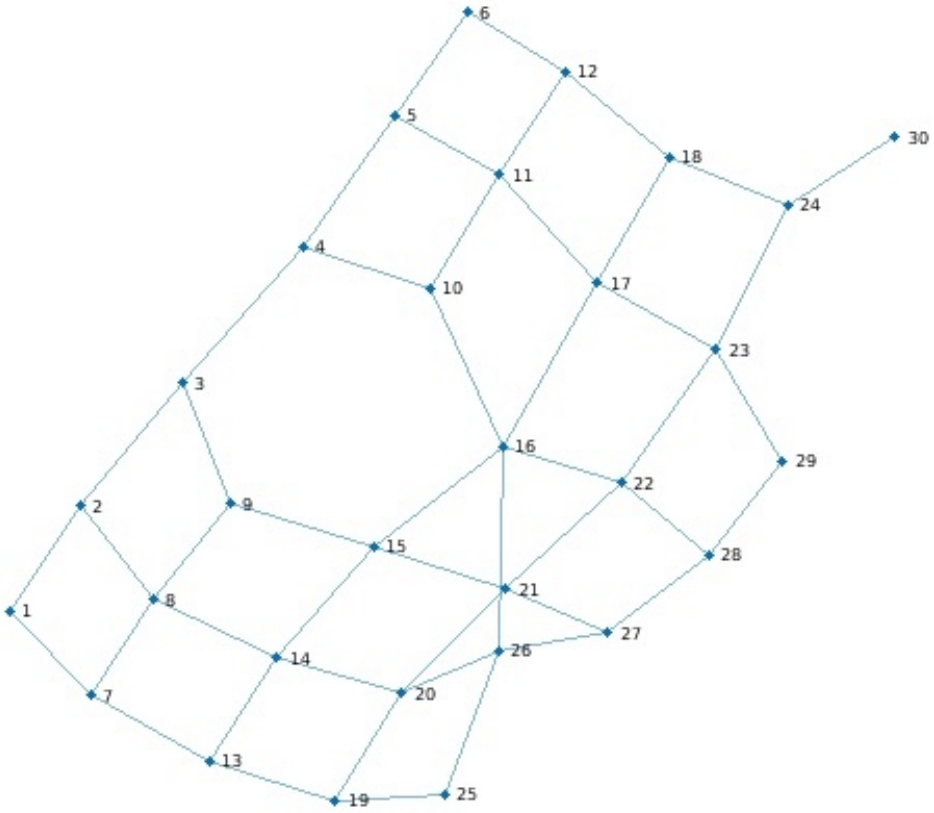


Figure 6: Example of a stitching graph from 30 images placed on a rectangular grid

to compute the final stitching vectors. Let $p_i$ be a path between the node $i$ and the reference node, that is to say a collection of edges allowing to go from node $i$ to the reference node, then the final stitching vector for image $i$ is equal to the sum of each stitching vectors corresponding to the edges in that path:

# References

[1] Sheel Shah, Eric Lubeck, Wen Zhou, and Long Cai. In Situ Transcription Profiling of Single Cells Reveals Spatial Organization of Cells in the Mouse Hippocampus. *Neuron*, 92(2):342–357, October 2016.

[2] Chee-Huat Linus Eng, Michael Lawson, Qian Zhu, Ruben Dries, Noushin Koulena, Yodai Takei, Jina Yun, Christopher Cronin, Christoph Karp, Guo-Cheng Yuan, and Long Cai. Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH+. *Nature*, 568(7751):235–239, April 2019.

[3] Simone Codeluppi, Lars E. Borm, Amit Zeisel, Gioele La Manno, Josina A. van Lunteren,

Camilla I. Svensson, and Sten Linnarsson. Spatial organization of the somatosensory cortex revealed by osmFISH. *Nature Methods*, 15(11):932–935, November 2018.

[4] Kok Hao Chen, Alistair N. Boettiger, Jeffrey R. Moffitt, Siyuan Wang, and Xiaowei Zhuang. Spatially resolved, highly multiplexed RNA profiling in single cells. *Science*, 348(6233), April 2015.

[5] Florian Mueller, Adrien Senecal, Katjana Tantale, Hervé Marie-Nelly, Nathalie Ly, Olivier Collin, Eugenia Basyuk, Edouard Bertrand, Xavier Darzacq, and Christophe Zimmer. FISH-quant: automatic counting of transcripts in 3d FISH images. *Nature Methods*, 10(4):277–278, April 2013.

[6] Pekka Ruusuvuori, Tarmo Äijö, Sharif Chowdhury, Cecilia Garmendia-Torres, Jyrki Selinummi, Mirko Birbaumer, Aimée M. Dudley, Lucas Pelkmans, and Olli Yli-Harja. Evaluation of methods for detection of fluorescence labeled subcellular objects in microscope images. *BMC Bioinformatics*, 11(1):248, May 2010.

[7] Ihor Smal, Marco Loog, Wiro Niessen, and Erik Meijering. Quantitative Comparison of Spot Detection Methods in Fluorescence Microscopy. *IEEE Transactions on Medical Imaging*, 29(2):282–301, February 2010.

[8] Karel Štěpka, Pavel Matula, Petr Matula, Stefan Wörz, Karl Rohr, and Michal Kozubek. Performance and sensitivity evaluation of 3d spot detection methods in confocal microscopy. *Cytometry. Part A: The Journal of the International Society for Analytical Cytology*, 87(8):759–772, August 2015.

[9] Haldo Spontón and Juan Cardelino. A Review of Classic Edge Detectors. *Image Processing On Line*, 5:90–123, June 2015.

[10] L. Vincent. Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. *IEEE Transactions on Image Processing*, 2(2):176–201, April 1993.

[11] Miguel Á Carreira-Perpiñán. A review of mean-shift algorithms for clustering. *arXiv:1503.00687 [cs, stat]*, March 2015. arXiv: 1503.00687.

[12] Veronika Foltánková, Pavel Matula, Dmitry Sorokin, Stanislav Kozubek, and Eva Bártová. Hybrid Detectors Improved Time-Lapse Confocal Microscopy of PML and 53bp1 Nuclear Body Colocalization in DNA Lesions. *Microscopy and Microanalysis*, 19(2):360–369, April 2013.

[13] Jiaoying Jin, Linjun Yang, Xuming Zhang, and Mingyue Ding. Vascular Tree Segmentation in Medical Images Using Hessian-Based Multiscale Filtering and Level Set Method, 2013.

[14] G. W. Zack, W. E. Rogers, and S. A. Latt. Automatic measurement of sister chromatid exchange frequency. *The Journal of Histochemistry and Cytochemistry: Official Journal of the Histochemistry Society*, 25(7):741–753, July 1977.

[15] Martin Kulldorff. A spatial scan statistic. *Communications in Statistics - Theory and Methods*, 26(6):1481–1496, January 1997.

[16] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On Spectral Clustering: Analysis and an Algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pages 849–856, Cambridge, MA, USA, 2001. MIT Press. event-place: Vancouver, British Columbia, Canada.

[17] Lihi Zelnik-Manor and Pietro Perona. Self-tuning Spectral Clustering. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, pages 1601–1608, Cambridge, MA, USA, 2004. MIT Press. event-place: Vancouver, British Columbia, Canada.

[18] Boaz Nadler, Stéphane Lafon, Ronald R. Coifman, and Ioannis G. Kevrekidis. Diffusion Maps, Spectral Clustering and Eigenfunctions of Fokker-Planck Operators. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, NIPS'05, pages 955–962, Cambridge, MA, USA, 2005. MIT Press. event-place: Vancouver, British Columbia, Canada.

[19] Richard Szeliski. Image Alignment and Stitching: A Tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006.

[20] Stephan Preibisch, Stephan Saalfeld, and Pavel Tomancak. Globally optimal stitching of tiled 3d microscopic image acquisitions. *Bioinformatics*, 25(11):1463–1465, June 2009.