SMART-Q Manual

[1. Installation](#_Toc10648410)

[Overview of the Workflow](#_Toc10648411)

[2. Raw Data Processing and Organization](#_Toc10648412)

[File Naming Conventions](#_Toc10648413)

[Separating images](#_Toc10648414)

[Create configuration files](#_Toc10648415)

[3. RNA Quantification](#_Toc10648416)

[Filtering](#_Toc10648417)

[Detection](#_Toc10648418)

[Segmentation](#_Toc10648419)

[Labeling Dots and Nuclei Assignment](#_Toc10648420)

[Produce Composite Images](#_Toc10648421)

[Save Data](#_Toc10648422)

# 1. Installation

SMART-Q was written based on Python 3.7 and adapted from starfish. Some functions might not work in earlier versions, especially in Python 2.

If you have not already installed Python 3.7, I recommend doing so using miniconda3, which you can download from [this link](https://docs.conda.io/en/latest/miniconda.html). Once completed, create an environment for installments of SMART-Q, using the code below.

|  |
| --- |
| conda create -n SMART-Q y conda activate SMART-Q *# if this doesn’t work, then use source activate SMART-Q* conda install -c conda-forge openjdk=8 pyimagej y |

Next, download SMART-Q from github and install locally.

|  |
| --- |
| git clone https://github.com/shenlab-ucsf/SMART-Q cd SMART-Q python -m venv .venv source .venv/bin/activate pip install -e . |

If you are using a Mac, then use the following two lines. Otherwise, there may be issues with matplotlib, the visualization installment used by Python.

|  |
| --- |
| mkdir ~/.matplotlib echo "backend: TkAgg" > ~/.matplotlib/matplotlibrc |

Install the rest of the dependencies (IPython, tabulate, XlsxWrite, etc.) using pip.

|  |
| --- |
| pip install ipython==*7.1.1* pip install tabulate==*0.8.3* pip install XlsxWriter==*1.1.4*  pip install xlrd==*1.2.0* pip install cython==*0.29.7* pip install pyimagej==*0.4.0*  pip install opencv-python==*4.1.0.25* |

To know if the installation has worked, then use the template IPython file (.ipynb) provided by SMART-Q. In your command line, navigate to SMART-Q/pipeline\_template, use the code below to open jupyter-notebook, then click on the .ipynb file.

|  |
| --- |
| jupyter-notebook |

Run only the first cell of the script by pressing Shift+Enter on your keyboard or by clicking Run. If no errors show up, then the installation was successful. If you experience issues while troubleshooting any errors in the installation, reach out to seth.bergenholtz@gmail.com for help.

Once this is finished, in order to make sure that the correct version of Python is used to run SMART-Q every time you open a new session of bash, replace your own file path into the example code below and run it.

|  |
| --- |
| export PATH=/Users/sethb/SMART-Q/.venv/bin:$PATH |

You may also add the below code to the file called ~/.bash\_profile so that you do not have run the above code below every time you open a new bash session, but be aware that if you want to use a different version of Python afterward, then you will have to comment this line out or delete it from ~/.bash\_profile

|  |
| --- |
| echo "export PATH=/Users/sethb/SMART-Q/.venv/bin:$PATH" >> ~/.bash\_profile |

Finally, if you do not already have Fiji, download it from <https://imagej.net/Fiji/Downloads>.

# 

# Overview of the Workflow

The basic workflow of SMART-Q is outlined below, differing in the type of RNA counted. In the quick version, only nascent RNA are counted, while the longer version can quantify both mature and nascent RNA. A more detailed description of these algorithms will be presented in the following sections.



***Figure 1.*** *Schematic of workflow for counting of nascent and mature mRNA in SMART-Q.*

# 

# 2. Raw Data Processing and Organization

## File Naming Conventions

During image acquisition, images may be saved as either LIF or LSM files. Those files must then be broken down, organized, and formatted for analysis in SMART-Q. The way that files were named during image acquisition is important to this process, as files are grouped according to their batch, series, channel numbers, and z-plane numbers. If the file names are written with the same naming convention we used, then no changes will need to be made. Otherwise, you will need to make changes to the Fiji Macro and/or the formatting files described in the following two subsections in order to match your own file naming convention.

**Our file name convention**: [BatchName].lif - [SeriesName] - C=[ChannelNumber].tif

BatchName: top level name from imaging acquisition; a batch contains multiple series

SeriesName: sublevel name from imaging acquisition; a series contains all images from every channel

ChannelNumber: number corresponding to each channel. See below for our channel number convention

ChannelNumber convention\*:

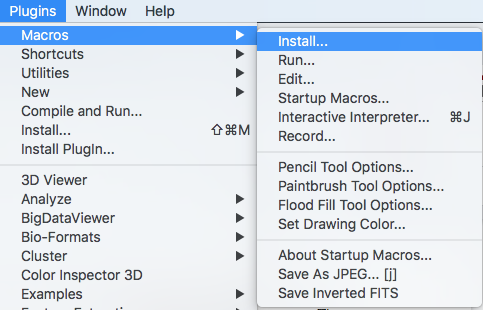
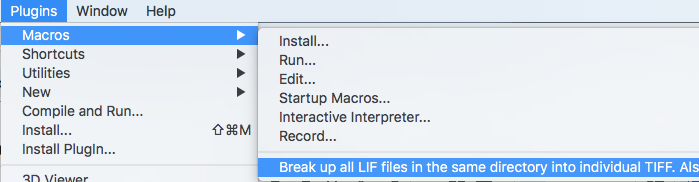
|  |  |
| --- | --- |
| C=0 | DAPI staining |
| C=1 | Cell type 1 |
| C=2 | Cell type 2 |
| C=3 | FISH/RNAscope signals |
| C=4,5,6,… | Any additional cell types |

\*This convention is only important to the next step as the Fiji Macro is currently hardcoded to match this convention. This will be changed in future updates.

For example: HFB-SSTR2-46.lif - SSTR2-46-1 - C=3--20. This file is part of a batch called HFB-SSTR2-46, a series called SSTR2-46-1, in the RNAscope channel, and is the 20th z-plane.

## Separating images

Fiji is used to break down LIF or LSM files into their separate TIF files by series, channel number and z-plane number. Fiji Macro scripts can be found in SMART-Q/fiji\_macros. There are three Macro scripts: one to process one specific LIF file, another to process all LIF files in a directory, and one to process all LSM files in a directory. In order to use any of these, follow the below steps:

1. Open Fiji, then select Plugins (in the tool bar) > Macros > Install
2. Navigate to and select the script of your choice.
3. Select Plugins > Macros > Run > the bottom-most option (its description will vary)  
   

4a. If processing a specific LIF file, navigate to and select that LIF file.

4b. If processing multiple LIF or LSM files, navigate to and select any one of those files.

Recall that they must all be in the same directory to be processed

1. Many files will continuously open and close as they are processed. Don’t click on the images as they are being processed, as this may upset the order they are processed in. The processed files will appear in a new directory called output, which will be located in the same directory the files are contained in.

After separation, the file names will follow this format: [BatchName].lif - [SeriesName] - C=[ChannelNumber]--[Z-PlaneNumber].tif

Z-PlaneNumber: placement of image in order of the z-axis

## Create configuration files

After the files are separated into their individual TIF files, they must then be organized and formatted into JSON files using bash commands and a Python script. If you have followed our file naming convention, then you only need to fill in the following variables in bash, then run the bash command:

nameBeforeDot: batch name

new\_dir: the name you want the new directory with raw and formatted files to be called

fileType: either lif or lsm

gene\_name: the name of the gene being analyzed

dirs: a list of all series names

raw\_image\_dir: the path to the directory containing the post-Fiji TIF files

For example, if I broke down a LIF file with the batch name HFB-GPX3-16 and series names ranging from 16-1 to 16-10 that analyzes the gene GPX3, then I would use the below variable entries:

|  |
| --- |
| nameBeforeDot=HFB-GPX3-16 new\_dir=rnascope\_${nameBeforeDot} fileType=lif gene\_name=GPX3 dirs="16-10 16-1 16-2 16-3 16-4 16-5 16-6 16-7 16-9" raw\_image\_dir=/Users/sb/Downloads/HFB-16/output |

More examples, the full script of bash commands, and the Python script can be found at SMART-Q/file organization templates.

If you have not followed our naming convention, then you will likely have to modify the Python script and possibly the bash commands as well. There is not a clear-cut way to do this since there are many ways that you can change the naming convention. It is not recommended to change the naming convention unless it is not adequate for your project. For help on how to accommodate the scripts for your naming convention, you may contact me at seth.bergenholtz@gmail.com.

# 

# 3. RNA Quantification

Nascent RNA should only be found in the nuclei. Because the DAPI staining gives us the best morphology and location of the nuclei, the nuclei are first labeled in the DAPI staining, and then all FISH/RNAscope dots are assigned to nuclei based on position. These nuclei are then assigned to their cell type channel(s).

Open Jupyter-Notebook

Navigate to the SMART-Q directory in your command line, then open Jupyter Notebook by typing jupyter-notebook. Run the first cell by pressing Shift+Enter. If there are no warnings, then all of the dependencies have been properly installed.

Write File Paths

In the second coding cell, change the directory names to match your setup, using the organization outlined below. This will tell the pipeline where the experiment.json file of the series of images you want to analyze is located. In addition, you will choose a directory in which the results and quality check of the analysis will be saved.

root\_dir: the directory that contains all of the batches. By default, this is the SMART-Q directory downloaded from Github

batch\_dir: the directory that contains all the series from a specific batch

GENE1: name of the gene in the experiment

series\_dir: the directory of the specific series being analyzed

outdir: the directory in which any figures and results produced will be saved

NOTE: the series\_dir name must be unique to each series, as the output directory’s name is based on the series\_dir name

For example, if your experiment.json file is located at "/Users/sb/Applications/SMART-Q/rnascope\_HES1/NT HES1/formatted/experiment.json", then you would write the following:

|  |
| --- |
| root\_dir = "/Users/sb/Applications/SMART-Q/" batch\_dir = "rnascope\_HES1/" GENE1 = "HES1" series\_dir = "HES1\_sample1-3/"  experiment = RNAscope(root\_dir + batch\_dir + series\_dir + "formatted/experiment.json")  output = "/Users/sb/Documents/RNAscope Results/Output/HES1\_sample1-3" |

When writing the outdir path, note that all directories except the last one must already be created, or else you will receive a FileNotFoundError.

## Initializing channels

SMART-Q contains three native classes that correspond to the three channel types: DotsChannel, NucleusChannel, and CellChannel. This format allows users to easily add any number of each channel type to the pipeline as necessary. Examine the code below to see how to create a channel object.

|  |
| --- |
| fov = experiment.fov()  dots = DotsChannel(fov['dots'])  nuclei = NucleusChannel(fov['nuclei'])  cellType1 = CellChannel(fov['channel1']) |

If you need to see all the channel names contained in fov, then you can call experiment.codebook.

Each channel type has unique functions and variables, which will be explained in detail throughout the following sections of the manual.

## Filtering

There are two reasons to perform filtering:

1. Amplify and smooth signals
2. Remove background noise

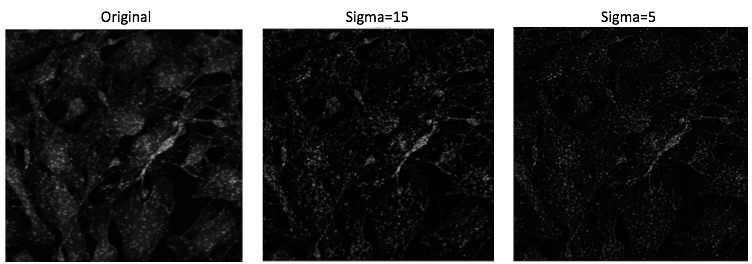
The first task is accomplished using a Gaussian Low Pass, which blurs the image and reduces edge content. The second task uses a Gaussian High Pass, which removes pixels with low frequencies, enhancing contrast and accentuating edges.[[1]](#footnote-1)

|  |  |  |
| --- | --- | --- |
| Parameters | Default Value | Effect of Changes |
| low\_sigma | For dots: 0.5  For nuclei or cells: 3 | Increasing low\_sigma will make the image blurrier and possibly introduce more noise. For images later used for segmentation, setting low\_sigma too low will result in “holes” in cells during Segmentation |
| high\_sigma | 5 | Decreasing high\_sigma will result in greater background reduction. Setting high\_sigma too high will leave background noise. |
| channelFile |  | Name of the filtered image that will be saved. In String format. For nuclei and cell type classes, this String must end with ‘.tiff’. |

**Table 1.** Variables used by filterDots() and filterCells().

Steps for Dots Filtering (Transcript Image):

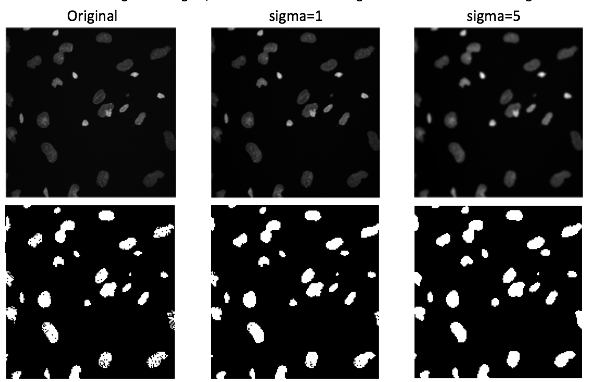
1. Run the cell starting with *# filter dots raw data*, keeping the parameters at default value.
2. Look at the resulting images. If a lot of background noise in between spots still remains, then decrease the value of high\_sigma. Re-run the cell.
3. Repeat Step #2 until desired results are ascertained.



***Figure 2. Gaussian High Pass on Transcripts Image.*** *Without filtering, the original image contains significant background noise, which would make detection inaccurate. At high\_sigma=15, the background noise is reduced but can still be improved. When filtered with high\_sigma=5, the image is ready for detection.*

Steps for Nuclei Filtering (DAPI staining):

1. Run the cell that starts with *# Filter nuclei raw data*, keeping the parameters at default value.
2. Continue the pipeline to Segmentation of nuclei. If there are small holes in the segmented nuclei that cannot be easily fixed by changing the parameters of Segmentation, then low\_sigma\_nuclei should be increased and then re-run. Increasing low\_sigma\_nuclei by too much runs the risk of making the nuclei appear significantly bigger in Segmentation than they actually are.



***Figure 3. Gaussian Low Pass on Nuclei Image.*** *Without filtering, the original image contains many black spots in each nucleus, signifying incomplete segmentation. When low\_sigma=1, the image is slightly blurred, thereby reducing the number of holes, but there are still a few black spots in the lower cells. When low\_sigma=5, there are no more black spots.*

Steps for Cell Filtering:

1. If you are quantifying nascent RNA transcripts, then filtering cells is not of significant importance. Simply use the default values.
2. If you are quantifying mature mRNA transcripts, then follow the same steps as nuclei filtering above, since the cells will eventually be segmented.

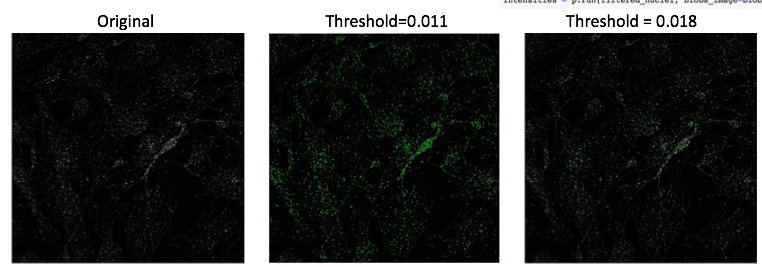
## Detection

In this step, we find the intensity and location of each fluorescent transcript, which present as spots in the FISH/RNAscope signal. Spots here are defined as bright circular objects with dark backgrounds. The min\_mass variable establishes the minimum brightness; all spots with intensity below min\_mass are excluded from detection. The value that you set the min\_mass at will depend on your microscope settings and the intensity of your samples’ fluorescence. There is a tool to determine a reasonable min\_mass, which will be described below.

NOTE: It is important to use the same parameters for each series in the same batch, so that the number of transcripts in the final results are comparable between series.

|  |  |  |
| --- | --- | --- |
| Parameters | Default Value | Effect of Changes |
| spot\_diameter | 5 | Expected spot size |
| min\_mass | Varies. For our mammalian samples, it ranged between 0.06 – 0.012 | Minimum integrated spot brightness/intensity. If too low, background noise will be included in transcript detection. If too high, valid spots will not be counted. |
| max\_size | 4 | Maximum radius of gyration of brightness |
| separation | spot\_diameter + 1 | Minimum distance between spots |

**Table 2.** Variables used by detection3D().



***Figure 4. Detection of Transcripts Image.*** *Using the original as a reference, we can see that setting threshold=0.018, many valid spots are omitted from detection and therefore the threshold is too high. Setting threshold=0.011 establishes a good distribution of detected spots.*

The steps of determining a proper threshold for your project are to:

1. Set the min\_mass, preferably to a lower value that results in slightly too much detection rather than too little. Run Detection.
2. In the coding cell below, set the lower\_threshold equal to min\_mass and the upper\_threshold to a slightly higher value, with a difference of 0.005-0.01. Run the cell. An image will appear with magenta circles displaying spots that had intensities between the lower\_threshold and upper\_threshold. Green dots represent detected spots above the upper\_threshold.
3. Zoom in to look at the contents inside the magenta circles. If the image is too pixelated and unclear, then increasing the size of your browser will increase the size and resolution of the image. If the contents are very dark, then the spots are either very weak or just background noise. In this case, increase both the lower\_threshold and upper\_threshold by the increment you previously chose. Run the cell again. Do NOT re-run detection, only the threshold tool, as detection takes much more time to run than the threshold tool.
4. Repeat step 3, incrementally increasing the values of lower\_threshold and upper\_threshold by 0.005-0.015 until the contents inside the magenta circles can be reasonably attributed to fluorescent transcripts. You may also decrease the difference between the lower\_threshold and upper\_threshold to gain precision.

## 

## Segmentation

In this step, we find the location of all nuclei or cells in the DAPI staining or cell type markers.

If you designed your fluorescent probes to target nascent RNA, then perform Segmentation only on the nuclei. These nuclei will be assigned to a cell type/channel in a later step. If you designed your fluorescent probes to target mature mRNA, then perform Segmentation on both the cells and the nuclei. This is because nascent RNA are present only in nuclei. Be aware that cell stains often do not fluoresce well in nuclear regions, so nuclei imaging and segmentation are generally necessary to obtain accurate location and morphology for each area.

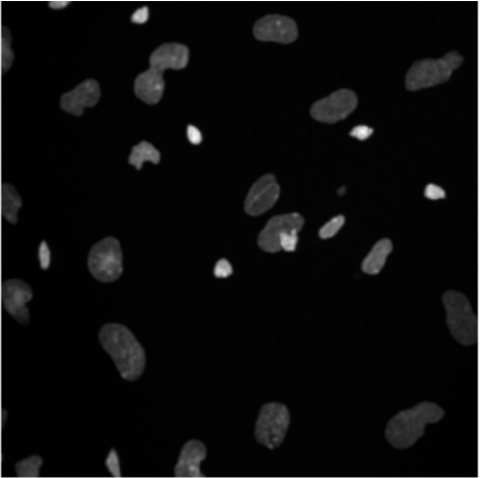
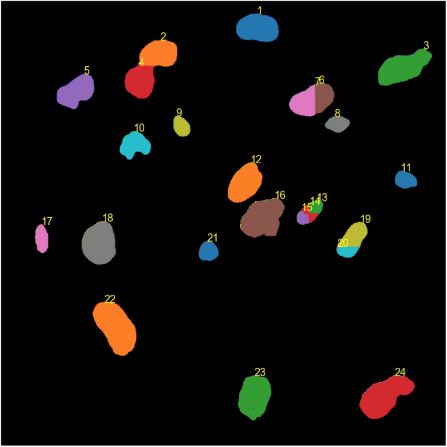
Segmentation relies on several parameters that you will have to actively change based on your experiment, unlike most other functions. Because Segmentation can be performed on either nuclei or cells, I will refer to them as blobs from here on. Segmentation allows you to establish the threshold, distance between blobs, size of the blobs, and the minimum depth. Depth here refers to the difference in brightness between pixels, where large depths indicate the edges of a blob.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameters | Definition | Default Value | Effect of Changes |
| dapi\_thresh | Minimum intensity; intensities below this will be excluded from segmentation | Ranges between 0.1 to 0.25 | Lowering threshold leads to greater signal from the nuclei. This can be useful when blobs are under-detected or if there are holes in blobs |
| min\_dist | Minimum distance between the centers of two blobs | 40 | Lowering distance results in greater segmentation, which is useful when two blobs are very close to each other or if there is noise very close to a blob which must be removed |
| min\_size | Minimum size of a blob. If a blob is below min\_size, it will be ignored and fail to appear | Nuclei: 25  Cells: 80 | Lowering size will allow small blobs to appear |
| min\_depth | Minimum depth between pixels that counts as an edge. | 0 | Using min\_depth will cause blobs to artificially under-segment. This is useful when a single blob is mistakenly segmented into two or more blobs. Min\_depth is 0 at default, but otherwise should be between .0001-0.000001 when being used. If min\_depth is too high, then it will have the opposite effect as intended. |
| segFilename | The desired file name of the segmentation process image |  |  |
| comparison Filename | The desired file name of the side-by-side comparison of the filtered blobs image with the segmented blobs |  |  |

**Table 3.** Variables used by detectNuclei().

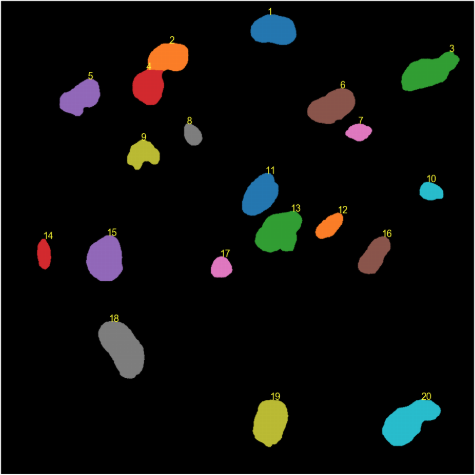
Steps:

1. Run Segmentation using the default values. An image of the segmentation process will be displayed, followed below by the filtered image on the left and the shapes of each segmented blob with IDs on the right. Compare the segmented blobs to the filtered image. There are several cases that would merit changing parameters, some of which are outlined below. For visualizations of the effects of changing parameters, see the figures in the next page.
   1. If significant blobs have incomplete coverage or weren’t segmented at all: decrease dapi\_thresh while avoiding the introduction of noise. Stop once noise is introduced, then decrease min\_dist instead.
   2. If significant blobs have too much coverage: Increase dapi\_thresh.
   3. If blobs are slightly over-segmented, meaning a few blobs are split into two or three segments: Refer to Step 3a on how to use mergeList.
   4. If blobs are extremely over-segmented, meaning many blobs are split into segments. Increase min\_dist to 35 or above or min\_depth to a value between 0.0001-0.000001. If the over-segmentation is still excessive, then return to Filtering and slightly increase the value of low\_sigma.
   5. If there is noise very close to a blob and they are segmented together: Decrease min\_dist with the goal of splitting the noise from the blob, then refer to step 3b on how to remove it.
   6. If noise was segmented: If the noise is small, then increase min\_size. Otherwise, refer to Step 3b on how to use removeList.
2. Repeat Step 1, prioritizing good coverage. Removing noise or reducing over-segmentation is more easily corrected in step 3. Move on once all significant blobs are segmented without holes in coverage.
3. Move to the cell below starting with # correct nuclei segmentation.
   1. If your image still has over-segmented blobs, then input the groups of blobs that must be merged together into mergeList in the following format: [(blob1a, blob1b), (blob2a,blob2b,blob2c), ...], where blob1 was split into 2 segments, blob2 was split into 3 segments. For example, the below image will use mergeList = [(6,7),(13,14,15),(19,20)]. The list does not need to be in any particular order; for example, [(15,13,14),(20,19),(7,6)] also works in this example.
   2. If your image has blobs that must be removed, then input them into removeList. There are 3 reasons to remove blobs: 1) the blob is noise and not actually a nucleus or cell, 2) the nucleus or cell does not match your experimental criteria (e.g. it is dead or belongs to the wrong cell type), and 3) there are two or more nuclei or cells overlapping with each other, which would skew the results. Write the list in the following format: removeList = [blob1,blob2,blob3]. For example, if the blobs with IDs 4, 15, and 32 are all undesired blobs, then removeList = [32,4,15]. They do not need to be in numerical order.

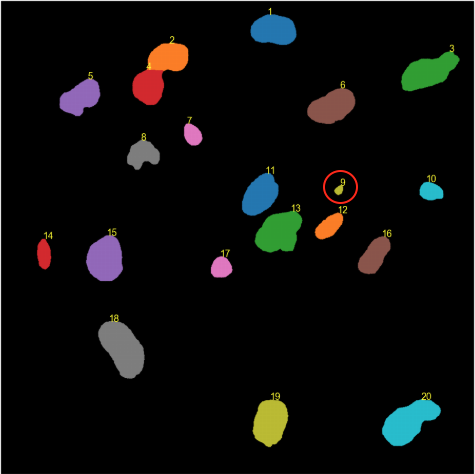
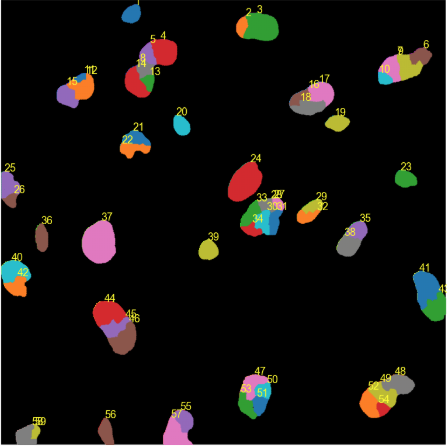
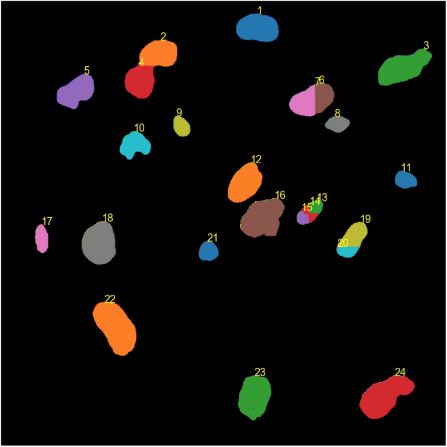
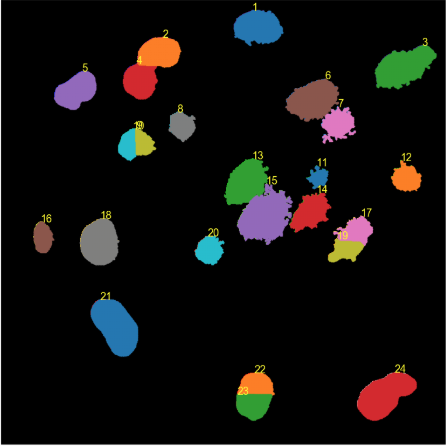
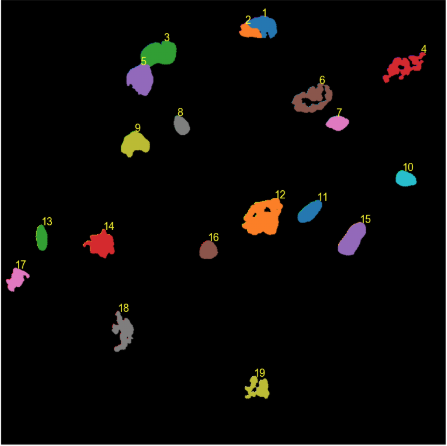


Original Filtered Image

Default segmented nuclei



Optimal Segmentation



A

B

C

D

E

F

G

H

min\_size = 10

min\_depth = 0.01

min\_dist = 20

min\_dist=15

dapi\_thresh = 0.24

dapi\_thresh = 0.036

***Figure 5. Optimization of nuclei segmentation.*** *By defining parameters at* *dapi\_thresh=0.12, min\_dist=35, min\_size=30, min\_depth=0.000001, mergeList=[(6,7)], optimal segmentation was obtained in* ***C****. D-H show examples about how parameters contribute to the segmentation.*

## 

## Labeling Dots and Nuclei Assignment

With Detection and Segmentation both completed, the data from these two steps are now ready to be combined. Labeling automatically combines these data to quantify RNA transcripts in segmented blobs. Finally, Nuclei Assignment is performed for each cell staining channel to indicate which nuclei belong to the positive cells in each channel. Because cells can be positive for more than one channel, nuclei can be assigned to any number of channels. Some nuclei may not belong to any channels.

If you are quantifying nascent RNA transcripts, then the pipeline will automatically provide an approximate estimate of which nuclei should be assigned to a particular channel, which you are able to verify and modify if necessary. If you are quantifying mature mRNA transcripts, then you will perform Segmentation on each cell staining channel, and then the nuclei will be automatically assigned to each channel.

Steps for nascent RNA quantification:

1. Run the coding cell starting with # count spots in segmented nuclei. This performs Labeling. You don’t need to change any parameters in this cell, unless you want to change the labeling quality assurance image.
2. Run the next coding cell, which contains the assignNuclei() function.
3. Compare the resulting image of the filtered Channel One cells to the image of the cells with Reference Nuclei. A positive cell in Channel One will appear as a bright blob; however, the location of the nucleus might be much less bright. Use the reference image to determine if the nucleus belongs to a cell. If so, then input its ID into the posNucleiCh1 list, which is located in the next cell. Run that cell to see the next channel.
4. Repeat Step 2 for Channel Two cells. Input all IDs of each nucleus positive for Channel Two into the posNucleiCh2 list, which is located in the cell below the previously run cell.
5. If you have any more channels, then you will have to add variables for this into the code. Simply copy the template of the other channels and change the variable names to indicate a third channel.
6. If you desire to know the combinations of nuclei that are positive/negative for any number of channels, such as nuclei that belong to cells that are positive for Channel One and negative for Channel Two, then use the compareResults function to generate images of these, using the following parameters detailed below. You can input only one channel if, for example, you only want to see the nuclei in cells positive for Channel One. Any combination of positive/negative channels can be displayed.

|  |  |  |
| --- | --- | --- |
| Parameters | Default Value | Effect of Changes |
| threshold | 70 | Minimum intensity for cell type image. Intensities below threshold will not be considered when assigning nuclei. |
| varName | "posNuc\_ch1" | The name of the posNuc variable, in String format. Look at the code below as an example. |

posNuc\_ch1, htm = channel1.assignNuclei(threshold, nuclei, "posNuc\_ch1")

**Table 4.** Variables used by assignNuclei().

|  |  |
| --- | --- |
| Parameters | Definition |
| positiveChannels | A list containing each channel object that should be positive in the resulting image. Only nuclei that appear in **every** list of positiveChannels and don’t appear in **any** list of negativeChannels will be displayed in the resulting image. |
| negativeChannels | A list containing channel objects that should be negative in the resulting image. All nuclei that do NOT appear in negativeChannels and do appear in positiveChannels will be displayed in the resulting image. |
| main\_channel | A CellChannel object. Its image will be used to display the positive nuclei and dot counts of the experimental group as a whole. For example, in the default pipeline, the choices are channel1, channel2, and channel4. |
| image\_title | The title of the comparison results image for main\_channel. For example, 'Ch1+, Ch2- (CRISPRi-mutated+, Negative Control-)'. |
| filename | The name of the images to be saved. One image containing just the resulting comparison image will be saved using filename. A second image containing the resulting image with the individual channels as reference will be saved using the name filename\_withIndivChs.png |

**Table 5.** Variables used by compareResults().

## 

## Produce Composite Images

This step is completely OPTIONAL. If you would like to combine the images produced so far for images in presentations or papers, then the pipeline can perform this using the compositeImageIJ function, which uses Fiji code in the Python pipeline itself.

1. Run the coding cell containing the imagej.init(fijiPath) function. This will initialize an object able to run Fiji macros.
2. Run the next coding cell containing the compositeImage() function. This will create an image of the final dots within the blobs analyzed and then produce a composite image with final quantifications and all filtered channel images.

|  |  |
| --- | --- |
| Parameters | Definition |
| dotsName | The desired name of the .tiff file with the final dots image. Must end with ‘.tiff’. |
| fijiPath | The file path to your Fiji application, in a string. Example: ‘/Applications/Fiji.app’ |

**Table 6.** Variables used by compositeImage().

## Save Data

This step saves all pertinent data in Excel sheets. Two Excel files are created:

1. A table that cumulatively keeps track of the most pertinent data for all series analyzed. The newest series will be appended to the end of this Excel file. For each series analyzed, this table contains the:
   * Total number of nuclei
   * Total number of spots
   * Number of spots in each nucleus
   * All cell IDs positive for each combination of interest, eg. Ch1+Ch2+, Ch2-, or any other combination
2. A table of all variables used at each step of analysis, useful for safekeeping of parameters used and re-creation of the analysis later on if necessary. This contains:

|  |  |  |
| --- | --- | --- |
| Parameter | Which step was it used in? | Brief Explanation |
| Dots Parameters | | |
| dots\_high\_sigma | Filtering | sigma for Gaussian High Pass of dots image |
| num\_high\_passes | Filtering | Number of times Gaussian High Pass filter is repeated |
| dots\_low\_sigma | Filtering | sigma for Gaussian Low Pass of dots image |
| spot\_diameter | Detection | Expected spot size |
| min\_mass | Detection | Minimum brightness |
| max\_size | Detection | Maximum radius of gyration of brightness |
| separation | Detection | Minimum distance between spots |
| Nuclei Parameters | | |
| low\_sigma\_nuclei | Filtering | sigma for Gaussian Low Pass of nuclei image |
| dapi\_thresh | Segmentation | Minimum intensity of DAPI staining |
| min\_dist | Segmentation | Minimum distance between center of nuclei |
| min\_size | Segmentation | Minimum size of nuclei |
| min\_depth | Segmentation | Minimum depth of edges of nuclei |
| mergeList | Segmentation | List of nucleus IDs to merge during segmentation correction |
| removeList | Segmentation | List of nucleus IDs to remove during segmentation correction |
| Cell Staining Parameters | | |
| low\_sigma\_cells | Filtering | sigma for Gaussian Low Pass of cell staining image |
| posNuclei | Nuclei Assignment | List of all positive nuclei IDs for that channel |

**Table 7.** Variables used by saveCumulativeData() and individual saving.

Any other variables that you may want to save can be added to the Excel sheets depending on which of the two Excel sheets you would like to add the value to:

1. To add data to the cumulative Excel file, add the data to the cumulative\_data dictionary. The data must be in a list the exact same size as the number of nuclei, using spacer to fill in any gaps in the list. For example, the number of nuclei only fills one index of a list, so spacer-1 is used to fill in the rest of the list:  
   'Number of Nuclei':[num\_nuclei\_corrected]+['']\*(spacer-1)
2. To add data to the individual Excel file, add it to an excel array from the current four choices: dotsExcelArray, nucleiExcelArray, channelOneExcelArray, channelTwoExcelArray. If your desired data does not belong to one of the images, then create a new excel array by copying one of the other excel arrays.

1. "Comparison between Butterworth and Gaussian High-pass Filters ...." 5 Jul. 2017, <http://paper.ijcsns.org/07_book/201707/20170716.pdf>. Accessed 16 May. 2019. [↑](#footnote-ref-1)