DeepHicIntegrator Documentation

Release 0.1

Helene Kabbech, Eduardo Gade Gusmao, Akis Papantonis

Jun 24, 2019

Contents

DeepHicIntegrator permits the integration of a Hi-C matrix with one or several histone marks by interpolating in the latent space of an Autoencoder.

CHAPTER 1

Table of Contents

1.1 DeepHicIntegrator

This tool permits the integration of a Hi-C matrix with one or several histone marks by interpolating in the latent space of an Autoencoder.

1.1.1 Installation

Clone the repository

```
git clone https://github.com/kabhel/DeepHicIntegrator.git
cd DeepHicIntegrator
```

Requirements

1. A **linux** distribution.
2. **Python3** and the following python packages : **tensorflow**, **keras**, **docopt**, **schema**, **pandas**, **numpy**, **scipy**, **matplotlib**, **sklearn**, **cooler**, **hic2cool** and **m2r** (for Sphinx).

```
pip3 install -r requirements.txt
```

1. A Hi-C matrix in `.hic` file format.

Please, download the **GSE63525 HUVEC** genome in order to run the toy example.

```
wget -i ftp://ftp.ncbi.nlm.nih.gov/geo/series/GSE63nnn/GSE63525/suppl/GSE63525_HUVEC_
↪combined_30.hic.gz
gunzip GSE63525_HUVEC_combined_30.hic.gz
```

1. One or several histone marks in 2D dimension.

1.1.2 Run the program

Toy example

```
./deep_hic_integrator data/hic_matrix/GSE63525_HUVEC_combined_30.hic data/histone_
↪marks/100K/
```

Get help

```
Usage:
    ./deep_hic_integrator <HIC_FILE> <HM_PATH> [--resolution INT]
                                               [--chr_train INT]
                                               [--chr_test INT]
                                               [--hist_mark_train STR]
                                               [--square_side INT]
                                               [--epochs INT]
                                               [--batch_size INT]
                                               [--encoder STR]
                                               [--decoder STR]
                                               [--output PATH]
                                               [--help]

Arguments:
    <HIC_FILE>                          Path of the Hi-C matrix file (.hic format)
    <HM_PATH>                           Path of the repository containing the histone␣
↪mark files

Options:
    -r, INT, --resolution INT           Resolution representing the number of pair-
↪ended reads
                                        spanning between a pair of bins. [default:␣
↪25000]
    -a INT, --chr_train INT             Chromosome used to train the autoencoder␣
↪[default: 1]
    -t INT, --chr_test INT              Chromosome used to test the autoencoder␣
↪[default: 20]
    -m STR, --hist_mark_train STR       Name of the histone mark used to train the␣
↪autoencoder
                                        [default: h3k4me3]
    -n INT, --square_side INT           Size N*N of a sub-matrix [default: 20]
    -p INT, --epochs INT                Number of epochs for the training [default:␣
↪50]
    -b INT, --batch_size INT            Batch size for the training [default: 64]
    -e STR, --encoder STR               Trained encoder model (H5 format) [default:␣
↪None]
    -d STR, --decoder STR               Trained decoder model (H5 format) [default:␣
↪None]
```

(continues on next page)

(continued from previous page)

```
    -o PATH, --output PATH              Output path [default: results/]
    -h, --help                          Show this
```

1.1.3 Documentation

The documentation is generated with Sphinx and built on ReadTheDocs.

1.1.4 Author

Hélène Kabbech : Bioinformatics master student intern at the Medical Center University of Goettingen (Germany)

1.1.5 License

1.2 Implemented classes

1.2.1 Autoencoder

1.2.2 Matrix

class src.matrix.**Hic**(*cooler*, **args*, ***kwargs*)

class Hic
This class inherits the Matrix class and set the matrix numpy array for a Hi-C data.

cooler
Storage of the Hi-C matrix

Type cooler

calculate_cum_length()
Calculates and returns the cumulated length from chromosome 1 to N.

Returns Informations on chromosomes, their length and cumulated length

Return type Pandas DataFrame

set_matrix()
Set the Hi-C numpy array of the chromosome chrom_num. The matrix is transformed into an upper triangular matrix and the values are converted in float32 and rescaled by log10 and normalized.

class src.matrix.**HistoneMark**(*bed_file*, **args*, ***kwargs*)

class HistoneModification
This class inherits the Matrix class and set the matrix numpy array for a histone mark

mark_df
Histone modification sparse matrix

Type Pandas Dataframe

set_matrix()
Set the histone modification numpy array of the chromosome chrom_num. The values of the matrix are converted in float32 and rescaled by log10 and normalized.

class src.matrix.**Matrix**(*resolution*, *chrom_num*, *side*)

class Matrix
This class stores a matrix and different related numpy array, plots and writes this ma

resolution
Resolution (or bin size) of the matrix

Type int

chrom_num
Chromosome chosen for processing

Type int

side
Square side of a numpy array sub-matrix

Type int

matrix
Matrix stored in a numpy array

Type numpy array

sub_matrices
The matrix is divided into S sub-matrices of size side*side and stored in a numpy array of shape (X, side, side, 1)

Type numpy array

white_sub_matrices_ind
Position of the blank sub-matrices

Type list

total_sub_matrices
Total number of sub-matrices

Type int

latent_spaces
Latent spaces (encoded sub-matrices) stored in a numpy array

Type numpy array

predicted_sub_matrices
Predicted sub_matrices (decoded latent spaces) stored in a numpy array

Type numpy array

plot_distribution_matrix(*matrix_type*, *path*)
Plot the distribution of the matrix.

Parameters

- **matrix_type** (*str*) – Matrix's name
- **path** (*str*) – Path of the output plot

plot_matrix(*matrix_type*, *color_map*, *path*)
The matrix is plotted in a file.

Parameters

- **matrix_type** (*str*) – Matrix's name
- **color_map** (*matplotlib.colors.ListedColormap*) – Color map
- **path** (*str*) – Path of the output plot

plot_sub_matrices(*matrix_type*, *index_list*, *color_map*, *path*)
40 random sub-matrices are plotted in a file.

Parameters

- **matrix_type** (*str*) – Matrix's name
- **index_list** (*list*) – List of the 40 sub-matrix indexes to plot
- **color_map** (*matplotlib.colors.ListedColormap*) – Color map
- **path** (*str*) – Path of the output plot

set_predicted_latent_spaces(*latent_spaces*)
Set the latent spaces predicted by the encoder.

Parameters **latent_spaces** (*numpy array*) – The predicted latent_spaces

set_predicted_sub_matrices(*predicted_sub_matrices*)
Set the sub-matrices predicted by the whole autoencoder.

Parameters **predicted_sub_matrices** (*numpy array*) – The predicted sub-matrices

set_sub_matrices()
Divide the matrix into S sub-matrices of size side*side. The empty sub-matrices (sum(values)==0) are removed from the data set. The S resulted sub-matrices are stored in a numpy array of shape (X, side, side, 1).

write_sparse_matrix(*matrix_type*, *path*)
The reconstructed and predicted Hi-C matrix is saved in a sparse matrix file.

Parameters

- **matrix_type** (*str*) – Matrix's name
- **path** (*str*) – Path of the output

1.2.3 Interpolation

class src.interpolation.**Interpolation**(*alphas*)

class Interpolation
This class groups attributes and functions which aim to construct, write in a sparse m
and plot two or several interpolated matrices.

alphas
List of float values to use for the interpolation (alpha parameter)

Type list

interpolated_submatrices
List of all the interpolated sub-matrices. Each item in the list contains an interpolation with a different alpha.

Type list

integrated_matrix
List of all the integrated (interpolated) reconstructed matrices. Each item in the list contains an interpolation with a different alpha.

Type list

construct_integrated_matrix(*hic*)
Construction of the whole integrated matrices from the interpolated sub-matrices.

Parameters **hic** (*Hic (Matrix) object*) – Hi-C matrix

plot_integrated_matrix(*hic*, *color_map*, *path*)
The integrated matrices are plotted for each alpha value.

Parameters

- **hic** (*Hic (Matrix) object*) – Hi-C matrix
- **color_map** (*matplotlib.colors.ListedColormap*) – Color map
- **path** (*str*) – Path of the output plot

plot_interpolated_submatrices(*hic*, *index_list*, *color_map*, *path*)
40 random integrated sub-matrices are plotted for each alpha value.

Parameters

- **hic** (*Hic (Matrix) object*) – Hi-C matrix
- **index_list** (*list*) – List of the 40 sub-matrix indexes to plot
- **color_map** (*matplotlib.colors.ListedColormap*) – Color map
- **path** (*str*) – Path of the output plot

write_predicted_sparse_matrix(*hic*, *path*, *threshold=0.0001*)
The integrated matrices are saved in sparse matrix files for each alpha value.

Parameters

- **hic** (*Hic (Matrix) object*) – Hi-C matrix
- **path** (*str*) – Path of the output
- **threshold** (*float*) – The values under the threshold will be set to 0

class src.interpolation.**InterpolationInLatentSpace**(**args*, ***kwargs*)

class InterpolationInLatentSpace
This class inherits the Interpolation class and interpolate sub-matrices in the latent

interpolate_latent_spaces(*hist_marks*, *hic_latent_spaces*)
Double linear interpolation of the latent spaces of the Hi-C and histone marks.

Parameters

- **hist_marks** (*dict*) – Dictionary containing all histone mark HistoneMark objects.
- **predicted_hic** (*numpy array*) – Predicted sub-matrices of the Hi-C

set_decoded_latent_spaces(*decoder*, *side*)
The interpolated latent spaces are decoded.

Parameters

- **decoder** (*keras model object*) – Hi-C matrix
- **side** (*int*) – Square side

class src.interpolation.**NormalInterpolation**(**args*, ***kwargs*)

class InterpolationInLatentSpace
This class inherits the Interpolation class and interpolate sub-matrices in the pixel s
(= without the use of encoder and decoder).

alphas
List of float values to use for the interpolation (alpha parameter)

Type list

interpolated_submatrices
List of all the interpolated sub-matrices. Each item in the list contains an interpolation with a different alpha.

Type list

integrated_matrix
List of all the integrated (interpolated) reconstructed matrices. Each item in the list contains an interpolation with a different alpha.

Type list

interpolate_predicted_img(*hist_marks*, *predicted_hic*)
Double linear interpolation of the predicted sub-matrices of the Hi-C and histone marks.

Parameters

- **hist_marks** (*dict*) – Dictionary containing all histone mark HistoneMark objects.
- **predicted_hic** (*numpy array*) – Predicted sub-matrices of the Hi-C

Python Module Index

i

m

s

Index

R

S

T

W