Supplementary Material for

FISH-quant v2: a scalable and modular analysis tool for smFISH image analysis

Arthur Imbert[1,2,3], Wei Ouyang[4], Adham Safieddine[5], Emeline Coleno[6], Christophe Zimmer[7], Edouard Bertrand[6], Thomas Walter[1,2,3,#], Florian Mueller[7,#]

1. Centre for Computational Biology (CBIO), MINES ParisTech, PSL University, 60 Boulevard Saint Michel, 75272 Paris Cedex 06, France
2. Institut Curie, 75248 Paris Cedex, France
3. INSERM, U900, 75248 Paris Cedex, France
4. Science for Life Laboratory, School of Engineering Sciences in Chemistry, Biotechnology and Health, KTH – Royal Institute of Technology, Stockholm, Sweden
5. Sorbonne Université, CNRS, Institut de Biologie Paris-Seine (IBPS), Laboratoire de Biologie du Développement,F-75005 Paris, France
6. IGH, University of Montpellier, CNRS, Montpellier, France
7. Imaging and Modeling Unit, Institut Pasteur, UMR 3691 CNRS, C3BI USR 3756 IP CNRS, Paris, France

[#] Corresponding authors. Thomas Walter: Thomas.Walter@mines-paristech.fr; Florian Mueller: muellerf.research@gmail.com

Supplementary Note 1: RNA Detection

Simulated images to validate spot detection algorithms

To evaluate our spot detection algorithm, we simulated 3D images with spots mimicking the smFISH signal produced by RNA molecules. We used simulated, but realistic images in order to have precise control over the ground truth in terms of noise levels and RNA counts. With experimental data, these parameters are not accessible, and make a comprehensive validation difficult. Lastly, manual annotation of 3D smFISH images is very time-consuming and subjective.

Bash and Python scripts used for these simulations are available on GitHub (https://github.com/fish-quant/sim-fish/tree/main). These images are obtained with three main steps:

1. We randomly draw the **number of spots** (this parameter can also be predetermined) and their **localization** in three dimensions.
2. For each spot we simulate a **3D Gaussian signal** with a predefined *amplitude* and *sigma*. The final intensity of every pixel is sampled from a Poisson distribution with the gaussian simulated value as expectation.
3. We simulate a **background** white noise over all the image, following a normal distribution centered around a predefined *noise level*.

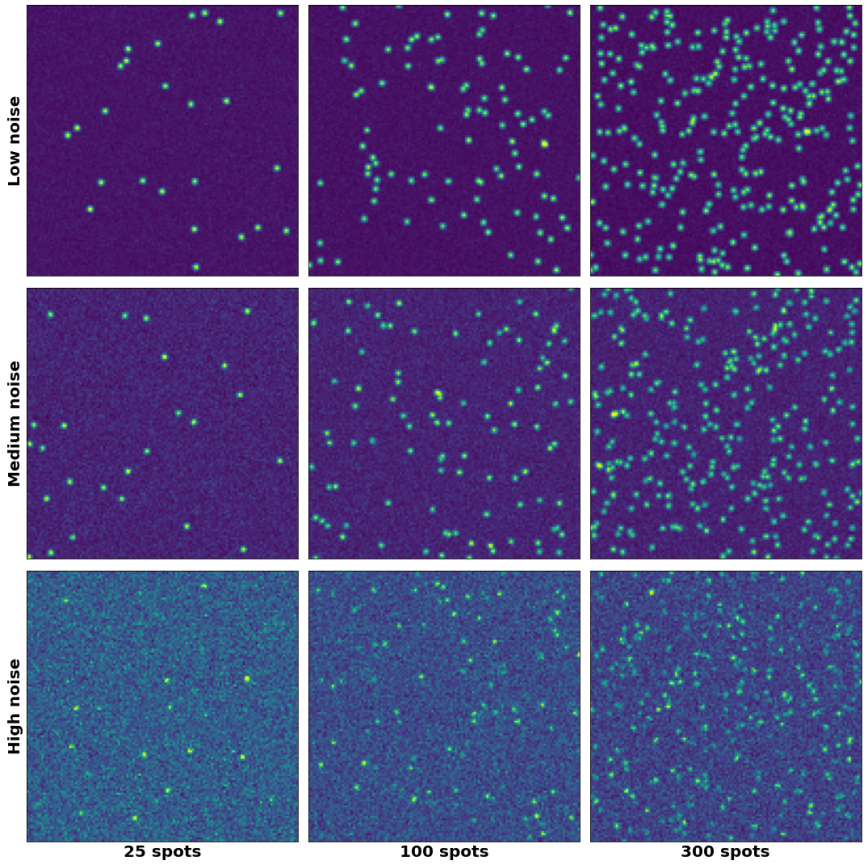

Figure S1. *Examples of simulated smFISH images with different noise levels and varying number of simulated spots.*

To evaluate detection in imaging conditions with different noise levels, we varied the parameters to simulate images. Signal quality was evaluated with the Signal-To-Noise ratio (SNR). SNR is calculated for each spot, where we define the background as a region surrounding its center and a radius twice the size of the spot radius, with the following equation:

$$SNR_{spot} = \frac{MAX(intensity_{spot}) - MEAN(intensity_{background})}{STD(intensity_{background})}$$

The function to calculate the SNR function is also available on GitHub: https://github.com/fish-quant/big-fish/tree/master/bigfish/detection/snr.py

For each image, we then estimated the mean SNR. We simulated images with SNR values between 2 and 30. We then group these SNR values in three regimes (**Figure S1**): high (SNR around 20), medium (SNR around 10) and low low noise (SNR around 3.5) .

Automated detection of individual and clustered RNAs

Automated spot detection

RNA spots are detected with a standard method (1) consisting of three steps:

1. Image is denoised and spots are enhanced by using a Laplacian of Gaussian (LoG) filter.
2. Peaks are detected in the filtered image with a local maximum detection algorithm.
3. An intensity threshold is applied to discriminate actual spots from noisy background.

In order to scale this spot detection to thousands of images, a fully automated approach is required. The size of the LoG filter can be derived from the expected size of the point spread function of our spots. This parameter is assumed to be known (or approximated) for a given experimental protocol. The critical parameter is the intensity threshold that usually needs to be adjusted for each probe-set targeting a specific RNA. We therefore implemented a heuristic approach to automatically set this threshold.

The detected local maxima are either actual spots or some background noise (autofluorescence, of-site binding of oligos, …). We assume that they have a different intensity distribution, and RNA spots have significantly higher intensity values since they are targeted by multiple oligos. Indeed, as we start increasing our detection threshold (**Figure S2**), we first observe a quick (and almost monotone) decrease in the number of spots detected. The threshold only removes background noise; real spots are too bright to be removed. At higher intensity values and good image quality, the number of detected spots reaches a plateau, corresponding to adequate intensity threshold. If the intensity threshold is further increased, true RNAs are missed and sensitivity decreases accordingly. While for high noise levels, this plateau may be less pronounced, we can still see an abrupt change in the slope of the curve, clearly separating the regimes of over- and underdetection.

In summary, if we plot the number of detected spots as a function of the intensity threshold, we observe an elbow curve. We then set the intensity threshold at the value where this curve breaks and has an abrupt change in gradient (red point in **Figure S2**). More specifically, we choose the threshold where the gradient is above the average gradient of the curve.

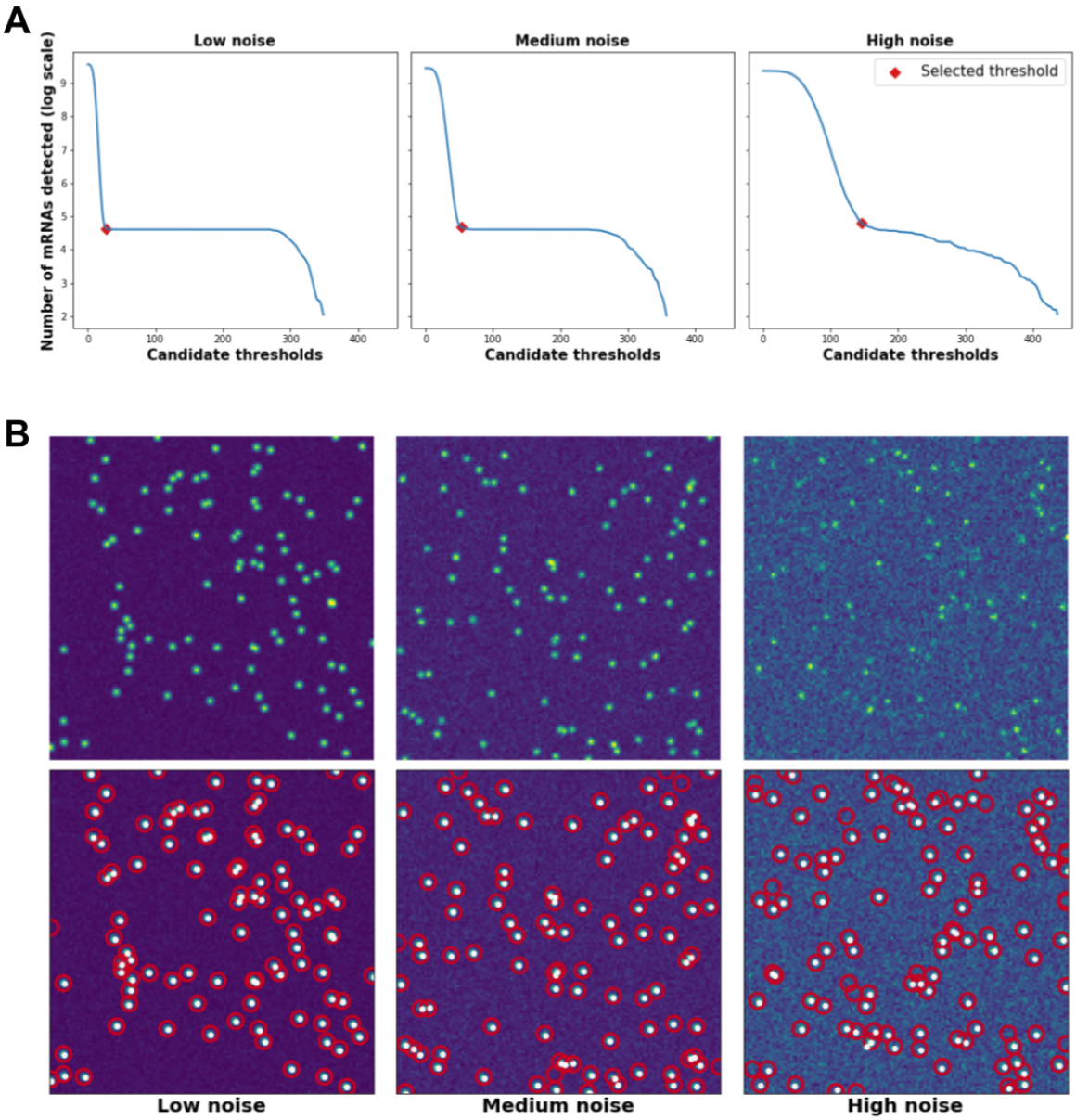

Figure S2. Automated detection of RNAs*. **(A)** Number of detected spots shown as a function of different intensity thresholds for different noise levels. Red dot indicates the value of the automatically determined threshold. **(B)** Simulated images with different noise levels containing 100 spots (top). Detected positions (red circle) and simulated positions (white dots).*

When testing our automated spot detection method on images with varying signal quality, we found that it performs very well for a SNR equal or greater than 8 (**Figure S3**). For a high level of noise (SNR between 2 and 5) we estimate an average error of 24% in terms of the number of detected spots when we simulate between 10 and 300 spots per image. However, for a medium or low level of noise (SNR between 8 and 26), we respectively estimate an average error of 5.5% and 1.4%.

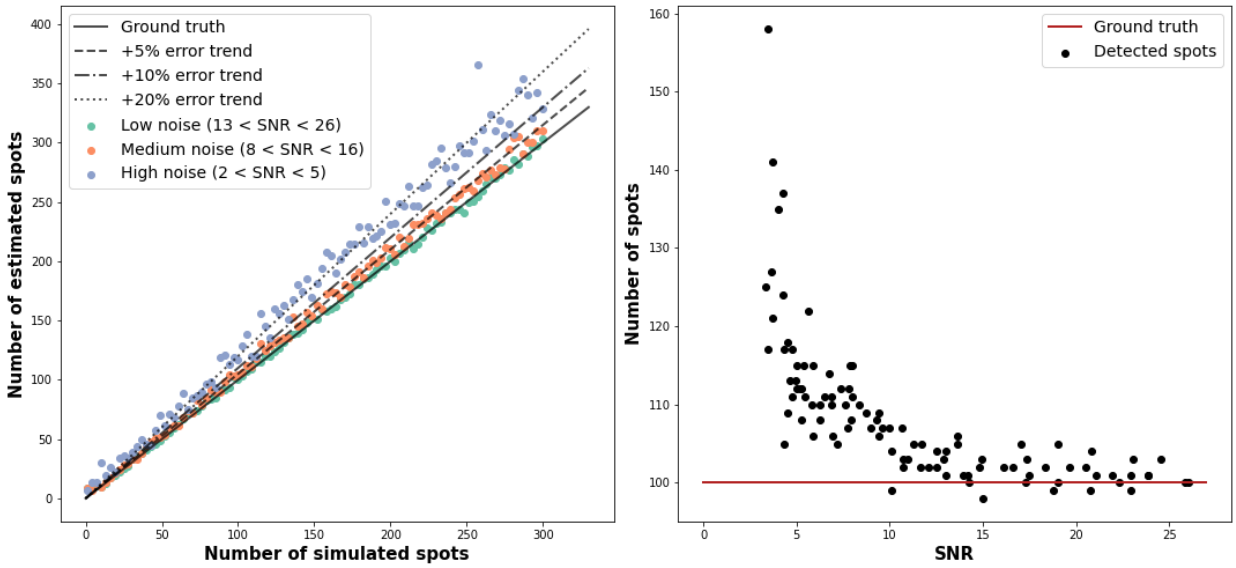

Figure S3. Impact of noise on automated detection. ***(Left)*** *Number of detected spots compared to the actual number of simulated spots simulated. For each noise regime, 100 images were simulated. Dashed trend lines indicate different estimation errors.* ***(Right)*** *Plot summarizes results for images simulated with 100 spots and different noise levels. Each dot corresponds to one image.*

Decomposition of dense regions and cluster detection

The above described standard spot detection approach fails to separate spots that aggregate in dense and bright areas, where no individual spots can be resolved. In this case, RNA counts are thus underestimated. To overcome this limitation, we provide an alternative approach where we first localize such dense areas and decompose them into individual RNAs. Secondly, we detect and quantify clustered RNAs using the individual RNA coordinates. Below, we describe these steps in more detail.

First, we implemented an approach where we **detect such dense areas and then iteratively populate them with additional RNAs** until the reconstructed image matches best the observed image. This process can be summarized in four main steps:

1. A reference spot is estimated from all detected spots, by calculating their median image.
2. The reference spot is fitted with a Gaussian function, and the estimated parameters represent our model for a noise free reference spot.
3. We detect potential spot clusters (candidate regions) as dense and bright areas, using a connected component algorithm. Such a candidate region has at least 2 connected pixels that are brighter than the reference spot.
4. For each candidate region, we implemented an iterative approach to create an image that matches best the identified candidate regions. Starting with an empty image, we compare at each iteration the simulated image with the actual image, and place a reference spot at the location with the maximum difference. Image similarity is estimated by their squared sum of squared residuals (SSR), and iteration is stopped once the SSR is not decreasing anymore.

Second, we can apply a spatial clustering algorithm (DBSCAN) (2) **on the obtained RNA point cloud to detect spatially localized clusters and count the RNAs inside**.

To evaluate this process, we simulated images with one cluster containing a specific number of spots (5, 10 and 15 spots) and different noise levels (**Figure S4**). Then, we used our method to infer the number of spots per simulated cluster. Please note, that this entire process from the initial RNA detection, decomposition to the cluster calling is entirely automated and requires no user intervention.

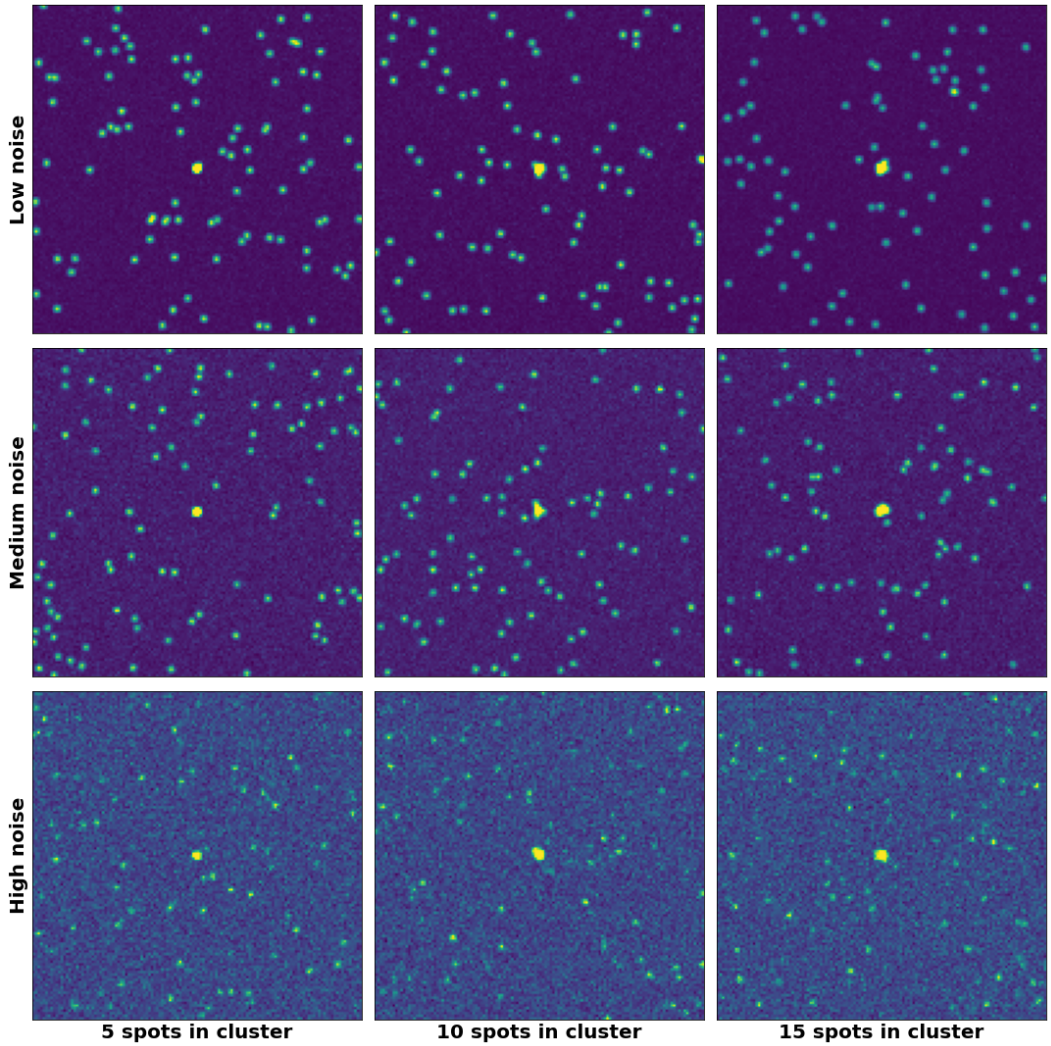

Figure S4. *Simulations with 100 spots, one unique cluster centered in the image and different levels of noise. We include different numbers of spots inside the cluster.*

Our decomposition and cluster detection approach performed robustly across the different noise levels of noise (**Figure S5**). The average error was 0.8 spots when simulating clusters with 5 or 10 spots, and 1.6 when simulating clusters with 15 spots.

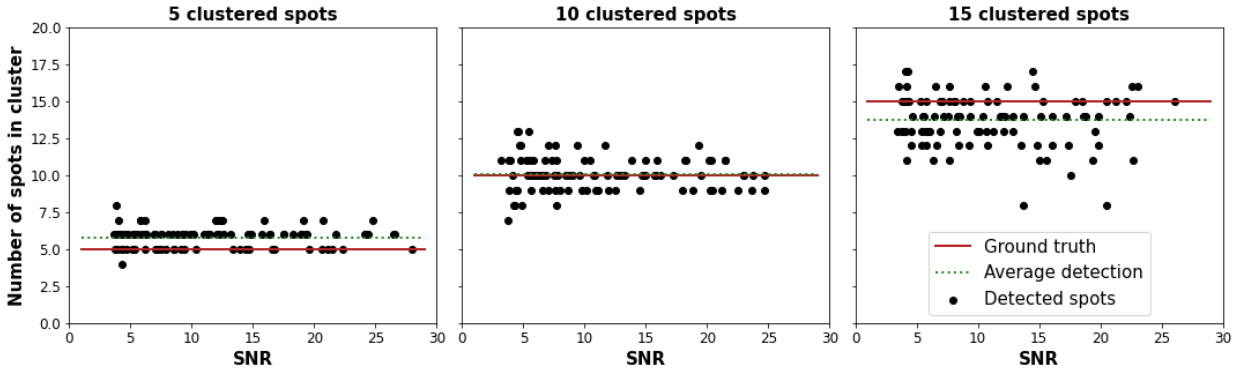

Figure S5. *Estimated number of RNAs per simulated cluster as a function of different SNR. Three different cluster sizes were simulated. Red line indicates the simulated ground truth, dashed green line the average number of detected RNAs per cluster.*

Supplementary Note 2: Segmentation with Deep Learning

We implemented deep learning models to segment nuclei directly from big-fish without using another API, package or framework. We train and evaluate these models on fluorescent images with 4 channels from a recent study (3) : one channel for nuclei segmentation (DAPI), three different channels with different image quality permitting cell segmentation (CellMask, smFISH and a GFP marker for centrosomes). To obtain ground-truth annotations, we pre-segmented 180 fields of view with Cellpose (5), and then manually corrected these predictions. We used 19 images for evaluation and the rest for training.

Our models use an encoder-decoder architecture like Unet (4, 5), with 4 downsampling stages (the spatial resolution of our images is divided by 16 at the bottom of the model). We also use residual blocks, for each stage, mimicking Cellpose (5). Models are trained with Adam optimizer until validation loss does not improve anymore. To evaluate our models, we use the *mean Average Precision (MAP)* as a metric. For each pair of predicted and ground truth instances, we compute their *Intersection over Union* score (IoU). They match if the IoU is above a specific value. For a given threshold, we can then compute *True Positives* (instances matched correctly), *False Positives* (predicted instances matching nothing), *False Negatives* (ground truth instances missed) and the *Average Precision* as:

$$AP = \frac{TP}{TP+FP+FN}$$

The *mean Average Precision (MAP)* is the average of this score for different IoU thresholds between 0.5 and 0.95. Higher values indicate better agreement between prediction and ground-truth.

Segmentation of nuclei is implemented as a pixelwise classification problem with three classes: background, foreground and nuclear boundary. The model then assigns each pixel to one of these 3 classes. To build a mask for every nucleus instance, we use its foreground predicted surface and apply a dilation of 1 pixel (**Figure S6A**). The model is trained with a categorical cross-entropy loss.

Model	Dapi	Cellmask	smFISH	GFP
3-classes Unet	0.6			

Distance map Unet		0.66	0.59	0.58
Distance map Unet (double input)		0.65	0.63	0.62

Table S1. *MAP for different input channels. For every channel, evaluation is performed over 19 images.*

For cell segmentation, we implemented two different models. First, the model uses one cell channel as an input (Cellmask, smFISH or GFP) and predicts the cell surface and a distance map to cell edges. These predictions are then processed with a watershed algorithm to both predictions (using segmented nuclei as seeds) to obtain a mask for every cell instance. We use a combined loss to train the model with a binary cross-entropy loss for the surface prediction and a mean absolute loss for the distance map. Best segmentation results were obtained with CellMask, while the smFISH and GFP channel yielded similar AP scores (**Table S1, Figure S6B-D**).

In a second approach, the model uses two input images (nucleus and cell channel) and predicts 3 images: the cell surface (as a binary map), a distance map to edges of the nuclei and a distance map to cell edges. Cell instances are obtained with the same strategy as above, i.e. by application of a watershed algorithm. We speculated that by adding the nuclear channel, segmentation of cells could become more accurate. As above, we use a combined loss to train the model, mixing a binary cross-entropy loss for the surface prediction and a mean absolute loss for both distance maps. This approach slightly improves segmentation results for the smFISH and GFP channel (**Table S1**).

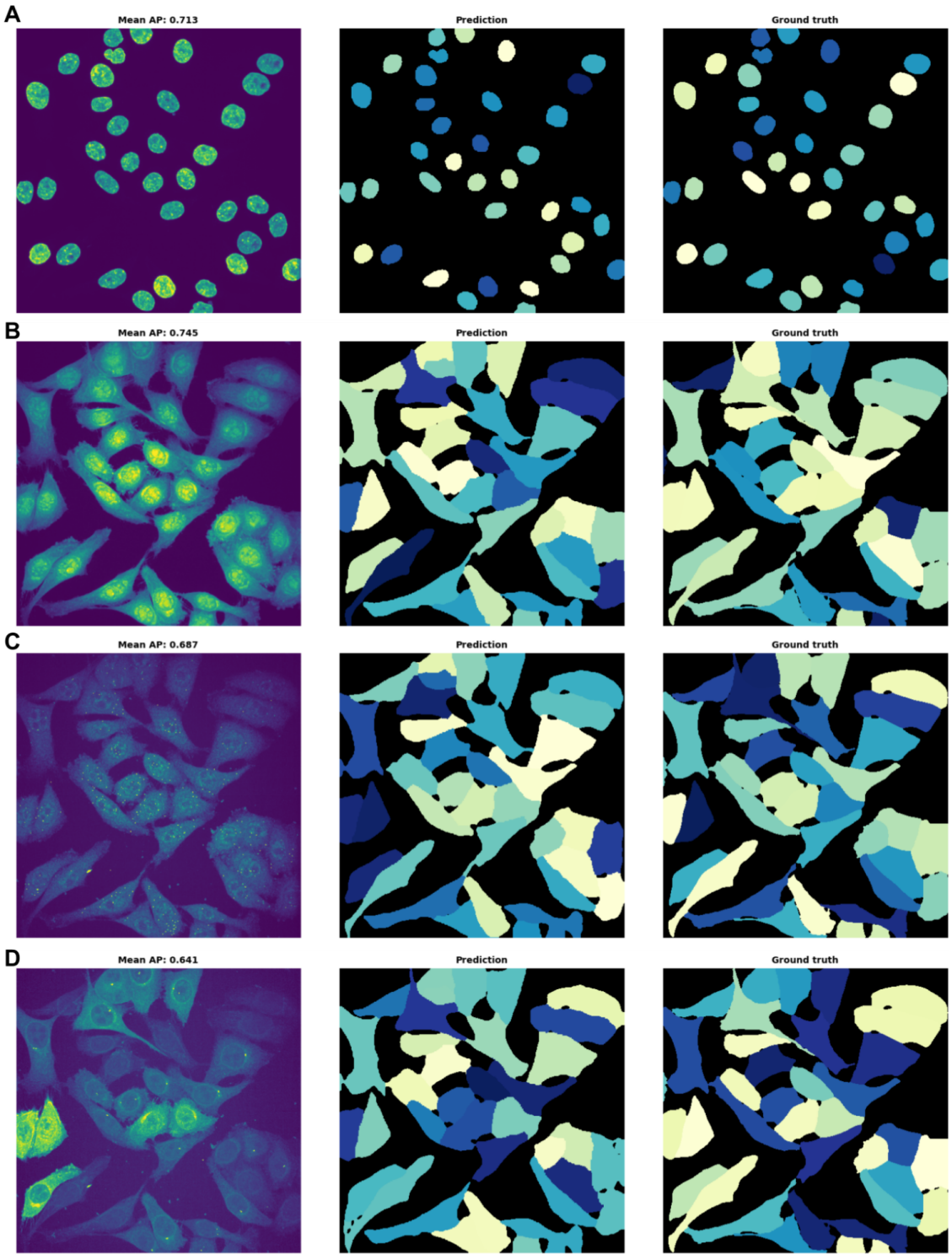

Figure S6. (A) *Example of nuclei segmentation with the DAPI channel as input.* ***(B-D)*** *Example of cell segmentation with a different channel as input (B=CellMask, C=smFISH, D=GFP). For cell segmentation, DAPI channel is also used as input with the double input model.*

Supplementary Note 3: Spatial features for RNA localization

Here, we list the different spatial features that are implemented in big-fish and permit a classification of cells based on their RNA localization patterns. Features are grouped into different classes.

Feature	Description
Area of nucleus	Measured in pixels.
Area of cell	Measured in pixels.
Area of cell extension	The cell area removed by an opening of size 3000nm (an erosion followed by a dilation). Measured in pixels.
Proportion of nucleus area	Proportion of the nucleus area compared to the entire cell.

Table S2. *Features to describe cell morphology.*

Feature	**Description**
Total number of mRNAs	-
Number of mRNAs inside nucleus	-
Proportion of mRNAs inside nucleus	Proportion of the mRNAs localizing inside the nucleus compared to the total number of mRNAs in the cell.
Proportion of mRNAs in foci	Proportion of mRNAs clustered in a foci compared to the total number of mRNAs in the cell.
Proportion of mRNAs in cell extension	Proportion of mRNAs in cell extension compared to the total number of mRNAs in the cell.
mRNA proportion along nuclear envelope	Proportion of mRNAs within 500 nm from the nuclear envelope compared to the total number of mRNAs in the cell.
Proportion of mRNAs in specific subcellular areas	Proportion of mRNAs in specific subcellular regions compared to the total number of mRNAs. Five concentric regions are defined around the nuclear envelope: 500-1000 nm, 1000-1500 nm, 1500-2000 nm, 2000-2500 nm and 2500-3000 nm. Six concentric regions are defined from the cell membrane: 0-500 nm, 500-1000 nm, 1000-1500 nm, 1500-2000 nm, 2000-2500 nm and 2500-3000 nm.
Polarization index	Measurement of the mRNA point cloud polarization in the cell. The higher the more polarized are the mRNAs. Index is computed based on the distance between the mRNAs centroid and the cell centroid. Details in (6).
Dispersion index	Measurement of the mRNA point cloud dispersion in the cell. The higher the more dispersed are the mRNAs. Details in (6).

Peripheral distribution index	Measurement of how close the mRNAs localize to the cell periphery. Details in (6).
Mean/median mRNA distance to centrosome	Expressed as an index: mean/median distance over the expected mean/median distance if mRNAs were distributed uniformly in the cell.
Proportion of mRNAs around centrosome	Proportion of mRNAs within 2000 nm from a centrosome compared to the total number of mRNAs in the cell.

Table S3. *Features to describe non-random RNA localization patterns.*

Supplementary note 4: Comparison of existing analysis tools

Method Framework	Automated spot detection	Segmentation	Localization features	Language	GUI	Ref
Object segmentation						
NucleAIzer	No	Yes	No	Python, MATLAB	Yes	(7)
Cellpose	No	Yes	No	Python	Yes	(5)
EmbedSeg	No	Yes	No	Python	No	(8)
Stardist	No	Yes	No	Python	Yes	(9)
Ilastik	No	Yes	No	Python	Yes	(10)
Spot detection in smFISH images						
RS-FISH	Semi-automated	No	No	Java	Yes	(11)
DeepBlink	Yes	No	No	Python	No	(12)
TrackMate	Semi-automated	Yes	No	Java	Yes	(13)
General purpose, modular image processing tools						
CellCognition	No	Yes	No	Python	Yes	(14)
CellProfiler	Semi-automated	Yes	To be adapted	Python	Yes	(15)
ICY	Semi-automated	Yes	To be adapted	Java	Yes	(16)
Frameworks for smFISH analysis						
StarFISH	Semi-automated	Yes	No	Python	No	(17)
FISH-quant v1	Semi-automated	Yes	Yes	MATLAB, Python	Yes	(1)
DypFISH	No	No	Yes	Python	No	(18)
FISH-quant v2	Yes	Yes	Yes	Python	Yes	

Table S1. *Comparison of different methods and software that can be used in a smFISH study and analyze sub-cellular RNA localization. This table provides a non-exhaustive overview of different tools.*

References

1. Mueller,F., Senecal,A., Tantale,K., Marie-Nelly,H., Ly,N., Collin,O., Basyuk,E., Bertrand,E., Darzacq,X. and Zimmer,C. (2013) FISH-quant: automatic counting of transcripts in 3D FISH images. *Nat. Methods*, **10**, 277–278.
2. Hahsler,M., Piekenbrock,M. and Doran,D. (2019) dbscan: Fast Density-Based Clustering with R. *J. Stat. Softw.*, **91**, 1–30.
3. Safieddine,A., Coleno,E., Salloum,S., Imbert,A., Traboulsi,A.-M., Kwon,O.S., Lionneton,F., Georget,V., Robert,M.-C., Gostan,T., *et al.* (2021) A choreography of centrosomal mRNAs reveals a conserved localization mechanism involving active polysome transport. *Nat. Commun.*, **12**, 1352.
4. Ronneberger,O., Fischer,P. and Brox,T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab,N., Hornegger,J., Wells,W.M., Frangi,A.F. (eds), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science. Springer International Publishing, pp. 234–241.
5. Stringer,C., Wang,T., Michaelos,M. and Pachitariu,M. (2021) Cellpose: a generalist algorithm for cellular segmentation. *Nat. Methods*, **18**, 100–106.
6. Stueland,M., Wang,T., Park,H.Y. and Mili,S. (2019) RDI Calculator: An Analysis Tool to Assess RNA Distributions in Cells. *Sci. Rep.*, **9**, 8267.
7. Hollandi,R., Szkalisity,A., Toth,T., Tasnadi,E., Molnar,C., Mathe,B., Grexa,I., Molnar,J., Balind,A., Gorbe,M., *et al.* (2020) nucleAIzer: A Parameter-free Deep Learning Framework for Nucleus Segmentation Using Image Style Transfer. *Cell Syst.*, **10**, 453-458.e6.
8. Lalit,M., Tomancak,P. and Jug,F. (2021) Embedding-based Instance Segmentation in Microscopy. *ArXiv210110033 Cs Eess*.
9. Schmidt,U., Weigert,M., Broaddus,C. and Myers,G. (2018) Cell Detection with Star-Convex Polygons. In Frangi,A.F., Schnabel,J.A., Davatzikos,C., Alberola-López,C., Fichtinger,G. (eds), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 265–273.
10. Berg,S., Kutra,D., Kroeger,T., Straehle,C.N., Kausler,B.X., Haubold,C., Schiegg,M., Ales,J., Beier,T., Rudy,M., *et al.* (2019) ilastik: interactive machine learning for (bio)image analysis. *Nat. Methods*, **16**, 1226–1232.
11. Bahry,E., Breimann,L., Epstein,L., Kolyvanov,K., Harrington,K.I.S., Lionnet,T. and Preibisch,S. (2021) RS-FISH: Precise, interactive and scalable smFISH spot detection using Radial Symmetry.
12. Eichenberger,B.T., Zhan,Y., Rempfler,M., Giorgetti,L. and Chao,J.A. (2021) deepBlink: threshold-independent detection and localization of diffraction-limited spots. *Nucleic Acids Res.*, **49**, 7292–7297.
13. Ershov,D., Phan,M.-S., Pylvänäinen,J.W., Rigaud,S.U., Blanc,L.L., Charles-Orszag,A., Conway,J.R.W., Laine,R.F., Roy,N.H., Bonazzi,D., *et al.* (2021) Bringing TrackMate into the era of machine-learning and deep-learning.
14. Held,M., Schmitz,M.H.A., Fischer,B., Walter,T., Neumann,B., Olma,M.H., Peter,M., Ellenberg,J. and Gerlich,D.W. (2010) CellCognition: time-resolved phenotype annotation in high-throughput live cell imaging. *Nat. Methods*, **7**, 747–754.
15. McQuin,C., Goodman,A., Chernyshev,V., Kamentsky,L., Cimini,B.A., Karhohs,K.W., Doan,M., Ding,L., Rafelski,S.M., Thirstrup,D., *et al.* (2018) CellProfiler 3.0: Next-generation image processing for biology. *PLOS Biol.*, **16**, e2005970.
16. de Chaumont,F., Dallongeville,S., Chenouard,N., Hervé,N., Pop,S., Provoost,T., Meas-Yedid,V., Pankajakshan,P., Lecomte,T., Le Montagner,Y., *et al.* (2012) Icy: an open bioimage informatics platform for extended reproducible research. *Nat. Methods*, **9**, 690–696.
17. Perkel,J.M. (2019) Starfish enterprise: finding RNA patterns in single cells. *Nature*, **572**, 549–551.
18. Savulescu,A.F., Brackin,R., Bouilhol,E., Dartigues,B., Warrell,J.H., Pimentel,M.R., Beaume,N., Fortunato,I.C., Dallongeville,S., Boulle,M., *et al.* (2021) Interrogating RNA and protein spatial subcellular distribution in smFISH data with DypFISH. *Cell Rep. Methods*, **1**, 100068.