
Hi-C-Analysis

Release 0.1.0

Lorenzo Barsotti

Mar 19, 2023

CONTENTS:

1	Preprocessing Module	1
1.1	The create_graph function	1
1.2	The remove_empty_axis function	1
1.3	The extract_diagonal_blocks function	2
1.4	The get_chromosome_list function	2
1.5	The ooe_normalization function	3
1.6	The build_projectors function	3
1.7	The reconstruct_matrix function	3
2	Visualizegraph Module	5
2.1	The weights_distribuition function	5
2.2	The degree_distribution function	5
2.3	The plot_chromosome_hics function	6
2.4	The plot_chromosomes_histograms function	6
2.5	The plot_matrix function	6
2.6	The scatter_plot function	7
2.7	The histogram function	7
2.8	The show_10_projectors function	8
2.9	The plot_matrix_comparison function	9
3	Indices and tables	11
	Index	13

PREPROCESSING MODULE

1.1 The `create_graph` function

`hicanalysis.preprocessing.create_graph(matrix_df: DataFrame, metadata_df: DataFrame) → Graph`

This function creates a `nx.Graph` object, starting from the adjacency matrix of the network and the label of each node.

1.1.1 Parameters

`matrix_df (pd.DataFrame):`

Data Frame containing the adjacency matrix of the network.

`metadata_df (pd.DataFrame):`

Data Frame containing the metadata of the nodes.

1.1.2 Returns

`nx.Graph:`

Graph object of NetworkX library representing described by the adjacency contained in the *matrix_df*.

1.2 The `remove_empty_axis` function

`hicanalysis.preprocessing.remove_empty_axis(matrix: ndarray) → ndarray`

This function returns the given matrix removing the rows and columns that contain only 0.

1.2.1 Parameters

`matrix (np.ndarray):`

original matrix from which remove empty rows and columns (containing all zeros).

1.2.2 Returns

np.ndarray:

submatrix of the original one, where the empty rows and columns have been removed.

1.3 The `extract_diagonal_blocks` function

`hicanalysis.preprocessing.extract_diagonal_blocks(matrix: ndarray, block_size: list) → list`

This function extract the diagonal blocks of dimension contained in the `block_size`, and return them under the form of a list of `np.ndarray`.

1.3.1 Parameters

matrix (np.ndarray):

initial matrix from which we want to extract the blocks of the diagonal

block_size (list):

list containing the dimension of each block in the diagonal.

1.3.2 Returns

list:

list of `np.ndarray` containing the blocks of the diagonal extracted from the original matrix.

1.4 The `get_chromosome_list` function

`hicanalysis.preprocessing.get_chromosome_list(metadata_df: DataFrame) → list`

This function, given a dataframe as input, containing the columns `start` and `end`, containing the number of row of start and end of each chromosome, returns a list with the difference between `end` element and `start` element.

1.4.1 Parameters

metadata_df (pd.DataFrame):

Dataframe containing a column named `start`, a column `end` and a column with the name of the chromosome (not necessary).

1.4.2 Returns

list:

list containing the number of segments into which a chromosome has been divided.

1.5 The ooe_normalization function

`hicanalysis.preprocessing.ooe_normalization(matrix: ndarray) → ndarray`

This function allows to normalize a given matrix with the Observed-Over-Expected (OOE) algorithm for a symmetric matrix. In particular this algorithm provides to divide each element for the mean of the diagonal it belongs to.

1.5.1 Parameters

matrix (np.ndarray):
symmetric matrix to be normalized.

1.5.2 Returns

np.ndarray:
matrix normalized with the OOE algorithm.

1.6 The build_projectors function

`hicanalysis.preprocessing.build_projectors(eigenvectors: array, number_of_projector_desired: int) → list`

This function allows to get a list containing the projectors corresponding to the eigenvectors provided as input. The eigenvectors must be contained in a np.array where the columns are the eigenvectors.

1.6.1 Parameters

eigenvectors (np.array) :
Array containing the in the columns the eigenvectors of a matrix.

number_of_projector_desired (int) :
Number of projectors that you want to build.

1.6.2 Returns

list:
Output list containing the projectors corresponding to the input eigenvectors.

1.7 The reconstruct_matrix function

`hicanalysis.preprocessing.reconstruct_matrix(projectors: list, number_of_projectors: int) → ndarray`

This function takes as input a list of matrices (projectors) and a number of projectors to be added together in order to reconstruct the matrix.

1.7.1 Parameters

projectors (list) :

List containing np.ndarray of the projectors. It is supposed that the projectors are ordered from the most to the least important.

number_of_projectors (int) :

Number of projectors to be added together.

1.7.2 Returns

np.ndarray:

Reconstructed matrix from the addition of the first *number_of_projectors* projectors.

1.7.3 Raises

ValueError:

When the *number_of_projectors* parameter is greater than the length of the *projectors* list.

VISUALIZEGRAPH MODULE

2.1 The `weights_distribuion` function

`hicanalysis.visualizegraph.weights_distribuion(matrix: ndarray, savepath: str | None = None)`

This function, given an adjacency matrix of a weighted network, plots and shows the distribution of the weights of the network. If the parameter *savepath* is given the image will be saved in the location provided.

2.1.1 Parameters

matrix

[np.ndarray] Adjacency matrix under the form of np.ndarray.

savepath

[Optional[str, path-like], optional] Parameter to chose the path and name where to save the file. If this parameter is not given, the image will not be saved. Defaults to None.

2.2 The `degree_distribution` function

`hicanalysis.visualizegraph.degree_distribution(network_graph: Graph, savepath: str | None = None)`

This function, given a network graph, plots and shows the distribution of the degree of the nodes of the network. If the parameter *savepath* is given the image will be saved in the location provided.

2.2.1 Parameters

network_graph

[nx.Graph] Network graph under the form of object of the Graph class of Networkx library.

savepath

[Optional[str, path-like], optional] Parameter to chose the path and name where to save the file. If this parameter is not given, the image will not be saved. Defaults to None.

2.3 The `plot_chromosome_hics` function

`hicanalysis.visualizegraph.plot_chromosome_hics(matrices: ndarray, names: ndarray, savepath: str | None = None)`

This function produce an image containing the HiC matrices of the 22+2 chromosomes of the DNA. The function don't apply any logarithm to the data. If the parameter `savepath` is given the image will be saved in the location provided.

2.3.1 Parameters

matrices

[np.ndarray] Array containing all the 24 matrices corresponding to the chromosomes

names

[np.ndarray] Array containing the names to label each plot

savepath

[Optional[str, path-like], optional] Parameter to chose the path and name where to save the file.
If this parameter is not given, the image will not be saved. Defaults to None.

2.4 The `plot_chromosomes_histograms` function

`hicanalysis.visualizegraph.plot_chromosomes_histograms(matrices: ndarray, names: ndarray, savepath: str | None = None)`

This function produce an image containing the histograms of the absolute value of the eigenvalues of the 22+2 chromosomes of the DNA. The function don't apply any logarithm to the data. If the parameter `savepath` is given the image will be saved in the location provided.

2.4.1 Parameters

matrices

[np.ndarray] Array containing all the 24 matrices corresponding to the chromosomes.

names

[np.ndarray] Array containing the names to label each plot.

savepath

[Optional[str, path-like], optional] Parameter to chose the path and name where to save the file.
If this parameter is not given, the image will not be saved. Defaults to None.

2.5 The `plot_matrix` function

`hicanalysis.visualizegraph.plot_matrix(matrix: ndarray, title: str | None = "", savepath: str | None = None)`

This function shows the image of the matrix with a colorbar. It is possible to set a title with the parameter `title`. If the parameter `savepath` is given the image will be saved in the location provided.

2.5.1 Parameters

matrix

[np.ndarray] Matrix in the form of a np.ndarray.

title

[Optional[str], optional] Title to put in the image. Defaults to ‘’.

savepath

[Optional[str, path-like], optional] Parameter to chose the path and name where to save the file.
If this parameter is not given, the image will not be saved. Defaults to None.

2.6 The scatter_plot function

`hicanalysis.visualizegraph.scatter_plot(image1: ndarray, image2: ndarray, label1: str | None = 'Image 1', label2: str | None = 'Image 2', savepath: str | None = None)`

This function generate a scatter plot to compare two images or matrices. Given two numpy.ndarray, and optionally the label to assign to each image, it plots the scatter plot of the values of the images. If the parameter *savepath* is given the image will be saved in the location provided.

2.6.1 Parameters

image1

[np.ndarray] First image, that lays on the x-axis.

image2

[np.ndarray] Second image, that lays on the y-axis.

label1 Optional[str], optional

Label to assign to the first image. Defaults to ‘Image 1’.

label2 Optional[str], optional

Label to assign to the second image. Defaults to ‘Image 2’.

savepath

[Optional[str, path-like], optional] Parameter to chose the path and name where to save the file.
If this parameter is not given, the image will not be saved. Defaults to None.

2.7 The histogram function

`hicanalysis.visualizegraph.histogram(data, label, xlabel: str | None = "", ylabel: str | None = "", xlim: tuple | None = None, title: str = "", ylogscale: bool = False, savepath: str | None = None)`

This function plot an histogram of the *data* with a predefined *darkgrid* style of seaborn. In the case *data* is an array, it will be shown the histogram of the values, if *data* is a sequence of array it will be shown an image containing all the corresponding histograms. With the *data* parameter it is necessary to provide also a *label* parameter, that must have the same dimension of *data*. *xlabel* and *ylabel* are string that are used to set a label to the corresponding axis, while *xlim* is a tuple that help to limits in the visualization of the plot. It is possible to set a title for the plot with the *title* parameter and to set the log scale in y-axis with *ylogscale* parameter. If the parameter *savepath* is given the image will be saved in the location provided.

2.7.1 Parameters

data

[(n,) array or sequence of (n,) arrays] Input values, this takes either a single array or a sequence of arrays which are not required to be of the same length.

label

[str or sequence of str] Names of input values, this takes either a single string or a sequence of strings depending on the dimension of the input data.

xlabel

[Optional[str], optional] Label to be shown in the x-axis of the plot. Default to “.

ylabel

[Optional[str], optional] Label to be shown in the y-axis of the plot. Default to “.

xlim

[Optional[tuple], optional] Tuple containing the lower and upper limits to display in the plot. Default to None

title

[Optional[str], optional] Title of the plot to visualize in the image. Defaults to “.

ylogscale

[Optional[bool], optional] Parameter to set the log scale on y-axis. Defaults to False.

savepath

[Optional[str, path-like], optional] Parameter to chose the path and name where to save the file. If this parameter is not given, the image will not be saved. Defaults to None.

2.8 The show_10_projectors function

`hicanalysis.visualizegraph.show_10_projectors(projectors: list, savepath: str | None = None)`

This function shows in a (2 rows x 5 columns) grid the ten projectors given as input. If the parameter *savepath* is given the image will be saved in the location provided.

2.8.1 Parameters

projectors

[list] List containing ten matrices of the projectors to be shown.

savepath

[Optional[str, path-like], optional] Parameter to chose the path and name where to save the file. If this parameter is not given, the image will not be saved. Defaults to None.

2.9 The `plot_matrix_comparison` function

```
hicanalysis.visualizegraph.plot_matrix_comparison(matrix1: ndarray, matrix2: ndarray, title1: str |
None = "", title2: str | None = "",
pixels_to_be_masked: list | None = None,
savepath: str | None = None)
```

This function plots two matrices in the same image in order to compare them visually. With the parameter `pixels_to_be_masked` it is possible to set to zero some specific pixels of the first image to make it a more readable. If the parameter `savepath` is given the image will be saved in the location provided.

2.9.1 Parameters

matrix1

[np.ndarray] Matrix containing the data of the first image.

matrix2

[np.ndarray] Matrix containing the data of the second image.

title1

[Optional[str], optional] String containing the title of the first image. Defaults to "".

title2

[Optional[str], optional] String containing the title of the second image. Defaults to "".

pixels_to_be_masked

[Optional[list], optional] List of tuples that contain the coordinates of the pixel to be masked of the first image. Defaults to None. Ex. With the list `[(1, 2), (3, 4)]` are masked the pixel whose coordinates are: (1, 2) and (3, 4).

savepath

[Optional[bool], optional] Parameter to chose the path and name where to save the file. If this parameter is not given, the image will not be saved. Defaults to None.

INDICES AND TABLES

- `genindex`
- `search`

INDEX

B

`build_projectors()` (in module *hicanalysis.preprocessing*), 3

C

`create_graph()` (in module *hicanalysis.preprocessing*), 1

D

`degree_distribution()` (in module *hicanalysis.visualizegraph*), 5

E

`extract_diagonal_blocks()` (in module *hicanalysis.preprocessing*), 2

G

`get_chromosome_list()` (in module *hicanalysis.preprocessing*), 2

H

`histogram()` (in module *hicanalysis.visualizegraph*), 7

O

`ooe_normalization()` (in module *hicanalysis.preprocessing*), 3

P

`plot_chromosome_hics()` (in module *hicanalysis.visualizegraph*), 6

`plot_chromosomes_histograms()` (in module *hicanalysis.visualizegraph*), 6

`plot_matrix()` (in module *hicanalysis.visualizegraph*), 6

`plot_matrix_comparison()` (in module *hicanalysis.visualizegraph*), 9

R

`reconstruct_matrix()` (in module *hicanalysis.preprocessing*), 3

`remove_empty_axis()` (in module *hicanalysis.preprocessing*), 1

S

`scatter_plot()` (in module *hicanalysis.visualizegraph*), 7

`show_10_projectors()` (in module *hicanalysis.visualizegraph*), 8

W

`weights_distribuion()` (in module *hicanalysis.visualizegraph*), 5