

# RNA Imaging with MERFISH - Data Analysis

Jeffrey R. Moffitt, Xiaowei Zhuang

## Abstract

The basic decoding of MERFISH data—the identification of specific RNA species for each spot in the sample—consists of three basic steps. First, fluorescent spots must be identified in all images. Second, slight changes in the stage position between images of the same region from different hybridization rounds must be corrected. Finally, sets of fluorescent spots that occupy the same location in the sample must be decoded into binary barcodes, which, in turn, are decoded into the specific RNA species. After these RNAs have been decoded, there are a series of simple calculations that can often be performed on the measured barcodes to determine important parameters associated with the performance of the measurements, such as estimates of the binary errors made at each bit or the relative confidence that can be ascribed to the counts for each RNA species. In this section, we provide protocols for performing these basic computational tasks. Example software and data can be found at:

<http://zhuang.harvard.edu/merfish/>.

**Citation:** Jeffrey R. Moffitt, Xiaowei Zhuang RNA Imaging with MERFISH - Data Analysis. **protocols.io**  
dx.doi.org/10.17504/protocols.io.metc3en

**Published:** 29 Mar 2018

## Guidelines

### Identification of Fluorescent Spots

There are a variety of software packages and approaches to identifying the location and brightness of fluorescent spots in images. In principle, any of these approaches should work reasonably well for this initial stage of the analysis of MERFISH data. However, some of the more advanced algorithms for the identification of fluorescent spots have been developed in the context of localization-based super-resolution microscopy. In these measurements, there is a desire to be able to properly localize molecules that are close enough together that their fluorescent spots partially overlap. Depending on the local density of RNAs within the cell, the signal from different RNAs does occasionally overlap, necessitating the use of these more advanced spot finders. The algorithm we use—3D-daoSTORM—was developed in our laboratory (Babcock et al., 2012) and is an extension of the daoPHOT algorithm originally developed for the analysis of astronomy data (Holden et al., 2011). 3D-daoSTORM is open source and can be found with documentation here:

<https://github.com/ZhuangLab/stormanalysis>.

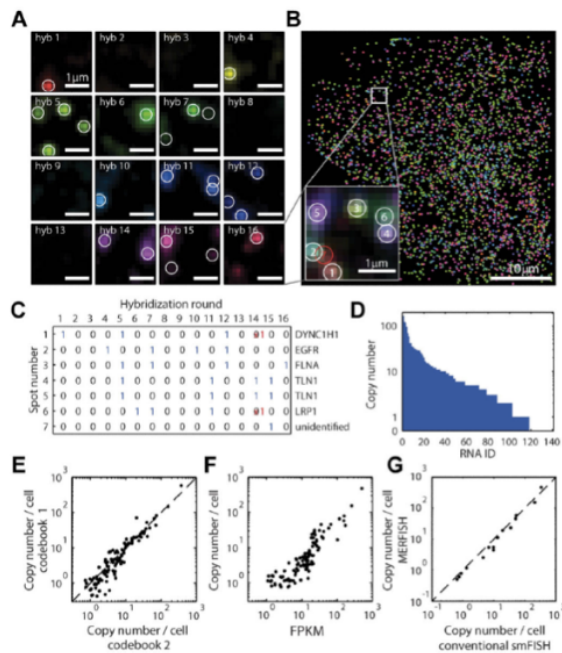
In this Section, we describe the basic steps for the use of this software for finding fluorescent spots in MERFISH data.

## Correction of Image Offsets

Every time the microscopy stage returns to a given location there will be a slight offset, often no more than a single pixel or two. Nonetheless, this small offset can significantly degrade the ability to align fluorescent spots within images of the same sample stained with different readout probes. For this reason, in each round of hybridization, we collect an image of small fluorescent beads stuck to the surface of the sample. Because the location of these beads are fixed with respect to the sample, apparent differences between the position of these beads between different rounds of hybridization can be used to correct these small stage offsets. We perform this correction by using the positions of these beads to create affine transformations that map the position of each spot in each round of hybridization back to the coordinate system defined by the image collected in the first round of hybridization.

## Decoding Barcodes

Once all spots have been transformed to the same coordinate system, spots in different rounds of hybridization that occupy the same or similar physical locations within the sample should be associated to create the specific binary barcode associated with the putative RNA at that location in the cell. We have explored several different approaches to performing this association. Here we present the algorithm that has performed best in our hands. One advantage of this algorithm is that it requires only one parameter: the maximum distance between spots in different rounds of hybridization. Figure 3A-C illustrates the decoding of an example MERFISH data set.

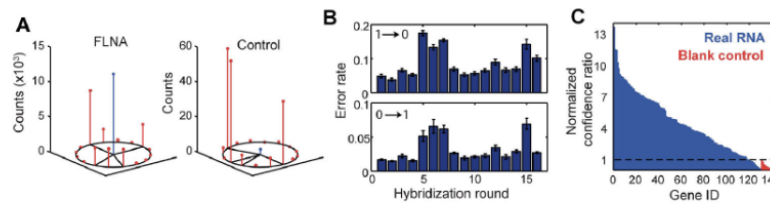


**Figure 3.** Example MERFISH data for a 16-bit MHD4 Code. (A) smFISH images from each of 16 rounds of hybridization of a small field of view of a Human fibroblast (IMR90) stained with encoding probes utilizing an 16-bit MHD4 code that encodes 140 RNAs. The label depicts the readout hybridization round corresponding to each image. Circles correspond to the locations of identified fluorescent spots. (B) A single 40- $\mu$ m-square field of view with all measured barcodes marked. The color of each marker represents the measured barcode. (Inset) An overlay of the small section of this field of view depicted in (A) with each set of overlapping spots labeled. White circles correspond to sets of spots that represent a barcode that can be decoded into a RNA while red represents a set of spots for which the measured barcode does not represent an RNA. (C) The measured binary barcodes for each set of spots in the small field of view depicted in (A) with the identity of the RNA represented by that barcode. Error correction was required for two barcodes in hybridization round 14 and is represented by red crosses. (D) The number of RNAs of each species identified in the single field of view depicted in (D). ~2000 RNAs were measured in this single field of view, and in a single measurement ~100 such fields of view containing 250,000 RNAs can be measured. (E) The average RNA copy number per cell measured with one implementation of the 16-bit MHD4 code versus the average copy number per cell for every RNA measured with another 16-bit MHD4 code in which each RNA was assigned a different barcode. The dashed line represents equality. (F) The average RNA copy number per cell versus the abundance as measured with bulk RNA-seq (FPKM). (G) The average copy number per cell measured via MERFISH versus that measured using conventional smFISH for 15 different RNAs. The dashed line represents equality. Panels reproduced with permission from (Chen et al., 2015).

## Calculate MERFISH Performance

The basic performance of a MERFISH measurement can be quickly assessed from several quantities that can be calculated from the barcodes decoded in the Decoding Barcodes Section. Here we describe the calculation of these quantities and their interpretation.

**The Per-Bit Error Rate**—If the encoding scheme has the capacity to correct errors, then it will also have the ability to determine the bit at which the corrected errors occurred. Using this ability, it is possible to compute the average '1'→'0' or '0'→'1' error rates associated with each round of hybridization. These quantities can be very useful in diagnosing the performance of a MERFISH run. For example, a high '1'→'0' error rate for one round of hybridization may indicate poor staining in that round or perhaps an inappropriately high threshold for spot fitting. Alternatively, a high '0'→'1' error rate may suggest that the spot fitting threshold is too low for that round of hybridization. To calculate this quantity, compute the number of times each RNA species was measured correctly  $N_c$  and the number of times it was measured with a word that contains a single-bit error at each possible position  $N_i$ . Figure 5A illustrates these counts for two example barcodes. These two quantities can then be used to estimate the probability of flipping the  $i$ th bit via  $N_i/(N_i + N_c)$ . (See the Materials and Methods of Chen et al. for a derivation of this expression.) Once the probability of flipping each bit is calculated for each RNA, an estimate of the bit flipping probability for each round is calculated from the average across all genes. See Figure 5B for an example of these per-bit error rates.



**Figure 5.**

MERFISH quality metrics. (A) Left: The number of molecule counts whose code exactly match that of FLNA (blue) and the number of molecule counts whose code differ in one bit from the FLNA barcode (red). (Right) as in (left) except for a barcode that was left intentionally unused to serve as a misidentification control (Control). Lines connecting the central exact counts to counts to barcodes that contain a single-bit error denote '1'→'0' errors. (B) The average rate at which a '1' → '0' (top) or '0' → '1' (bottom) error occurs at each bit. These error rates are derived from the ratios of the counts to the correct barcode (A, blue) relative to the counts to the barcodes that differ in a single-bit (A, red). (C) The confidence ratio for each used barcode (Real RNA, blue) from a 16-bit, MHD4 measurement normalized to the largest confidence ratio observed for the 'blank' barcodes (Blank control, red). All data are reproduced with permission from (Chen et al., 2015).

**Background Counts and the Confidence Ratio**—We recommend leaving a few percent of the possible barcodes unassigned. Measurements of these 'blank' or 'control' barcodes can then serve as a simple measure of the background count rate in MERFISH and can be used to assign a qualitative degree of confidence to the observed counts for different RNAs.

Perhaps the most obvious use of these 'blank' barcodes is to set a lower limit on the number of counts that can be trusted. For example, if an RNA-encoding barcode is counted fewer times than a 'blank' barcode, then it might be reasonable to assume that the counts for this RNA are dominated by background counts and should not be trusted. However, this use of the 'blank' barcodes provides an overly cautious measure of the rate of background counts because it does not properly take into consideration the way in which different binary barcodes are related to one another. For example, imagine that one 'blank' word is counted more frequently than a barcode representing a low abundance RNA. If these two barcodes are separated by a large HD, i.e. these barcodes have very few '1' bits in common, then it is possible that the errors that produced observations of this 'blank' barcode may not be relevant for the measurement of the barcode associated with the low abundance RNA.

To partially capture the relationship between different barcodes and improve estimates of the accuracy of different barcode measurements, we have developed a metric that we term the confidence ratio. This ratio is defined as the number of measured binary barcodes that exactly match the binary barcode for a given RNA over the sum of this number and the number of counts for all binary words that differ from this barcode in only one bit, i.e. the ratio of the blue bars to the sum of the blue bar and all red bars in Figure 5A. For an errorcorrecting encoding scheme, this quantity is simply the ratio of the number of times an RNA was observed without the application of error correction to the number that was observed with error correction applied. However, error-correction is not required to compute the confidence ratio. All that is required is that all barcodes are separated by

at least a HD of 2.

To illustrate the usefulness of this ratio, consider the following scenario. Imagine that one specific hybridization round is particularly problematic, and the '1'→'0' error rate for that round is higher than others. (As Figure 5B illustrates, there can be variation between errors rates in different rounds of hybridization). Then RNAs encoded with binary barcodes that contain a '1' in that bit will tend to have more single-bit errors. This will decrease the confidence ratio for these words by generating a larger proportion of counts for these singlebit errors. However, RNAs encoded with binary words that do not contain a '1' in that bit will not see an increase in the number of single-bit errors in this round, and the confidence ratio for these RNAs will not be lowered.

Finally, a word of caution: the confidence ratio does not imply a quantitative level of confidence in the counts for a RNA. For example, a ratio of 0.3 does not imply that 30% of the measured counts are correct. The proper interpretation of this ratio is more qualitative: one should trust the counts for a RNA with a confidence ratio of 0.5 more than those for a value of 0.3. To provide a heuristic cut-off on the confidence ratio, we use the measured confidence ratios for the 'blank' barcodes. If an RNA-encoding barcode has a confidence ratio larger than the largest confidence ratio observed for the 'blank' barcodes then we generally trust the counts associated with this RNA. See Figure 5C. Of course, this cut-off is still somewhat arbitrary, and we only recommend using it as a rule of thumb. The confidence ratio is one of the simplest quantities one could compute in an effort to exploit the natural geometric connectivity of the different measured binary barcodes. In the future, we expect that more complicated calculations could be performed to exploit this connectivity and further improve both the accuracy of the measured counts with MERFISH and our metrics of the performance of a given measurement.

## **(Optional) Iterative Identification of Optimal Thresholds**

As discussed in the Section of Identification of Fluorescent Spots, we observe that different rounds of hybridization produce somewhat different levels of background and spot brightness. Thus, it is typically advisable to tune the spot-finding threshold for each round of hybridization. To further improve this process, it is possible to automate this threshold search by using some of the performance metrics discussed in the Section of Calculate MERFISH Performance to determine the optimal threshold for each hybridization round.

## **Protocol**

### **Identification of Fluorescent Spots**

#### **Step 1.**

Data conversion. The first challenge associated with any analysis software will be to insure that the image data is in an appropriate format. Our data is typically stored in a custom file format. While conversion to this format is possible, we instead recommend converting data to the more commonly used tif format, which the 3D daoSTORM algorithm can process. Many microscopes can already save images as tif format, so conversion may not be necessary.

## Identification of Fluorescent Spots

### Step 2.

Determine the appropriate intensity threshold for spot finding. The daoSTORM algorithm requires that a threshold be selected to determine whether a spot is bright enough to fit or not. It is useful to first run spot finding software for a range of thresholds to determine which value performs the best at discriminating between bright spots and low intensity spots, which are likely due to off-target binding of encoding or readout probes. The brightness of spots can differ between different hybridization rounds, so it is often worthwhile to explore different thresholds for different readout probes.

#### 📌 NOTES

**Max Brown** 28 Dec 2017

In practice, it may be difficult to determine the optimal threshold for discriminating background spots and spots that correspond to RNAs by simple visual inspection. Thus, we recommend analyzing data with a range of thresholds and using an iterative approach to select the threshold that produces the best decoding of MERFISH data. This process is discussed in detail in Section Iterative Identification of Optimal Thresholds.

## Identification of Fluorescent Spots

### Step 3.

Batch analyze the data. Once an appropriate threshold has been selected, use this software to identify the location of potential molecules in both the smFISH images and the images of fiducial beads.

## Correction of Image Offsets

### Step 4.

Find fiducial bead centroids. We utilize the same spot fitting approach described in the Section of Identification of Fluorescent Spots to determine the location of the fiducial beads in the images of these beads corresponding to each round of hybridization.

## Correction of Image Offsets

### Step 5.

Create the affine transformations. For each field of view, load the locations of the fiducial beads in each round of hybridization. To use these beads as control points in the construction of affine transformation, each bead in one image must be associated with beads in another image. This can be accomplished by associating each bead with its nearest neighbor in the other image. However, this simple approach will fail if the offset between images is substantially larger than the average distance between beads within a single field of view since the nearest neighbor between images would likely correspond to a different bead. In this case, an initial crude offset can be found by using image cross



correlation either with the original bead images or with a lower resolution image created from the 2D histogram of the bead positions. This crude offset can be used to correct the positions of these beads to sufficient accuracy to allow simple association of nearest neighbors, which then allows the construction of the final affine transformation. Using this approach, we can re-register different frames with a residual error of roughly 20 nm, much better than a single pixel.

## Correction of Image Offsets

### Step 6.

Correct the location of fluorescent spots. Once the affine transformations are calculated, apply these transformations to the locations of all found fluorescent spots in each smFISH image and save the revised locations.

## Decoding Barcodes

### Step 7.

Create a list of all found spots. For a single field of view, create a list of the location of all spots in all rounds of hybridization. This algorithm will create an N-bit binary barcode for each of these spots.

## Decoding Barcodes

### Step 8.

Construct a barcode for each found spot. For each spot in the list created in Step 7, compute the distance to the nearest spot in each round of hybridization. This will create a vector of N distances. To convert this vector into a binary barcode, compare the *i*th distance to the maximum allowed distance (a parameter of this algorithm). If this distance is less than or equal to this maximum distance, assign a '1' to the *i*th bit of the barcode for this spot. Otherwise, assign a '0' to that bit.

## 📌 NOTES

**Max Brown** 28 Dec 2017

We use a maximum distance of 160 nm, i.e. one camera pixel. Practically, we found that a smaller maximum distance between spots tends to discard more real barcodes than false barcodes, despite the fact that we can resolve the centroid of each RNA spot to better than 160 nm. This observation may be due to the fact that the position of the centroid for the observed spot for a single RNA varies on average by ~100 nm between different rounds of hybridization—an effect that may be due to the finite cellular volume occupied by each RNA.

## Decoding Barcodes

### Step 9.

Remove redundant barcodes. This algorithm creates a barcode for all spots in all rounds of hybridization, ignoring the fact that spots in different rounds will be combined to form individual barcodes. Thus, each barcode may be replicated in the results returned by Step 8. Redundant barcodes will share the same set of spots, and, thus, the average centroid of the spots that form a barcode will be unique for each barcode. Thus, to remove these redundant barcodes, we compute the average centroid of all spots that comprise each barcode and remove extra copies of barcodes that share the same centroid.

We perform this step because it is computationally faster to create redundant barcodes and then remove them in this fashion than to identify such barcodes as they are created.

## Decoding Barcodes

### Step 10.

Associate each binary barcode with the RNA it encodes. Using the codebook designed in [Design of the Codebook](#) create a look-up table that associates each encoded RNA with all binary barcodes that encode it. If the encoding scheme is not capable of error correction, then there will only be one binary barcode for each RNA. If, on the other hand, the encoding scheme allows error correction, then each RNA will have multiple binary barcodes that correspond to that word. For the purposes of calculating several properties associated with the basic performance of MERFISH, it is useful to create two look up tables: one that corresponds to the correct binary barcodes and one that corresponds to all binary barcodes that can be matched to an RNA after error correction. The use of two look up tables will allow each decoded RNA to be marked as an exact match, i.e. no error correction was applied, or an error corrected match.

## Calculate MERFISH Performance

### Step 11.

Please see Guidelines for Calculating MERFISH Performance protocol.

## (Optional) Iterative Identification of Optimal Thresholds

### Step 12.

Run the spot-fitting algorithm for a range of spot-brightness thresholds. We analyze each MERFISH data set with 10-20 thresholds spanning a range set by examining these data by hand.

## (Optional) Iterative Identification of Optimal Thresholds

### Step 13.

Select a starting set of thresholds and decode barcodes using these found spots. The initial thresholds can be selected at random or one can start at one limit—i.e. all thresholds are the lowest or highest possible values. With these found spots, run the decoding algorithm discussed in the Section of Decoding Barcodes.

## (Optional) Iterative Identification of Optimal Thresholds

### Step 14.

Compute a quality metric. For the MHD4 code, where it is possible to calculate the per-bit error rates, we calculate the geometric mean of all  $1 \rightarrow 0$  and  $0 \rightarrow 1$  error rates. For the MHD2 code which does not have the ability to correct errors and calculate these perbit error rates, we utilize the ratio of the number of barcodes with four  $1$  bits to those measured with three or five  $1$  bits, i.e. the average confidence ratio.

## (Optional) Iterative Identification of Optimal Thresholds



## Step 15.

Iterate. Repeat Steps 13 and 14 for a variety of threshold combinations, and select the threshold combination that maximizes the selected quality metric. Typically, we change our threshold combinations during each iteration round by selecting one hybridization round, fixing the threshold for all other rounds, and screening through all possible threshold values for this round.

### NOTES

**Max Brown** 28 Dec 2017

Once all thresholds have been screened for this round, we select the threshold value for that round that produces the highest quality metric and then fix this threshold value for this hybridization round for subsequent iterations. We then repeat this process for each subsequent hybridization round. Once we have screened the thresholds for all hybridization rounds in this fashion, we repeat this process, often for a total of three times. We find that both quality metrics that we have used typically converge with only a few iterations through all hybridization rounds. More complicated search strategies are also possible.