
FISH_Processing

Release 0.0.1

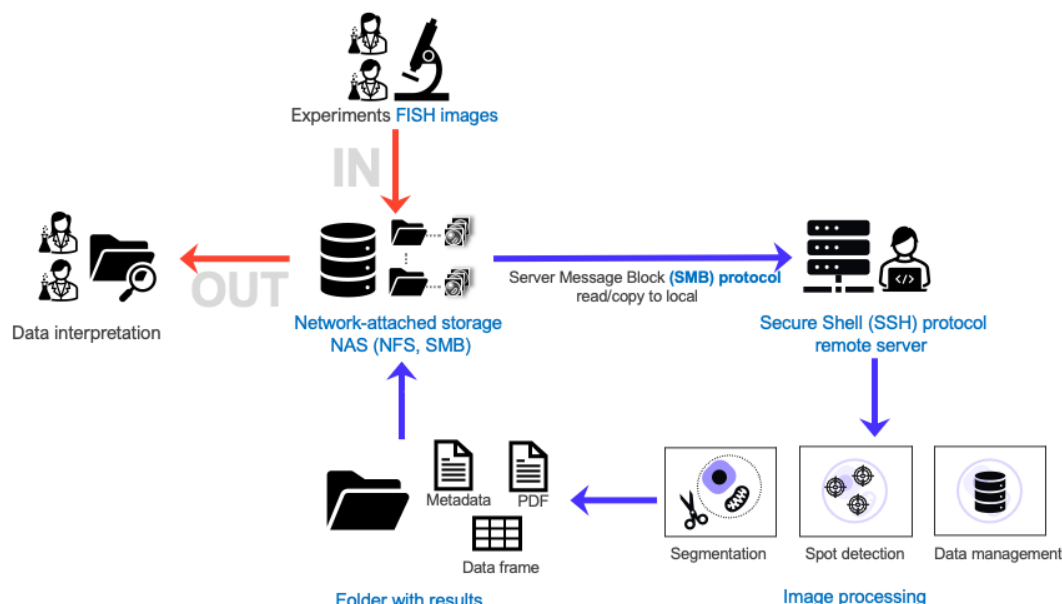
Luis Aguilera

Aug 16, 2023

CONTENTS:

1	Indices and tables	17
	Python Module Index	19
	Index	21

Code architecture



This module uses [pysmb](#) to allow the user to transfer data between a Network-attached storage (NAS) and remote or local server. Then it uses [Cellpose](#) to detect and segment cells on microscope images. [Big-FISH](#) is used to quantify the number of spots per cell. Data is processed using Pandas data frames for single-cell and cell population statistics. Authors: Luis U. Aguilera, Joshua Cook, Tim Stasevich, and Brian Munsy.

If you use this repository, make sure to cite:

[Big-FISH](#) : Imbert, Arthur, et al. “FISH-quant v2: a scalable and modular tool for smFISH image analysis.” RNA (2022): rna-079073.

[Cellpose](#) : Stringer, Carsen, et al. “Cellpose: a generalist algorithm for cellular segmentation.” Nature Methods 18.1 (2021): 100-106.

```
class fish_analyses.BigFISH(image, FISH_channel, voxel_size_z=300, voxel_size_yx=103, psf_z=350,
                             psf_yx=150, cluster_radius=350, minimum_spots_cluster=4,
                             show_plots=False, image_name=None, save_all_images=False,
                             display_spots_on_multiple_z_planes=False,
                             use_log_filter_for_spot_detection=True, threshold_for_spot_detection=None)
```

This class is intended to detect spots in FISH images using [Big-FISH](#) Copyright © 2020, Arthur Imbert. The format of the image must be [Z, Y, X, C].

Parameters

The description of the parameters is taken from [Big-FISH](#) BSD 3-Clause License. Copyright © 2020, Arthur Imbert. For a complete description of the parameters used check the [Big-FISH documentation](#).

image

[NumPy array] Array of images with dimensions [Z, Y, X, C] or [Y, X, C].

FISH_channel

[int] Specific channel with FISH spots that are used for the quantification

voxel_size_z

[int, optional] Height of a voxel, along the z axis, in nanometers. The default is 300.

voxel_size_yx

[int, optional] Size of a voxel on the yx plan in nanometers. The default is 150.

psf_z

[int, optional] Theoretical size of the PSF emitted by a spot in the z plan, in nanometers. The default is 350.

psf_yx

[int, optional] Theoretical size of the PSF emitted by a spot in the yx plan in nanometers.

cluster_radius

[int, optional] Maximum distance between two samples for one to be considered as in the neighborhood of the other. Radius expressed in nanometer.

minimum_spots_cluster

[int, optional] Number of spots in a neighborhood for a point to be considered as a core point (from which a cluster is expanded). This includes the point itself.

show_plots

[bool, optional] If True shows a 2D maximum projection of the image and the detected spots. The default is False

image_name

[str or None.] Name for the image with detected spots. The default is None.

save_all_images

[Bool, optional.] If true, it shows a all planes for the FISH plot detection. The default is False.

display_spots_on_multiple_z_planes

[Bool, optional.] If true, it shows a spots on the plane below and above the selected plane. The default is False.

use_log_filter_for_spot_detection

[bool, optional] Uses Big_FISH log_filter. The default is True.

threshold_for_spot_detection: scalar or None.

Indicates the intensity threshold used for spot detection, the default is None, and indicates that the threshold is calculated automatically.

detect()

This method is intended to detect RNA spots in the cell and Transcription Sites (Clusters) using [Big-FISH](#) Copyright © 2020, Arthur Imbert.

Returns

clusterDetectionCSV

[np.int64 Array with shape (nb_clusters, 5) or (nb_clusters, 4).] One coordinate per dimension for the centroid of the cluster (zyx or yx coordinates), the number of spots detected in the clusters, and its index.

spotDetectionCSV

[np.int64 with shape (nb_spots, 4) or (nb_spots, 3).] Coordinates of the detected spots. One coordinate per dimension (zyx or yx coordinates) plus the index of the cluster assigned to the spot. If no cluster was assigned, the value is -1.

```
class fish_analyses.CellSegmentation(image: ndarray, channels_with_cytosol=None,  
                                     channels_with_nucleus=None, diameter_cytosol: float = 150,  
                                     diameter_nucleus: float = 100,  
                                     optimization_segmentation_method='default',  
                                     remove_fragmented_cells: bool = False, show_plots: bool = True,  
                                     image_name=None, NUMBER_OF_CORES=1,  
                                     running_in_pipeline=False)
```

This class is intended to detect cells in FISH images using [Cellpose](#) . This class segments the nucleus and cytosol for every cell detected in the image. The class uses optimization to generate the meta-parameters used by cellpose. For a complete description of Cellpose check the [Cellpose documentation](#) .

Parameters

image

[NumPy array] Array of images with dimensions [Z, Y, X, C] or maximum projection with dimensions [Y, X, C].

channels_with_cytosol

[List of int] List with integers indicating the index of channels for the cytosol segmentation. The default is None.

channels_with_nucleus

[list of int] List with integers indicating the index of channels for the nucleus segmentation. The default is None.

diameter_cytosol

[int, optional] Average cytosol size in pixels. The default is 150.

diameter_nucleus

[int, optional] Average nucleus size in pixels. The default is 100.

optimization_segmentation_method: str

Method used for the segmentation. The options are: 'default', 'intensity_segmentation', 'z_slice_segmentation_marker', 'gaussian_filter_segmentation', and None.

remove_fragmented_cells: bool, optional

If true, it removes masks in the border of the image. The default is False.

show_plots

[bool, optional] If true, it shows a plot with the detected masks. The default is True.

image_name

[str or None.] Name for the image with detected spots. The default is None.

calculate_masks()

This method performs the process of cell detection for FISH images using **Cellpose**.

Returns

masks_complete_cells

[NumPy array. np.uint8] Image containing the masks for every cell detected in the image. Numpy array with format [Y, X].

masks_nuclei

[NumPy array. np.uint8] Image containing the masks for every nuclei detected in the image. Numpy array with format [Y, X].

masks_cytosol_no_nuclei

[NumPy array. np.uint8] Image containing the masks for every cytosol (removing the nucleus) detected in the image. Numpy array with format [Y, X].

class fish_analyses.Cellpose(*image: ndarray, num_iterations: int = 6, channels: list = [0, 0], diameter: float = 120, model_type: str = 'cyto', selection_method: str = 'cellpose_max_cells_and_area', NUMBER_OF_CORES: int = 1*)

This class is intended to detect cells by image masking using [Cellpose](#) . The class uses optimization to maximize the number of cells or maximize the size of the detected cells. For a complete description of Cellpose check the [Cellpose documentation](#) .

Parameters**image**

[NumPy array] Array of images with dimensions [Z, Y, X, C].

num_iterations

[int, optional] Number of iterations for the optimization process. The default is 5.

channels

[List, optional] List with the channels in the image. For gray images use [0, 0], for RGB images with intensity for cytosol and nuclei use [0, 1]. The default is [0, 0].

diameter

[float, optional] Average cell size. The default is 120.

model_type

[str, optional] Cellpose model type the options are 'cyto' for cytosol or 'nuclei' for the nucleus. The default is 'cyto'.

selection_method

[str, optional] Option to use the optimization algorithm to maximize the number of cells or maximize the size options are 'max_area' or 'max_cells' or 'max_cells_and_area'. The default is 'max_cells_and_area'.

NUMBER_OF_CORES

[int, optional] The number of CPU cores to use for parallel computing. The default is 1.

calculate_masks()

This method performs the process of image masking using **Cellpose**.

Returns**selected_masks**

[List of NumPy arrays] List of NumPy arrays with values between 0 and the number of detected cells in the image, where an integer larger than zero represents the masked area for each cell, and 0 represents the background in the image.

```
class fish_analyses.ColocalizationDistance(df, list_spot_type_to_compare=[0, 1], time_point=0,  
                                         threshold_intensity_0=0, threshold_intensity_1=0,  
                                         threshold_distance=2, show_plots=False,  
                                         voxel_size_z=None, psf_z=None, voxel_size_yx=None,  
                                         psf_yx=None,  
                                         report_codedetected_spots_in_both_channels=False)
```

This class is intended to calculate the Euclidean 2nd norm distance between the spots detected in two FISH channels.

Parameters**dataframe**

[Pandas Dataframe] Pandas dataframe with the following columns. image_id, cell_id, spot_id, nuc_loc_y, nuc_loc_x, cyto_loc_y, cyto_loc_x, nuc_area_px, cyto_area_px, cell_area_px, z, y, x, is_nuc, is_cluster, cluster_size, spot_type, is_cell_fragmented. The default must contain spots detected in two different color channels.

list_spot_type_to_compare

[list, optional] List indicating the combination of two values in spot_type to compare from the dataframe. The default is list_spot_type_to_compare =[0,1] indicating that spot_types 0 and 1 are compared.

time_point

[int, optional.] Integer indicating the time point at which the data was collected. This number is displayed as a column in the final dataframe. The default value is 0.

threshold_intensity_0

[int, optional] Integer indicating the intensity threshold used to collected the data for the first color channel. This number is displayed as a column in the final dataframe. The default value is 0.

threshold_intensity_1

[int, optional] Integer indicating the intensity threshold used to collected the data for the second color channel. This number is displayed as a column in the final dataframe. The default value is 0.

threshold_distance

[float, optional.] This number indicates the threshold distance in pixels that is used to determine if two spots are co-located in two different color channels if they are located inside this threshold_distance. The default value is 2.

show_plots

[Bool, optional.] If true, it shows a spots on the plane below and above the selected plane. The default is False.

voxel_size_z, voxel_size_yx: float, optional.

These values indicate the microscope voxel size. These parameters are optional and should be included only if a normalization to the z-axis is needed to calculate distance.

psf_z, psf_yx: float, optional.

These values indicate the microscope point spread function value. These parameters are optional and should be included only if a normalization to the z-axis is needed to calculate distance.

report_cotdetected_spots_in_both_channels

[bool, optional] This option report the number of co-detected spots in channel both channels. Notice that this represents the total number of cotdetected spots in ch0 and ch1. The default is True.

extract_spot_classification_from_df()

This method calculates the distance between the spots detected in two color channnels.

Returns

dataframe

[Pandas dataframe]

Pandas dataframe with the following columns: [time, ts_intensity_0, ts_intensity_1,

ts_distance, image_id, cell_id, num_0_only, num_1_only, num_0_1, num_0, num_1, total].

num_0_only = num_type_0_only num_1_only = num_type_1_only num_0_1 = num_type_0_1

num_0 = num_type_0_only + num_type_0_1 num_1 = num_type_1_only + num_type_0_1

num_0_total = total number of spots detected on ch 0. num_1_total = total number of spots detected on ch 1. total = num_type_0_only + num_type_1_only + num_type_0_1

```
class fish_analyses.DataProcessing(spotDetectionCSV, clusterDetectionCSV, image, masks_complete_cells,
                                masks_nuclei, masks_cytosol_no_nuclei, channels_with_cytosol,
                                channels_with_nucleus, yx_spot_size_in_px, spot_type=0,
                                dataframe=None, reset_cell_counter=False, image_counter=0,
                                number_color_channels=None)
```

This class is intended to extract data from the class SpotDetection and return the data as a dataframe. This class contains parameter descriptions obtained from [Big-FISH Copyright © 2020, Arthur Imbert](#). For a complete description of the parameters used check the [Big-FISH documentation](#).

Parameters

spotDetectionCSV: np.int64 Array with shape (nb_clusters, 5) or (nb_clusters, 4).

One coordinate per dimension for the cluster's centroid (zyx or yx coordinates), the number of spots detected in the clusters, and its index.

clusterDetectionCSV

[np.int64 with shape (nb_spots, 4) or (nb_spots, 3).] Coordinates of the detected spots . One coordinate

per dimension (zyx or yx coordinates) plus the index of the cluster assigned to the spot. If no cluster was assigned, the value is -1.

image

[NumPy array] Array of images with dimensions [Z, Y, X, C] .

masks_complete_cells

[List of NumPy arrays or a single NumPy array] Masks for every cell detected in the image. The list contains the mask arrays consisting of one or multiple Numpy arrays with format [Y, X].

masks_nuclei: List of NumPy arrays or a single NumPy array

Masks for every cell detected in the image. The list contains the mask arrays consisting of one or multiple Numpy arrays with format [Y, X].

masks_cytosol_no_nuclei

[List of NumPy arrays or a single NumPy array] Masks for every cell detected in the image. The list contains the mask arrays consisting of one or multiple Numpy arrays with format [Y, X].

channels_with_cytosol

[List of int] List with integers indicating the index of channels for the cytosol segmentation. The default is None.

channels_with_nucleus

[list of int] List with integers indicating the index of channels for the nucleus segmentation. The default is None.

yx_spot_size_in_px

[int] Size of the FISH spot in pixels.

spot_type

[int, optional] A label indicating the spot type, this counter starts at zero, increasing with the number of channels containing FISH spots. The default is zero.

dataframe

[Pandas dataframe or None.] Pandas dataframe with the following columns. image_id, cell_id, spot_id, nuc_loc_y, nuc_loc_x, cyto_loc_y, cyto_loc_x, nuc_area_px, cyto_area_px, cell_area_px, z, y, x, is_nuc, is_cluster, cluster_size, spot_type, is_cell_fragmented. The default is None.

reset_cell_counter

[bool] This number is used to reset the counter of the number of cells. The default is False.

image_counter

[int, optional] counter for the number of images in the folder. The default is zero.

get_dataframe()

This method extracts data from the class SpotDetection and returns the data as a dataframe.

Returns

dataframe

[Pandas dataframe] Pandas dataframe with the following columns. image_id, cell_id, spot_id, nuc_loc_y, nuc_loc_x, cyto_loc_y, cyto_loc_x, nuc_area_px, cyto_area_px, cell_area_px, z, y, x, is_nuc, is_cluster, cluster_size, spot_type, is_cell_fragmented.

class fish_analyses.GaussianFilter(video: ndarray, sigma: float = 1)

This class is intended to apply high and low bandpass filters to the video. The format of the video must be [Z, Y, X, C]. This class uses **difference_of_gaussians** from skimage.filters.

Parameters

video

[NumPy array] Array of images with dimensions [Z, Y, X, C].

sigma

[float, optional] Sigma value for the gaussian filter. The default is 1.

apply_filter()

This method applies high and low bandpass filters to the video.

Returns

video_filtered

[np.uint16] Filtered video resulting from the bandpass process. Array with format [T, Y, X, C].

class fish_analyses.**Intensity**(*original_image: ndarray, spot_size: int = 5, array_spot_location_z_y_x=[0, 0, 0], method: str = 'disk_donut'*)

This class is intended to calculate the intensity in the detected spots.

Parameters

original_image

[NumPy array] Array of images with dimensions [Z, Y, X, C]. spot_location_z_y_x

class fish_analyses.**MergeChannels**(*directory: str, substring_to_detect_in_file_name: str = '._C0.tif', save_figure: bool = False*)

This class takes images as arrays with format [Z, Y, X] and merges them in a NumPy array with format [Z, Y, X, C]. It recursively merges the channels in a new dimension in the array. The minimum number of channels 2 maximum is 4.

Parameters

directory: str or PosixPath

Directory containing the images to merge.

substring_to_detect_in_file_name: str

String with the prefix to detect the names of the files.

save_figure: bool, optional

If True, it saves the merged images as tif. The default is False.

checking_images()

Method that reads all images in the folder and returns a flag indicating if each channel in the image is separated in an independent file.

Returns

Flag

[Bool] If True, it indicates that each channel is split into different files. If False, it indicates that the image is contained in a single file.

merge()

Method takes all the images in the folder and merges those with similar names.

Returns

list_file_names

[List of strings] List with strings of names.

list_merged_images

[List of NumPy arrays] List of NumPy arrays with format np.uint16 and dimensions [Z, Y, X, C].

number_files

[int] The number of merged images in the folder.

```
class fish_analyses.Metadata(data_dir, channels_with_cytosol, channels_with_nucleus,
                             channels_with_FISH, diameter_nucleus, diameter_cytosol,
                             minimum_spots_cluster, list_voxels=None, list_psf=None,
                             file_name_str=None, list_segmentation_successful=True,
                             list_counter_image_id=[], threshold_for_spot_detection=[],
                             number_of_images_to_process=None, remove_z_slices_borders=False,
                             NUMBER_Z_SLICES_TO_TRIM=0, CLUSTER_RADIUS=0,
                             list_thresholds_spot_detection=[None], list_average_spots_per_cell=[None],
                             list_number_detected_cells=[None], list_is_image_sharp=[None],
                             list_metric_sharpness_images=[None], remove_out_of_focus_images=False,
                             sharpness_threshold=None)
```

This class is intended to generate a metadata file containing used dependencies, user information, and parameters used to run the code.

Parameters

data_dir: str or PosixPath

Directory containing the images to read.

channels_with_cytosol

[List of int] List with integers indicating the index of channels for the cytosol segmentation.

channels_with_nucleus

[list of int] List with integers indicating the index of channels for the nucleus segmentation.

channels_with_FISH

[list of int] List with integers indicating the index of channels for the FISH detection using.

diameter_cytosol

[int] Average cytosol size in pixels. The default is 150.

diameter_nucleus

[int] Average nucleus size in pixels. The default is 100.

minimum_spots_cluster

[int] Number of spots in a neighborhood for a point to be considered as a core point (from which a cluster is expanded). This includes the point itself.

list_voxels

[List of lists or None] List with a tuple with two elements (voxel_size_z, voxel_size_yx) for each FISH channel.

list_psf

[List of lists or None] List with a tuple with two elements (psf_z, psf_yx) for each FISH channel.

file_name_str

[str] Name used for the metadata file. The final name has the format metadata_<<file_name_str>>.txt

list_counter_cell_id

[str] Counter that keeps track of the number of images in the folder.

threshold_for_spot_detection

[int] Threshold value used to discriminate background noise from mRNA spots in the image.

write_metadata()

This method writes the metadata file.

```
class fish_analyses.NASConnection(path_to_config_file, share_name='share')
```

This class is intended to establish a connection between Network-Attached storage and a remote (or local) computer using `pysmb`. The class allows the user to connect to NAS, download specific files, and write backfiles to

NAS. This class doesn't allow the user to delete, modify or overwrite files in NAS. For a complete description of pysmb check the [pysmb documentation](#) . To use this class, you need to:

- 1) Use the university's network or use the two-factor authentication to connect to the university's VPN.
- 2) You need to create a configuration YAML file with the following format:

```
user:
username: name_of_the_user_in_the_nas_server
password: user_password_in_the_nas_server
remote_address : ip or name for the nas server
domain: domain for the nas server
```

Parameters

path_to_config_file

[str, or Pathlib object] The path in the local computer contains the config file.

share_name: str

Name of the share partition to access in NAS. The default is 'share'.

connect_to_server(timeout=60)

This method establishes the connection to the NAS.

Parameters

timeout

[int, optional] Time in seconds to maintain a connection with the NAS. The default is 60 seconds.

copy_files(remote_folder_path, local_folder_path, timeout=600, file_extension='.tif')

This method downloads tif files from NAS to a temporal folder in the local computer.

Parameters

remote_folder_path

[str, Pathlib obj] The path in the remote folder to download.

local_folder_path

[str, Pathlib obj] The path in the local computer where the files will be copied.

timeout

[int, optional] Time in seconds to maintain a connection with the NAS. The default is 60 seconds.

file_extension

[str, optional.] String representing the file type to download.

download_file(remote_file_path, local_folder_path, timeout=600)

This method download an specific file

Parameters

remote_file_path

[str, Pathlib obj] The path in the remote file to download.

local_folder_path

[str, Pathlib obj] The path in the local computer where the files will be copied.

timeout

[int, optional] Time in seconds to maintain a connection with the NAS. The default is 60 seconds.

read_files(*remote_folder_path*, *timeout=60*)

This method reads all files from a NAS directory

Parameters

remote_folder_path

[str, Pathlib obj] The path in the remote folder to download.

timeout

[int, optional] Time in seconds to maintain a connection with the NAS. The default is 60 seconds.

write_files_to_NAS(*local_file_to_send_to_NAS*, *remote_folder_path*, *timeout=600*)

This method writes files from a local computer to NAS

Parameters

local_file_to_send_to_NAS

[str, Pathlib obj] The path in the file to send to the NAS.

remote_folder_path

[str, Pathlib obj] The path in the remote folder to download.

timeout

[int, optional] Time in seconds to maintain a connection with the NAS. The default is 60 seconds.

class fish_analyses.**PipelineFISH**(*data_folder_path*, *channels_with_cytosol=None*,
channels_with_nucleus=None, *channels_with_FISH=None*,
diameter_nucleus=100, *diameter_cytosol=200*,
minimum_spots_cluster=None, *masks_dir=None*, *show_plots=True*,
voxel_size_z=500, *voxel_size_yx=160*, *psf_z=350*, *psf_yx=160*,
file_name_str=None, *optimization_segmentation_method='default'*,
save_all_images=True, *display_spots_on_multiple_z_planes=False*,
use_log_filter_for_spot_detection=True,
threshold_for_spot_detection=[None], *NUMBER_OF_CORES=1*,
list_selected_z_slices=None, *save_filtered_images=False*,
number_of_images_to_process=None, *remove_z_slices_borders=True*,
remove_out_of_focus_images=False, *sharpness_threshold=1.1*,
save_pdf_report=True)

This class is intended to perform complete FISH analyses including cell segmentation and spot detection.

Parameters *data_folder_path* : str or Pathlib object,

Path to the folder with the images to process.

parameter: bool, optional

parameter description. The default is True.

voxel_size_z

[int, optional] Microscope conversion px to nanometers in the z axis. The default is 500.

voxel_size_yx

[int, optional] Microscope conversion px to nanometers in the xy axis. The default is 160.

psf_z

[int, optional] Theoretical size of the PSF emitted by a [rna] spot in the z plan, in nanometers. The default is 350.

psf_yx: int, optional

Theoretical size of the PSF emitted by a [rna] spot in the yx plan, in nanometers. The default is 160.

list_masks

[List of Numpy or None.] list of Numpy arrays where each array has values from 0 to n where n is the number of masks in the image.

save_all_images

[Bool, optional.] If true, it shows a all planes for the FISH plot detection. The default is True.

display_spots_on_multiple_z_planes

[Bool, optional.] If true, it shows a spots on the plane below and above the selected plane. The default is False.

use_log_filter_for_spot_detection

[bool, optional] Uses Big_FISH log_filter. The default is True.

threshold_for_spot_detection: scalar, list, or None.

Indicates the intensity threshold used for spot detection, the default is None, and indicates that the threshold is calculated automatically.

list_selected_z_slices : list or None number_of_images_to_process: int or None, optional

This number indicates a subset of images to process from a given repository. The default is None, and this indicates that the code will process all images in the given repository.

remove_z_slices_borders

[bool optional] This flag indicates the removal of the two first and last 2 z-slices from the segmentation and quantification. This needed to avoid processing images out of focus. The default is True.

class fish_analyses.Plots

This class contains miscellaneous methods to generate plots. No parameters are necessary for this class.

plot_cell_size_spots(*channels_with_cytosol, channels_with_nucleus, cell_size, number_of_spots_per_cell, cyto_size, number_of_spots_per_cell_cytosol, nuc_size, number_of_spots_per_cell_nucleus, output_identification_string=None, spot_type=0*)

This function is intended to plot the spot count as a function of the cell size.

plot_images(*image, figsize=(8.5, 5), image_name='temp', show_plots=True*)

This method is intended to plot all the channels from an image with format [Z, Y, X, C].

Parameters

image: NumPy array

Array of images with dimensions [Z, Y, X, C].

figsize

[tuple with figure size, optional.] Tuple with format (x_size, y_size). the default is (8.5, 5).

class fish_analyses.ReadImages(*directory: str, number_of_images_to_process=None*)

This class reads all tif images in a given folder and returns each image as a Numpy array inside a list, the names of these files, path, and the number of files.

Parameters

directory: str or PosixPath

Directory containing the images to read.

read()

Method takes all the images in the folder and merges those with similar names.

Returns

list_images

[List of NumPy arrays] List of NumPy arrays with format np.uint16 and dimensions [Z, Y, X, C] or [T, Y, X, C].

path_files

[List of strings] List of strings containing the path to each image.

list_files_names

[List of strings] List of strings where each element is the name of the files in the directory.

number_files

[int] The number of images in the folder.

class fish_analyses.**RemoveExtrema**(*image: ndarray, min_percentile: float = 1, max_percentile: float = 99, selected_channels=None*)

This class is intended to remove extreme values from an image. The format of the image must be [Y, X], [Y, X, C], or [Z, Y, X, C].

Parameters

image

[NumPy array] Array of images with dimensions [Y, X], [Y, X, C], or [Z, Y, X, C].

min_percentile

[float, optional] Lower bound to normalize intensity. The default is 1.

max_percentile

[float, optional] Higher bound to normalize intensity. The default is 99.

selected_channels

[List or None, optional] Use this option to select a list channels to remove extrema. The default is None and applies the removal of extrema to all the channels.

remove_outliers()

This method normalizes the values of an image by removing extreme values.

Returns

normalized_image

[np.uint16] Normalized image. Array with dimensions [Y, X, C], [Y, X], or [Z, Y, X, C].

class fish_analyses.**ReportPDF**(*directory, filenames_for_pdf_report, channels_with_FISH, save_all_images, list_z_slices_per_image, threshold_for_spot_detection, list_segmentation_successful=True*)

This class intended to create a PDF report including the images generated during the pipeline.

Parameters

directory_results: str or PosixPath

Directory containing the images to include in the report.

channels_with_FISH

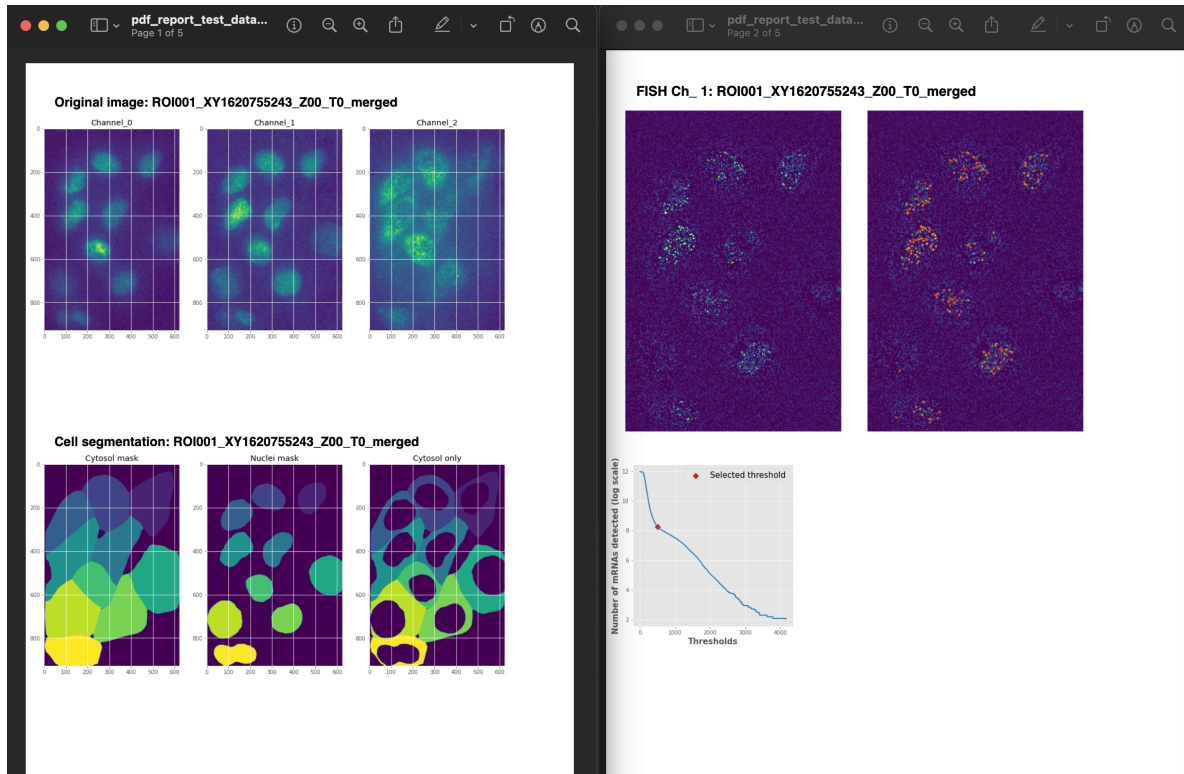
[list of int] List with integers indicating the index of channels for the FISH detection using.

save_all_images

[Bool, optional.] If true, it shows a all planes for the FISH plot detection. The default is True.

list_z_slices_per_image

[int] List containing all z-slices for each figure.



This PDF file is generated, and it contains the processing steps for each image in the folder.

create_report()

This method creates a PDF with the original images, images for cell segmentation and images for the spot detection.

```
class fish_analyses.SpotDetection(image, FISH_channels, channels_with_cytosol, channels_with_nucleus,
                                cluster_radius=500, minimum_spots_cluster=4,
                                masks_complete_cells=None, masks_nuclei=None,
                                masks_cytosol_no_nuclei=None, dataframe=None, image_counter=0,
                                list_voxels=[[500, 160]], list_psfs=[[350, 160]], show_plots=True,
                                image_name=None, save_all_images=True,
                                display_spots_on_multiple_z_planes=False,
                                use_log_filter_for_spot_detection=True,
                                threshold_for_spot_detection=None)
```

This class is intended to detect spots in FISH images using [Big-FISH](#). The format of the image must be [Z, Y, X, C]. This class is intended to extract data from the class SpotDetection and return the data as a dataframe. This class contains parameter description obtained from [Big-FISH](#) Copyright © 2020, Arthur Imbert. For a complete description of the parameters used check the [Big-FISH documentation](#).

Parameters

image

[NumPy array] Array of images with dimensions [Z, Y, X, C].

FISH_channels

[int, or List] List of channels with FISH spots that are used for the quantification

channels_with_cytosol

[List of int] List with integers indicating the index of channels for the cytosol segmentation. The default is None.

channels_with_nucleus

[list of int] List with integers indicating the index of channels for the nucleus segmentation. The default is None.

cluster_radius

[int, optional] Maximum distance between two samples for one to be considered as in the neighborhood of the other. Radius expressed in nanometer.

minimum_spots_cluster

[int, optional] Number of spots in a neighborhood for a point to be considered as a core point (from which a cluster is expanded). This includes the point itself.

masks_complete_cells

[NumPy array] Masks for every cell detected in the image are indicated by the array's values, where 0 indicates the background in the image, and integer numbers indicate the ith mask in the image. Array with format [Y, X].

masks_nuclei: NumPy array

Masks for every nucleus detected in the image are indicated by the array's values, where 0 indicates the background in the image, and integer numbers indicate the ith mask in the image. Array with format [Y, X].

masks_cytosol_no_nuclei

[NumPy array] Masks for every cytosol detected in the image are indicated by the array's values, where 0 indicates the background in the image, and integer numbers indicate the ith mask in the image. Array with format [Y, X].

dataframe

[Pandas Dataframe] Pandas dataframe with the following columns. image_id, cell_id, spot_id, nuc_loc_y, nuc_loc_x, cyto_loc_y, cyto_loc_x, nuc_area_px, cyto_area_px, cell_area_px, z, y, x, is_nuc, is_cluster, cluster_size, spot_type, is_cell_fragmented. The default is None.

image_counter

[int, optional] counter for the number of images in the folder. The default is zero.

list_voxels

[List of tuples or None] list with a tuple with two elements (voxel_size_z, voxel_size_yx) for each FISH channel. voxel_size_z is the height of a voxel, along the z axis, in nanometers. The default is 300. voxel_size_yx is the size of a voxel on the yx plan in nanometers. The default is 150.

list_psf

[List of tuples or None] List with a tuple with two elements (psf_z, psf_yx) for each FISH channel. psf_z is the size of the PSF emitted by a spot in the z plan, in nanometers. The default is 350. psf_yx is the size of the PSF emitted by a spot in the yx plan in nanometers.

show_plots

[bool, optional] If True, it shows a 2D maximum projection of the image and the detected spots. The default is False.

image_name

[str or None.] Name for the image with detected spots. The default is None.

save_all_images

[Bool, optional.] If true, it shows a all planes for the FISH plot detection. The default is False.

display_spots_on_multiple_z_planes

[Bool, optional.] If true, it shows a spots on the plane below and above the selected plane. The default is False.

use_log_filter_for_spot_detection

[bool, optional] Uses Big_FISH log_filter. The default is True.

threshold_for_spot_detection: scalar or None.

Indicates the intensity threshold used for spot detection, the default is None, and indicates that the threshold is calculated automatically.

class fish_analyses.Utilities

This class contains miscellaneous methods to perform tasks needed in multiple classes. No parameters are necessary for this class.

convert_to_int8(*image, rescale=True, min_percentile=1, max_percentile=98*)

This method converts images from int16 to uint8. Optionally, the image can be rescaled and stretched.

Parameters

image

[NumPy array] NumPy array with dimensions [Y, X, C]. The code expects 3 channels (RGB). If less than 3 values are passed, the array is padded with zeros.

rescale

[bool, optional] If True it rescales the image to stretch intensity values to a 95 percentile, and then rescale the min and max intensity to 0 and 255. The default is True.

dataframe_extract_data(*dataframe, spot_type, minimum_spots_cluster=2*)

This function is intended to read a dataframe and returns number_of_spots_per_cell, number_of_spots_per_cell_cytosol, number_of_spots_per_cell_nucleus, number_of_TS_per_cell, ts_size, cell_size

download_data_NAS(*path_to_config_file, data_folder_path, path_to_masks_dir, share_name, timeout=200*)

This method is intended to download data from a NAS. to a local directory. path_to_config_file data_folder_path path_to_masks_dir share_name, timeout

extracting_data_for_each_df_in_directory(*list_local_folders, current_dir, spot_type=0, minimum_spots_cluster=2*)

This method is intended to extract data from the dataframe

function_get_df_columns_as_array(*df, column_to_extract, extraction_type='all_values'*)

This method is intended to extract a column from a dataframe and convert its values to an array format. The argument <<<extraction_type>>> accepts two possible values.

values_per_cell: this returns an unique value that represents a cell parameter and is intended to be used with the following columns

'nuc_int_ch', 'cyto_int_ch', 'nuc_loc_y', 'nuc_loc_x', 'cyto_loc_y', 'cyto_loc_x', 'nuc_area_px', 'cyto_area_px', 'cell_area_px'

all_values: this returns all fields in the dataframe for the specified column.

merge_masks(*list_masks*)

This method is intended to merge a list of images into a single image (Numpy array) where each cell is represented by an integer value.

Parameters

list_masks

[List of Numpy arrays.] List of Numpy arrays, where each array has dimensions [Y, X] with values 0 and 1, where 0 represents the background and 1 the cell mask in the image.

separate_masks(*masks*)

This method is intended to separate an image (Numpy array) with multiple masks into a list of Numpy arrays where each cell is represented individually in a new NumPy array.

Parameters

masks

[Numpy array.] Numpy array with dimensions [Y, X] with values from 0 to n where n is the number of masks in the image.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

f

fish_analyses, 1

A

`apply_filter()` (*fish_analyses.GaussianFilter* method), 7

B

`BigFISH` (class in *fish_analyses*), 1

C

`calculate_masks()` (*fish_analyses.Cellpose* method), 4

`calculate_masks()` (*fish_analyses.CellSegmentation* method), 3

`Cellpose` (class in *fish_analyses*), 3

`CellSegmentation` (class in *fish_analyses*), 2

`checking_images()` (*fish_analyses.MergeChannels* method), 7

`ColocalizationDistance` (class in *fish_analyses*), 4

`connect_to_server()` (*fish_analyses.NASConnection* method), 9

`convert_to_int8()` (*fish_analyses.Utilities* method), 15

`copy_files()` (*fish_analyses.NASConnection* method), 9

`create_report()` (*fish_analyses.ReportPDF* method), 13

D

`dataframe_extract_data()` (*fish_analyses.Utilities* method), 15

`DataProcessing` (class in *fish_analyses*), 5

`detect()` (*fish_analyses.BigFISH* method), 2

`download_data_NAS()` (*fish_analyses.Utilities* method), 15

`download_file()` (*fish_analyses.NASConnection* method), 9

E

`extract_spot_classification_from_df()` (*fish_analyses.ColocalizationDistance* method), 5

`extracting_data_for_each_df_in_directory()` (*fish_analyses.Utilities* method), 15

F

`fish_analyses` module, 1

`function_get_df_columns_as_array()` (*fish_analyses.Utilities* method), 15

G

`GaussianFilter` (class in *fish_analyses*), 6

`get_dataframe()` (*fish_analyses.DataProcessing* method), 6

I

`Intensity` (class in *fish_analyses*), 7

M

`merge()` (*fish_analyses.MergeChannels* method), 7

`merge_masks()` (*fish_analyses.Utilities* method), 15

`MergeChannels` (class in *fish_analyses*), 7

`Metadata` (class in *fish_analyses*), 7

module
 fish_analyses, 1

N

`NASConnection` (class in *fish_analyses*), 8

P

`PipelineFISH` (class in *fish_analyses*), 10

`plot_cell_size_spots()` (*fish_analyses.Plots* method), 11

`plot_images()` (*fish_analyses.Plots* method), 11

`Plots` (class in *fish_analyses*), 11

R

`read()` (*fish_analyses.ReadImages* method), 11

`read_files()` (*fish_analyses.NASConnection* method), 9

`ReadImages` (class in *fish_analyses*), 11

`remove_outliers()` (*fish_analyses.RemoveExtrema* method), 12

`RemoveExtrema` (class in *fish_analyses*), 12

`ReportPDF` (class in *fish_analyses*), 12

S

`separate_masks()` (*fish_analyses.Utilities* method), 15

`SpotDetection` (*class in fish_analyses*), 13

U

`Utilities` (*class in fish_analyses*), 15

W

`write_files_to_NAS()` (*fish_analyses.NASConnection*
method), 10

`write_metadata()` (*fish_analyses.Metadata* method), 8