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

Arthur Imbert, Wei Ouyang, Adham Safieddine, Emeline Coleno, Christophe Zimmer, Edouard Bertrand, Thomas Walter, Florian Mueller

FISH-quant v2: a scalable and modular 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] and FLORIAN MUELLER[7]

[1]Centre for Computational Biology (CBIO), MINES ParisTech, PSL University, 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, 17165 Solna, 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, 34090 Montpellier, France
[7]Imaging and Modeling Unit, Institut Pasteur, UMR 3691 CNRS, C3BI USR 3756 IP CNRS, 75015 Paris, France

ABSTRACT

Regulation of RNA abundance and localization is a key step in gene expression control. Single-molecule RNA fluorescence in situ hybridization (smFISH) is a widely used single-cell-single-molecule imaging technique enabling quantitative studies of gene expression and its regulatory mechanisms. Today, these methods are applicable at a large scale, which in turn come with a need for adequate tools for data analysis and exploration. Here, we present FISH-quant v2, a highly modular tool accessible for both experts and non-experts. Our user-friendly package allows the user to segment nuclei and cells, detect isolated RNAs, decompose dense RNA clusters, quantify RNA localization patterns and visualize these results both at the single-cell level and variations within the cell population. This tool was validated and applied on large-scale smFISH image data sets, revealing diverse subcellular RNA localization patterns and a surprisingly high degree of cell-to-cell heterogeneity.

Keywords: image analysis; RNA localization; transcription; smFISH

INTRODUCTION

Regulation of gene expression is essential for a cell to fulfill its basic functions, and its dysregulation can lead to serious failures at the cellular, tissular and organism level. Transcription levels are not only tightly regulated, but for many genes it has now been demonstrated that their transcripts accumulate in specific regions in the cell, thereby producing intricate localization patterns. Such subcellular targeting of mRNAs is thought to play an important role for the spatial control of gene expression and improper RNA trafficking is linked to an increasing number of diseases (Buxbaum et al. 2014; Chin and Lécuyer 2017). However, the function and mechanisms of RNA localization are not fully understood and we still lack a view of this process at the transcriptomic scale.

RNA abundance and localization can be studied at a large scale by image-based assays, where individual mRNA molecules are visualized by single-molecule Fluorescence in situ hybridization (smFISH). This technique allows for the detection of individual mRNA molecules in their native cellular environment (Raj et al. 2008; Tsanov et al. 2016) by targeting each mRNA with several fluorescently labeled oligonucleotides. Many variants of this method exist, with optimizations regarding signal-to-noise ratio (SNR), experimental protocol, targeting specificity, scalability, automatization, and cost (for review, see Pichon et al. 2018). Furthermore, an increasing number of multiplexing methods have also been proposed over the last years, enabling the simultaneous imaging of up to 10,000 RNA species in cells and tissues (Moffitt and Zhuang 2016; Eng et al. 2019). Usually, smFISH experiments are complemented by the use of one or several fluorescent markers highlighting relevant compartments in the

Corresponding authors: Thomas.Walter@mines-paristech.fr, fmueller@pasteur.fr

Article is online at http://www.rnajournal.org/cgi/doi/10.1261/rna.079073.121. Freely available online through the *RNA* Open Access option.

cell, such as the nucleus, the cytoplasm or any organelle that might serve as a reference, depending on the focus of the study.

These scalable imaging techniques produce extremely large and complex image data sets exploring spatial distributions of large portions of the transcriptome. While large-scale imaging methods provide a systematic tool to understand RNA localization at a systems level, they come at a price: the need for fully automated, robust image analysis and user-friendly software tools to analyze such data sets and to fully exploit their potential (Pichon et al. 2018; Das et al. 2021).

Several specifications can be defined a priori for such an analysis tool. It should be simple enough to be mastered by non-experts, especially noncoders. Yet, it should be flexible enough to address different experimental designs and rely on a common algorithmic backbone. With the same modules, users should be able to both perform a high content screening analysis in a remote cluster, and a local analysis of a single image. Finally, the software should integrate the latest generation of computer vision algorithms, in particular deep-learning-based methods for image segmentation (Ronneberger et al. 2015; Falk et al. 2019; Stringer et al. 2021).

Here, we introduce a Python-based version of our widely adopted software package FISH-quant (Mueller et al. 2013) for the analysis of smFISH images. Contrary to the first version of FISH-quant in Matlab, we address and improve on each of the specifications mentioned above. The switch to Python allows us to develop a flexible, free and fully open-source software. FISH-quant v2 enjoys a better integration to other open source tools and frameworks, from data analysis to web-based user interaction. Importantly, FISH-quant v2 facilitates the use of machine learning or deep learning algorithms with the import of dedicated packages, such as scikit-learn (Pedregosa et al. 2011) or TensorFlow (Abadi et al. 2016). We also improve the scalability and the modularity of the package: the software has now been applied to several High Content Screening projects (Chouaib et al. 2020; Pichon et al. 2021; Safieddine et al. 2021). Lastly, by using ImJoy (Ouyang et al. 2019), a recently developed data analysis framework, we provide web-based graphical user interfaces (GUI) for both launching image analysis and downstream analysis of the results, and the computation can be performed locally or seamlessly scale to powerful remote computing servers.

RESULTS

The analysis of smFISH images aims at localizing and counting individual RNAs with respect to single cells and other subcellular landmarks. It typically encompasses a sequence of interconnected steps: (i) segmenting cells and the relevant cellular compartments such as nuclei (depending on the focus of the study and the markers used), (ii) detecting isolated and clustered RNA molecules, (iii) assignment of spots to cells, and (iv) analysis of expression levels and RNA localization patterns (Battich et al. 2013; Mueller et al. 2013; Stoeger et al. 2015; Tsanov et al. 2016; Samacoits et al. 2018), potentially in combination with other phenotypic features (Battich et al. 2015; Safieddine et al. 2021).

Overview of existing analysis solutions

While several tools exist for each of these steps, there is currently—to our knowledge—no tool available that permits performing the entire analysis in one framework (see Supplemental Note 4). A complete analysis pipeline has then to be built by mixing these tools and requires some in-house developments, which can be daunting for non-specialists and may provide solutions that are unstable and difficult to scale.

For the first step of object segmentation, deep-learning has become the method of choice with dramatic improvements in segmentation accuracy as compared to traditional methods. Several approaches exist that allow segmentation of cells and/or nuclei with minimal adjustment on new data sets, thanks to optimized models and large and diverse training data (Schmidt et al. 2018; Hollandi et al. 2020; Lalit et al. 2021; Stringer et al. 2021). The second step, fluorescence spot detection, has been addressed by a number of approaches in the literature, and more recently solutions specifically adapted to smFISH have been proposed. RS-FISH allows robust and accurate detection of fluorescent spots in 2D and 3D through radial symmetry but requires parameter tuning before being scaled to a large set of images (Bahry et al. 2021). DeepLink is a parameter-free deep-learning-based method, but is currently only available for 2D data and might require retraining (Eichenberger et al. 2021). Lastly, assigning spot counts to segmentation results and the subsequent analysis of RNA levels and/or RNA localization requires custom-written code (Stoeger et al. 2015; Samacoits et al. 2018).

General image analysis tools such as CellProfiler (McQuin et al. 2018) permit us to establish an analysis framework daisy-chaining some of these analysis steps, but do not permit us to perform the entire analysis. A number of approaches, specifically dedicated to the analysis of smFISH are available. In our own software FISH-quant v1 (Mueller et al. 2013) and also (Stoeger et al. 2015), the core of the analysis was performed in Matlab while cell segmentation was performed with the Python-based CellProfiler. DypFISH (Savulescu et al. 2021) permits the study of the spatial distribution of mRNAs and proteins of micropatterned cells, mixing tools implemented in Python and Icy (de Chaumont et al. 2012). Lastly, StarFISH (Perkel 2019) is an ongoing software

development mainly aiming at solving problems related to multiplex smFISH data for application in spatial transcriptomics.

FISH-quant v2: a complete toolbox for smFISH analysis

While an impressive range of methods already exists, a unified framework is lacking, which prevents users, especially non-specialist, from performing their smFISH analysis. To address this, we designed FISH-quant v2 to fulfill the above-described requirements in a flexible and efficient way. This version is entirely open-source and hosted on GitHub under the FISH-quant organization (Fig. 1, https://github.com/FISH-quant). Using a GitHub organization allowed us to provide dedicated repositories with well defined and dedicated scope. Further, it gives the flexibility for future extension where new projects can be integrated as new, independent repositories, without affecting and complexifying the already existing code. The user can choose the adequate code for the analysis needs, without the overhead of installing unnecessary packages.

This GitHub organization is organized in several resources with dedicated repositories and documentation. First, a Python package (Big-FISH) providing the core code for performing computation and analysis. Second, detailed interactive examples with test data for each analysis step implemented in Jupyter notebooks. These examples can be run directly on Binder (Project Jupyter et al. 2018), a free and reproducible Jupyter notebook service, without local installation. Third, a repository containing code to simulate different subcellular RNA localization patterns (Sim-FISH). We recently showed how such images can be used to develop and validate analysis pipelines with the goal to quantify such intracellular RNA distributions (Samacoits et al. 2018; Dubois et al. 2019). Fourth, ImJoy plugins (Ouyang et al. 2019) provide a graphical user-interface for the most commonly used workflows, and an interactive tutorial that can also run directly without local installation. Lastly, code from future projects either using or further improving FISH-quant will also be hosted here, creating a valuable, centralized resource for the community. A landing page (https://fish-quant.github.io/) directs new users to the most relevant resource for their analysis needs.

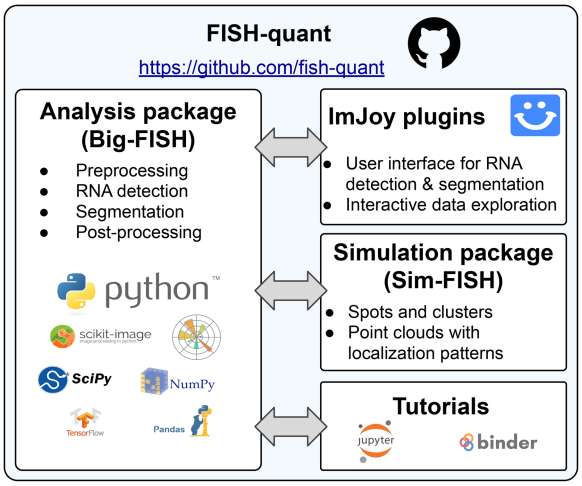

FIGURE 1. Organization of FISH-quant. FISH-quant is hosted on GitHub and consists of several interconnected repositories. The Python core package contains the entire analysis code, which is used by both the ImJoy plugins and the example and tutorial repository.

Big-FISH: Python package for smFISH analysis

We chose Python for the implementation of the core analysis package for several reasons: it allows the development of a free and fully open-source software, it provides established libraries for data and image analysis and is the language of choice for deep-learning implementations. Lastly, it can be interfaced with other tools and frameworks, from data analysis to web design, for instance with ImJoy (Ouyang et al. 2019) to provide interactive tools for user interaction and data inspection.

Our Python package includes several independent subpackages fitting the described workflow (see Materials and Methods for more details): preprocessing, segmentation, detection, and analysis. We designed each subpackage with clearly defined input and output data formats, which will be automatically checked. This then allows using each of these packages independently in a modular fashion. Users can thus create a customized analysis workflow, starting from preprocessing of images to statistical interpretation of results. These workflows can be implemented in Python and Bash scripts and run both on local and remote computational resources. The modular design also permits the easy integration of external methods, for instance, a new segmentation method can be combined with our spot detection algorithm. Lastly, we provide a subpackage to visualize the results of each intermediate step in the analysis workflow and thus provide valuable visual quality control.

Here, we will only provide an overview of these subpackages (Fig. 2). For a more detailed description of algorithms and methods, we refer to the documentation (https://big-fish.readthedocs.io/en/stable/) and the dedicated tutorials (https://github.com/fish-quant/big-fish-examples). These tutorials can be run directly in the browser with provided test data, and thus allow new users to immediately test these tools. The described methods were developed and validated with the data from two large-screen smFISH studies (Chouaib et al. 2020; Safieddine et al. 2021) (see Materials and Methods).

For image handling and preprocessing, we implemented a number of different utility functions to read, write, normalize, cast, filter, and project images. Different image file

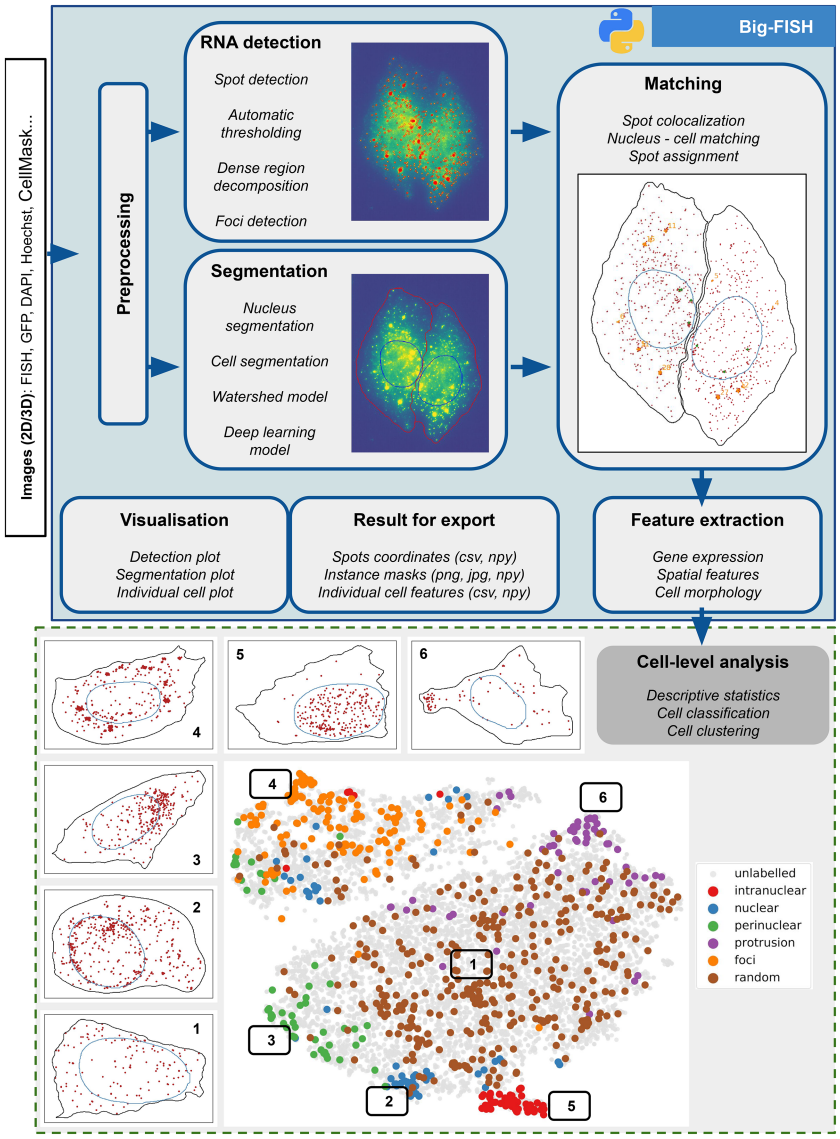

FIGURE 2. Big-FISH: the core analysis Python analysis package. (*Upper* part) Main modules illustrated with a typical analysis workflow. Shown are also the inputs and outputs that are created at the different steps. (*Lower* part) As a final result of the analysis of Big-FISH, each cell is described with a set of features reflecting RNA abundance and localization. These features can then be used to perform analysis on the cell population. Shown are results from our RNA localization screen where cells are grouped based on their RNA localization pattern (Chouaib et al. 2020). The t-SNE plot projects 15 localization features for smFISH experiments against 27 different genes. Each dot is one cell. The color-coded dots are manual annotations of six different localization patterns. Images are examples of individual cells displaying a typical localization pattern of this region of the t-SNE plot.

formats are natively supported and both 2D and 3D images can be processed.

The detection subpackage implements the methods required to detect spots in 2D or 3D images (Figs. 2, 3A–E). An important aspect of the detection subpackage is its ability to detect spots without setting any pixel intensity threshold. We implemented a method to automatically infer this threshold from the image. The curve describing the number of detected spots as a function of the intensity threshold (Fig. 3A,B) has an elbow shape, resulting from the superposition of the fast decreasing false positive detections (low intensity noise) and the slowly decreasing true positives. The threshold selected corresponds to the kink in the elbow, and corresponds thus to the highest threshold outside the high-noise regime. In order to validate this approach, we simulated realistic smFISH images with varying noise levels (Fig. 3A,B; Supplemental Note 1). We found that our method only leads to a moderate over-estimation of detected spots (<5%–10%) for images with moderate to high SNR values (>5). Such automatization overcomes human intervention and allows scaling to large data sets, such that the subpackage can process thousands of images. While initially designed to detect individual mRNAs, the same methods can also be used to detect other spot-like structures (Safieddine et al. 2021), such as centrosomes, P-bodies, etc (Fig. 3E).This subpackage further permits us to perform localization of RNAs with subpixel accuracy by using a Gaussian fitting (Mueller et al. 2013). Lastly, we provide the possibility to perform a colocalization analysis between spot detection performed in multiple channels (Cornes et al. 2021).

Strong local accumulation of RNAs, for example, active transcription sites, RNA foci, or areas of local translation (Chouaib et al. 2020), can lead to an underdetection since such accumulations are counted as single RNAs. For such cases, we provide tools to decompose these dense regions and estimate the number of spots based on our earlier work (Fig. 3C; Samacoits et al. 2018). We validated this approach again on simulated data (Fig. 3D; see Supplemental Note 1), and found consistent performance across relevant noise levels.

The segmentation subpackage contains several algorithms and utility functions for segmentation and post-processing. It provides deep-learning-based approaches to segment cells and nuclei (Figs. 2, 3F,G; Supplemental Note 2). Furthermore, we provide post-processing tools to refine and clean the segmentation result, such as boundary smoothing, removal of small

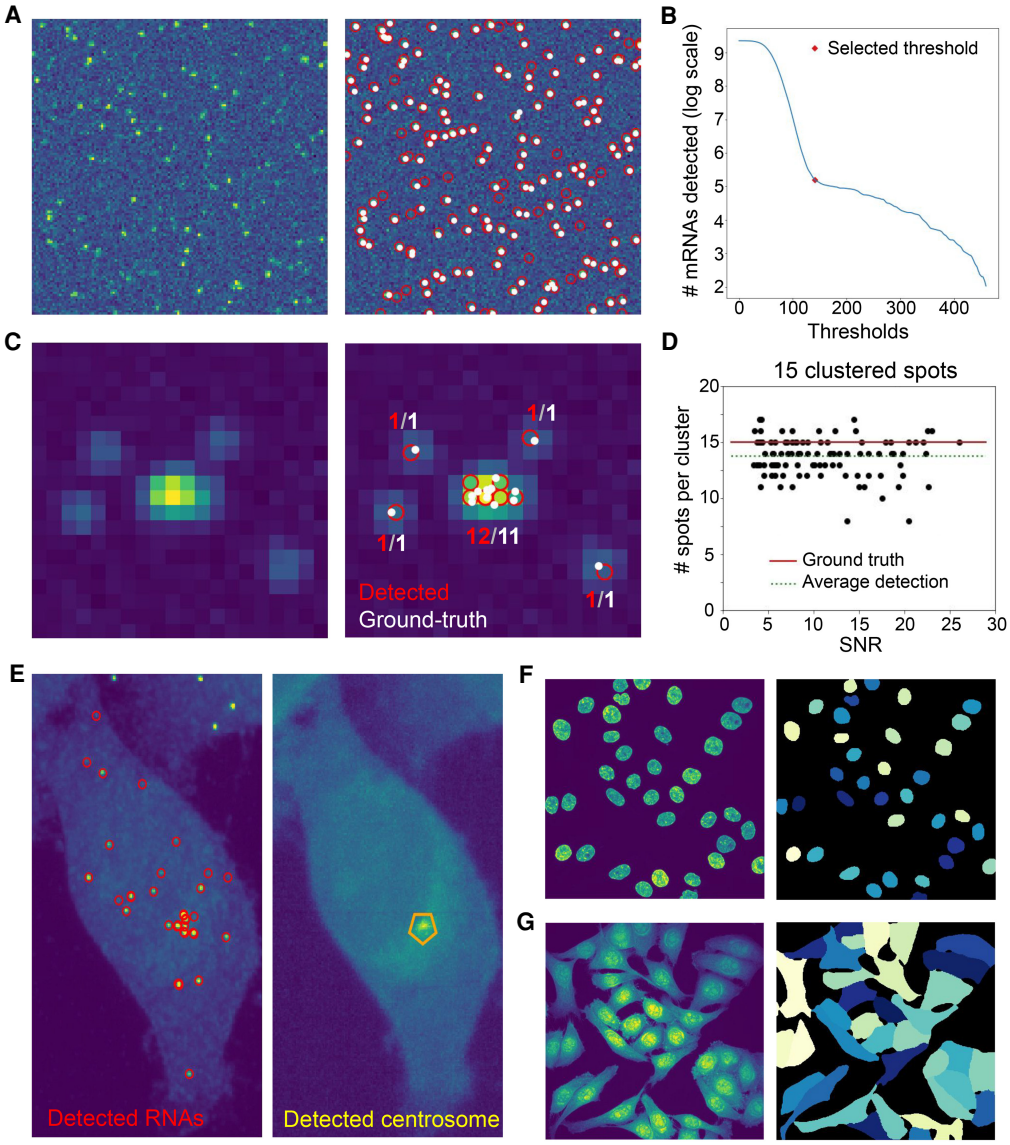

FIGURE 3. (*A*) Automated spot detection. Simulated image (*left*) and detection results (*right*) with detected spots in red and ground truth in white. (*B*) Elbow curve used for automated threshold setting, red dot indicates identified intensity threshold. (*C*) Decomposition of dense regions. Simulated image (*left*) and decomposition results (*right*) with detected spots in red and ground truth in white. Number of simulated and detected spots are shown in white and red, respectively. (*D*) Algorithm to decompose dense regions was evaluated with 100 simulated images containing a cluster of 15 spots and different noise levels. (*E*) Example of automated detection of *BICD2* mRNAs (*left*) and centrosome (*right*) in HeLa cells. (*F*) Example of nucleus segmentation from a DAPI image. (*G*) Example of cell segmentation from a CellMask image.

objects or filling of small holes. Lastly, morphological properties, such as the area of cells, nuclei or protrusions, can be computed for these components (Supplemental Note 3).

The cell matching subpackage allows combining results from detection and segmentation, permitting us to analyze RNA abundance and distribution at the single-cell level. Detected spots can be assigned to a specific region of interest, for instance, a cell or a nucleus. Using the same method, RNA clusters can be assigned to a nucleus and thus be considered as transcription sites. RNA expression levels are extracted within this subpackage, as this is usually the minimum information that is extracted from this kind of image.

The localization feature extraction subpackage permits the extraction of further information to study the subcellular spatial distribution of mRNA molecules. It gathers methods to format spot positions and coordinates of cellular landmarks and compute several spatial features at the single-cell level (Fig. 2; Supplemental Note 3). These features allow a statistical description of the cell population (Pichon et al. 2021; Safieddine et al. 2021) or can feed a classification model permitting us to classify individual cells based on their RNA localization patterns (Fig. 2; Chouaib et al. 2020).

Sim-FISH: simulation of smFISH images and RNA localization patterns

Simulations can be used to validate different steps of the analysis pipeline, ranging from the spot detection (Tsanov et al. 2016) to a statistical framework to quantitatively study RNA localization (Samacoits et al. 2018; Dubois et al. 2019). As mentioned above, we validated both our spot detection and our decomposition method for dense regions with this package (Fig. 3A–D; Supplemental Note 1). We simulate realistic smFISH images in three steps (see Supplemental Note 1). First, we randomly generate 2D or 3D spot coordinates, which can be random or display a specific subcellular RNA localization pattern. Further, clustered RNAs can be added. Second, we simulate a realistic image from these coordinates by modeling a RNA spot with a Gaussian function. Third, we add a noisy background to this image.

ImJoy: interactive user interfaces and data exploration

Our Python core analysis package provides flexibility and scalability since its components can be adapted to the specific analysis need of a given project. However, they require at least a minimum knowledge of Python to establish a complete workflow by using the provided tutorials.

To provide simpler access for users with no computational background and no programming skills, we implemented several plugins with graphical user interfaces for our computational platform ImJoy (Ouyang et al. 2019). These plugins provide the most commonly used analysis workflow, as we determined from the usage of the Matlab version of FISH-quant, and will thus be suited for a large number of use cases (Fig. 4). First, a plugin to perform deep-learning-based segmentation. This is currently built on top of CellPose (Stringer et al. 2021), but thanks to our modular design, this can be easily exchanged if more performant methods are available in the future. Second, detection of both isolated and clustered RNA. Detection results can be conveniently inspected with the Kaibu image viewer plugin in ImJoy and different detection settings interactively investigated. Batch processing of entire folders is also possible. Lastly, detection results can be assigned to segmented cells and nuclei. We provide an interactive demo version of this plugin that can run directly in the browser without any local installation (https://fish-quant.github.io/fq-interactive-docs/#/fq-imjoy).

Using ImJoy provides several advantages beyond simply providing a user interface. Due to its distributed design that separates GUI from computation plugins, it natively supports user-friendly remote computing which allows access to massive data storage and powerful computation resources including GPUs. ImJoy is a browser-based app where the user-interface plugin is implemented with JavaScript/CSS/HTML. ImJoy then transparently calls the computation functions in the Big-FISH package running on a Python plugin engine (e.g., Jupyter server) to perform the actual smFISH analysis task (Fig. 4). While this plugin can run on a local workstation, it can be executed on a computational cluster or even in the cloud or seamlessly switching between them. This is illustrated by the demo version, where the engine is running on Binder (Project Jupyter et al. 2018). Once the plugin engine is installed on the remote resource, the end-user can connect with ImJoy and will be confronted with the same interface, independently of where the analysis is actually performed. Interestingly, this front-end interface can also be opened with mobile devices, providing easy access.

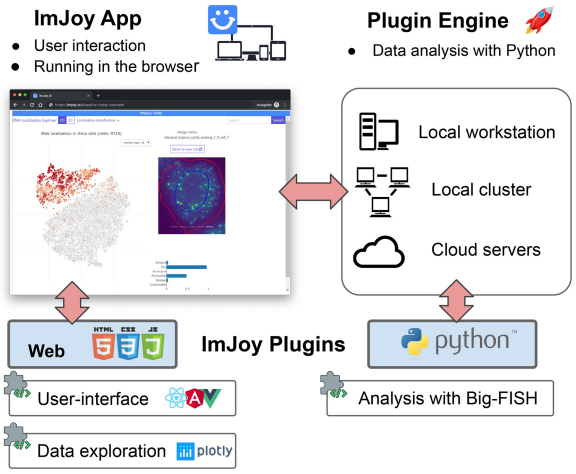

FIGURE 4. ImJoy. Schematic view of Imjoy's architecture. ImJoy's core is a Progressive Web App whose functionalities are provided by plugins that can be written in different programming languages. ImJoy can perform computations in the browser (including offline), locally or remotely via plugin engines.

ImJoy plugins implemented in JavaScript not only provide modern and reactive user-interfaces, but also profit from the extensive JavaScript data visualization libraries to build interactive data-inspection tools. Such interactivity is becoming increasingly important, especially when large and complex data sets are analyzed where static plots are too limited. As a case example, we provide an interactive t-SNE plot for the data shown in Figure 2 (https://fish-quant.github.io/fq-interactive-docs/#/rnaloc-tsne). This plugin can be run without local installation and enables the user to explore and interact with these complex data.

Case studies

We developed and validated FISH-quant v2 for two large-scale smFISH studies (see Materials and Methods). These two examples are typical use cases and exemplify the kind of quantitative results provided by this software.

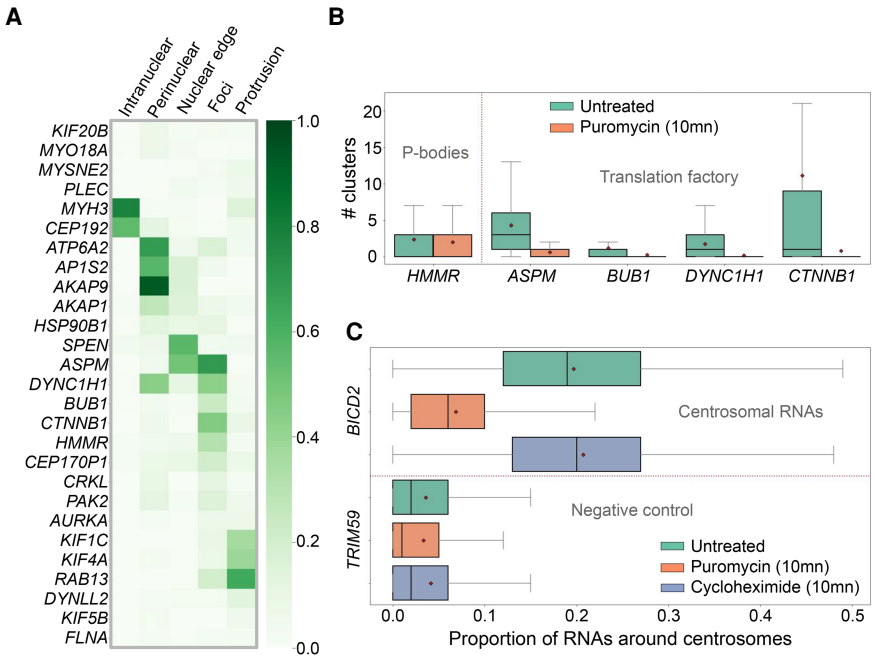

FIGURE 5. (*A*) Heatmap depicting the fraction of cells classified in the indicated pattern, for the different genes analyzed by the automated pipeline. (*B*) Impact of treatment with translational inhibitor puromycin on the number of detected RNA clusters. *HMMR* shows a similar number of clusters, while all other genes have significantly fewer, indicating an implication of translation in cluster formation. (*C*) Proportion of mRNAs within 2000 nm of a centrosome. Distance threshold was empirically defined as the typical distance between clustered RNAs and the centrosomes. Compared are untreated cells, and cells treated with two different translation inhibitors: cycloheximide, blocking ribosome elongation, or puromycin, inducing premature chain termination. *BICD2* has a centrosomal localization pattern, while *TRIM59* is a negative control with a random intracellular localization. Results are displayed with different treatments.

In Chouaib et al. (2020), we performed a high-content screen in HeLa cells and analyzed 10,000 segmented cells. FISH-quant v2 was used for spot detection, cell segmentation and the computation of localization features that allowed us to apply supervised and unsupervised machine learning to identify localization patterns and classify single cells into predefined pattern classes. We observed several distinct mRNA localization patterns, including RNA accumulating (i) in foci, (ii) in cytoplasmic protrusions, (iii) in the perinuclear area (which could be subdivided in endosomal, RE, Golgi and centrosome associate), (iv) forming a rim at the nuclear edge, or (v) inside the nucleus (Fig. 2). Interestingly, automated classification done on a single-cell level revealed a high degree of cell-to-cell heterogeneity in RNA localization, with 10% to 80% of the cells displaying the expected pattern depending on the RNA (Fig. 5A). In addition, for each pattern, only a fraction of the mRNA appeared to localize, revealing a high degree of plasticity in RNA localization mechanisms. This appears to be specific to cell lines as RNA localization in embryos is usually much more stereotyped. We also quantified how translation inhibition affected RNA localization and found that most mRNAs localize in a translation-dependent manner, which is unexpected (Fig. 5B). This also enabled us to discover translation factories, small cytoplasmic structures where specific mRNAs accumulate to be translated.

In Safieddine et al. (2021), we studied RNA localization at centrosomes (3600 images and 54,000 cells). Here, we added an automated detection for centrosomes (Fig. 3E), and implemented localization features describing this localization pattern (Fig. 5C). This enabled us to discover a family of eight centrosomal mRNAs whose localization to centrosome is cell cycle dependent and conserved from humans to drosophila.

Altogether, these analyses demonstrate the power of FISH-quant v2 in processing large smFISH data sets, and classifying RNA localization patterns in an automated way.

DISCUSSION

Here, we present FISH-quant v2, a user-friendly Python-based software for the complete analysis of smFISH images. It is built around a core-analysis package, implemented following rigorous software development guidelines, with detailed interactive documentation and tutorials. This package consists of several interchangeable modules permitting the construction of highly flexible workflows for specific analysis needs. For standard workflows, we provide user interfaces in ImJoy accessible to biologists without programming skills, which can be used locally or scaled to larger remote computational resources. Finally, FISH-quant hosts a simulation package to generate smFISH

images with nonrandom intracellular RNA localization patterns. These simulated images can be used to develop and evaluate analysis pipelines to study such RNA localization (Samacoits et al. 2018; Dubois et al. 2019). As demonstrated in two recently published studies (Chouaib et al. 2020; Safieddine et al. 2021), FISH-quant v2 can be used for large screening data sets thanks to its scalability. Spot detection, segmentation, feature extraction and pattern recognition can be performed over thousands of cells without fine-tuning parameters for every image.

We designed FISH-quant v2 based on the successful previous implementation in Matlab (Mueller et al. 2013) integrating new features and user feedback we obtained from several projects over several years. The entire core package is written in Python since this allowed us to address the above-mentioned requirements for a smFISH analysis tool. We use established scientific libraries (see Materials and Methods), and keep these dependencies to a minimum facilitating installation, maintenance and the integration with other analysis frameworks. These libraries are developed, validated and maintained by a large scientific community, ensuring long-term support and availability. We further use strict version control, guaranteeing reproducibility. Lastly, all dependencies, as well as FISH-quant v2, are open-source, thus can be used free of charge, both on local and remote computational infrastructures, and thus analysis can easily be scaled to larger data volumes.

The organization of the analysis subpackages in the core package matches key steps in smFISH image analysis, with a special focus on flexibility. All steps (preprocessing, RNA detection, segmentation as well as data inspection and analysis) can be run independently or replaced by external code, by respecting a strict data format. This allows FISH-quant to be adapted to the respective analysis needs, and build custom workflows.

While this flexibility is important, many users require a standard workflow and do not have programming experience. For these cases, we provide ImJoy plugins with a convenient user interface running in the browser (Ouyang et al. 2019). These interfaces are built with modern web libraries and are thus intuitive, and no experience in Python is required to analyze data. Lastly, these ImJoy plugins can be readily extended by more experienced users to further adapt them to their needs. A detailed documentation and interactive tutorial further help new users to get started quickly.

In summary, we present with FISH-quant v2 a rigorously validated analysis platform for smFISH data, developed to match the analysis requirements of large data sets. Its modularity permits the creation of flexible workflows ranging from the analysis of small data sets with the help of a graphical user-interface to custom-tailored investigation of large-scale screens requiring computational clusters.

MATERIALS AND METHODS

Python core packages

The repository Big-FISH contains the Python code used for the actual analysis. It is organized in several subpackages performing dedicated steps:

- I/O operations, images preprocessing and (*bigfish.stack*)
- mRNA spot detection (*bigfish.detection*)
- nucleus and cell segmentation (*bigfish.segmentation*)
- post-processing and analysis of results from different channels, such as the merging of RNA detections and segmentation masks or colocalization analysis (*bigfish.multistack*)
- feature computation, point cloud analysis and classification (*bigfish.classification*)
- visual reports of the obtained results (*bigfish.plot*)
- application of deep learning algorithms for segmentation (*bigfish.deep_learning*)

The repository Sim-FISH contains the Python code used for simulations. It includes several modules to generate 3D spots coordinates (both random and with a specific subcellular localization pattern). From these coordinates, simulated smFISH images with a noisy background can be generated.

Dependencies are limited to standard Python scientific libraries: scientific computing (numpy [Harris et al. 2020] and SciPy [Virtanen et al. 2020]), data wrangling (pandas [McKinney 2010]), image analysis (scikit-image [van der Walt et al. 2014]), visualization (matplotlib [Hunter 2007]), parallel computing (joblib, https://github.com/joblib/joblib) and machine learning (scikit-learn [Pedregosa et al. 2011], TensorFlow [Abadi et al. 2016]).

The GitHub repositories are using continuous integration providing increased robustness of the released code, through unitary testing, version control and automatically generated up-to-date documentation. Packages are hosted under a BSD 3-Clause License.

Example data sets

Two data sets were used for the development and validation of FISH-quant. First, from a screen studying local translation and consisting of 526 fields of view (DAPI and smFISH channels) from 57 separate experiments (27 different mRNAs under different experimental conditions [Chouaib et al. 2020]). For this screen, 3D images with a z-spacing of 0.3 µm were acquired on two different systems: (i) a Zeiss AxioimagerZ1 wide-field microscope equipped with a motorized stage, a camera scMOS ZYLA 4.2 MP, using 63× and 100× oil objectives, (ii) Nikon Ti fluorescence microscope equipped with ORCA-Flash 4.0 digital camera (HAMAMATSU). Second, from a screen focusing on local translation of centrosomal mRNAs. The data set consisted of 3678 fields of view (Dapi, smFISH, CellMask and GFP channels) from 218 experiments (Safieddine et al. 2021). 3D images were acquired with an automated spinning disk microscope (Opera, PerkinElmer), equipped with a 63× water objective. *Z*-spacing was 0.3 µm.

DATA DEPOSITION

The entire code for the analysis described in this paper is available on GitHub: https://github.com/fish-quant. This study includes no data deposited in external repositories.

SUPPLEMENTAL MATERIAL

Supplemental material is available for this article.

ACKNOWLEDGMENTS

This work was funded by the ANR (ANR-19-CE12-0007) and Institut Pasteur and by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). Furthermore, we also acknowledge France-BioImaging infrastructure supported by the French National Research Agency (ANR-10-INBS-04).

Received December 2, 2021; accepted February 19, 2022.

REFERENCES

Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, et al. 2016. TensorFlow: a system for large-scale machine learning. *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, OSDI '16*, pp. 265–283. USENIX Association, Savannah, GA. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

Bahry E, Breimann L, Epstein L, Kolyvanov K, Harrington KIS, Lionnet T, Preibisch S. 2021. RS-FISH: precise, interactive and scalable smFISH spot detection using radial symmetry. bioRxiv doi:10.1101/2021.03.09.434205

Battich N, Stoeger T, Pelkmans L. 2013. Image-based transcriptomics in thousands of single human cells at single-molecule resolution. *Nat Methods* **10:** 1127–1133. doi:10.1038/nmeth.2657

Battich N, Stoeger T, Pelkmans L. 2015. Control of transcript variability in single mammalian cells. *Cell* **163:** 1596–1610. doi:10.1016/j.cell.2015.11.018

Buxbaum AR, Haimovich G, Singer RH. 2014. In the right place at the right time: visualizing and understanding mRNA localization. *Nat Rev Mol Cell Biol* **16:** 95–109. doi:10.1038/nrm3918

Chin A, Lécuyer E. 2017. RNA localization: making its way to the center stage. *Biochim Biophys Acta Gen Subj* **1861:** 2956–2970. doi:10.1016/j.bbagen.2017.06.011

Chouaib R, Safieddine A, Pichon X, Imbert A, Kwon OS, Samacoits A, Traboulsi A-M, Robert M-C, Tsanov N, Coleno E, et al. 2020. A dual protein-mRNA localization screen reveals compartmentalized translation and widespread co-translational RNA targeting. *Dev Cell* **54:** 773–791.e5. doi:10.1016/j.devcel.2020.07.010

Cornes E, Bourdon L, Singh M, Mueller F, Quarato P, Wernersson E, Bienko M, Li B, Cecere G. 2021. piRNAs initiate transcriptional silencing of spermatogenic genes during *C. elegans* germline development. *Dev Cell* **57:** 180–196.e7. doi:10.1016/j.devcel.2021.11.025

Das S, Vera M, Gandin V, Singer RH, Tutucci E. 2021. Intracellular mRNA transport and localized translation. *Nat Rev Mol Cell Biol* **22:** 483–504. doi:10.1038/s41580-021-00356-8

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. doi:10.1038/nmeth.2075

Dubois R, Imbert A, Samacoïts A, Peter M, Bertrand E, Müller F, Walter T. 2019. A deep learning approach to identify mRNA localization patterns. *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pp. 1386–1390. doi:10.1109/ISBI.2019.8759235

Eichenberger BT, Zhan Y, Rempfler M, Giorgetti L, Chao JA. 2021. deepBlink: threshold-independent detection and localization of diffraction-limited spots. *Nucleic Acids Res* **49:** 7292–7297. doi:10.1093/nar/gkab546

Eng C-HL, Lawson M, Zhu Q, Dries R, Koulena N, Takei Y, Yun J, Cronin C, Karp C, Yuan G-C, et al. 2019. Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH+. *Nature* **568:** 235–239. doi:10.1038/s41586-019-1049-y

Falk T, Mai D, Bensch R, Çiçek Ö, Abdulkadir A, Marrakchi Y, Böhm A, Deubner J, Jäckel Z, Seiwald K, et al. 2019. U-Net: deep learning for cell counting, detection, and morphometry. *Nat Methods* **16:** 67–70. doi:10.1038/s41592-018-0261-2

Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, et al. 2020. Array programming with NumPy. *Nature* **585:** 357–362. doi:10.1038/s41586-020-2649-2

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. doi:10.1016/j.cels.2020.04.003

Hunter JD. 2007. Matplotlib: a 2D graphics environment. *Comput Sci Eng* **9:** 90–95. doi:10.1109/MCSE.2007.55

Lalit M, Tomancak P, Jug F. 2021. Embedding-based instance segmentation in microscopy. *Proceedings of the Fourth Conference on Medical Imaging with Deep Learning. PMLR* **143:** 399–415. https://proceedings.mlr.press/v143/lalit21a.html

McKinney W. 2010. Data structures for statistical computing in Python. In Proceedings of the 9th Python Science Conference, pp. 56–61. SciPy, Austin, TX. doi:10.25080/Majora-92bf1922-00a

McQuin C, Goodman A, Chernyshev V, Kamentsky L, Cimini BA, Karhohs KW, Doan M, Ding L, Rafelski SM, Thirstrup D, et al. 2018. CellProfiler 3.0: next-generation image processing for biology. *PLoS Biol* **16:** e2005970. doi:10.1371/journal.pbio.2005970

Moffitt JR, Zhuang X. 2016. RNA imaging with multiplexed error-robust fluorescence in situ hybridization (MERFISH). *Methods Enzymol* **572:** 1–49. doi:10.1016/bs.mie.2016.03.020

Mueller F, Senecal A, Tantale K, Marie-Nelly H, Ly N, Collin O, Basyuk E, Bertrand E, Darzacq X, Zimmer C. 2013. FISH-quant: automatic counting of transcripts in 3D FISH images. *Nat Methods* **10:** 277–278. doi:10.1038/nmeth.2406

Ouyang W, Mueller F, Hjelmare M, Lundberg E, Zimmer C. 2019. ImJoy: an open-source computational platform for the deep learning era. *Nat Methods* **16:** 1199–1200. doi:10.1038/s41592-019-0627-0

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. 2011. Scikit-learn: machine learning in python. *J Mach Learn Res* **12:** 2825–2830.

Perkel JM. 2019. Starfish enterprise: finding RNA patterns in single cells. *Nature* **572:** 549–551. doi:10.1038/d41586-019-02477-9

Pichon X, Lagha M, Mueller F, Bertrand E. 2018. A growing toolbox to image gene expression in single cells: sensitive approaches for demanding challenges. *Mol Cell* **71:** 468–480. doi:10.1016/j.molcel.2018.07.022

Pichon X, Moissoglu K, Coleno E, Wang T, Imbert A, Robert M-C, Peter M, Chouaib R, Walter T, Mueller F, et al. 2021. The kinesin

KIF1C transports APC-dependent mRNAs to cell protrusions. *RNA* **27:** 1528–1544. doi:10.1261/rna.078576.120

Project Jupyter, Bussonnier M, Forde J, Freeman J, Granger B, Head T, Holdgraf C, Kelley K, Nalvarte G, Osheroff A, et al. 2018. Binder 2.0—reproducible, interactive, sharable environments for science at scale. In Proceedings of the 17th Python in Science Conference (SCIPY 2018), pp. 113–120. SciPy, Austin, TX. doi:10.25080/Majora-4af1f417-011

Raj A, van den Bogaard P, Rifkin SA, van Oudenaarden A, Tyagi S. 2008. Imaging individual mRNA molecules using multiple singly labeled probes. *Nat Methods* **5:** 877–879. doi:10.1038/nmeth.1253

Ronneberger O, Fischer P, Brox T. 2015. U-Net: convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2015* (ed. Navab N, et al.). *Lecture Notes in Computer Science*, Vol. 9351, pp. 234–241. Springer International Publishing, Cham. doi:10.1007/978-3-319-24574-4_28

Safieddine A, Coleno E, Salloum S, Imbert A, Traboulsi A-M, Kwon OS, 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. doi:10.1038/s41467-021-21585-7

Samacoits A, Chouaib R, Safieddine A, Traboulsi A-M, Ouyang W, Zimmer C, Peter M, Bertrand E, Walter T, Mueller F. 2018. A computational framework to study sub-cellular RNA localization. *Nat Commun* **9:** 4584. doi:10.1038/s41467-018-06868-w

Savulescu AF, Brackin R, Bouilhol E, Dartigues B, Warrell JH, Pimentel MR, Beaume N, Fortunato IC, Dallongeville S, Boulle M, et al. 2021. Interrogating RNA and protein spatial subcellular distribution in smFISH data with DypFISH. *Cell Rep Methods* **1:** 100068. doi:10.1016/j.crmeth.2021.100068

Schmidt U, Weigert M, Broaddus C, Myers G. 2018. Cell detection with star-convex polygons. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018* (ed. Frangi AF, et al.). *Lecture Notes in Computer Science, Vol. 11071*, pp. 265–273. Springer International Publishing, Cham. doi:10.1007/978-3-030-00934-2_30

Stoeger T, Battich N, Herrmann MD, Yakimovich Y, Pelkmans L. 2015. Computer vision for image-based transcriptomics. *Methods* **85:** 44–53. doi:10.1016/j.ymeth.2015.05.016

Stringer C, Wang T, Michaelos M, Pachitariu M. 2021. Cellpose: a generalist algorithm for cellular segmentation. *Nat Methods* **18:** 100–106. doi:10.1038/s41592-020-01018-x

Tsanov N, Samacoits A, Chouaib R, Traboulsi AM, Gostan T, Weber C, Zimmer C, Zibara K, Walter T, Peter M, et al. 2016. SmiFISH and FISH-quant: a flexible single RNA detection approach with super-resolution capability. *Nucleic Acids Res* **44:** e165. doi:10.1093/nar/gkw784

van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, Gouillart E, Yu T. 2014. . scikit-image: image processing in Python. *PeerJ* **2:** e453. doi:10.7717/peerj.453

Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, et al. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods* **17:** 261–272. doi:10.1038/s41592-019-0686-2

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

Arthur Imbert, Wei Ouyang, Adham Safieddine, et al.

RNA 2022 28: 786-795 originally published online March 28, 2022
Access the most recent version at doi:10.1261/rna.079073.121