'EpiCompass' Pipeline

July 17, 2025

Type Pipeline

Title Phenotype associated chromosomal region interpretation using epigenomic data

Version 1.0.1

Depends Python 3.11

Imports dask, matplotlib, pandas, pyarrow, scipy, seaborn, statsmodels, tqdm

Description Epigenetic data can provide critical insights into disease mechanisms, particularly regarding the phenotypic abnormalities they may cause. **EpiCompass** is a Python pipeline for reading, processing, and interpreting epigenetic data from the EpiMap repository (Epigenome Integration across Multiple Annotation Projects).

The EpiMap dataset provides genome-wide epigenetic state annotations across 859 biosamples, using an 18-state chromatin model. EpiCompass offers tools to extract and convert genomic regions of interest into a matrix of chromatin states. This matrix can then be analyzed to identify enriched states using a hypergeometric test, based on user-defined criteria such as tissue, sex, or life stage.

License MIT

Date 2025-07-17

NeedsCompilation no

Encoding UTF-8

git_url https://github.com/enriquealbprz/epicompass

git_branch main

git_last_commit f17962a

git_last_commit_date 2025-07-31

Repository epicompass

Date/Publication 2025-07-30

Author Enrique Albarrán Pérez

Contents

`EpiCompass`	*Phenotype associated chromosomal region interpretation using epigenomic data*

Description

Epigenetic data can provide critical insights into disease mechanisms, particularly regarding the phenotypic abnormalities they may cause. **EpiCompass** is a Python library for reading, processing, and interpreting epigenetic data from the [EpiMap repository](#) (Epigenome Integration across Multiple Annotation Projects).

Details

The EpiMap dataset provides genome-wide epigenetic state annotations across 859 biosamples, using an 18-state chromatin model. EpiCompass offers tools to extract and convert genomic regions of interest into a matrix of chromatin states. This matrix can then be analyzed to identify enriched states using a hypergeometric test, based on user-defined criteria such as tissue, sex, or life stage.

`classmap`	*Map samples by classification criteria*

Description

Generates a sample classification map (classmap) based on the criterion entered by the user, and saves it as a tab-separated values (.tsv) file. This classmap will be required as argument in order to run epimatrix.

Usage

```
python3 classmap.py criterion [--output OUTPUT]

# Example

python3 classmap.py tissue --output sort_by_tissue.tsv
```

Arguments

`criterion`	Sample classification criterion. Valid options: 'tissue', 'age', 'sex', or 'lifestage'.
`--output`	Output file for classmap (.tsv). Default: 'classmap.tsv'.

---

epimatrix   *Chromatin state count matrix*

Description

Reads and processes all the raw data files from the EpiMap repository stored in the specified directory. It divides the given genomic regions into fixed-size windows, determines the dominant chromatin state in each window, and generates a chromatin state count matrix sorted by the specified parameter. Optionally, it can collapse the 18-state model into broader categories and/or generate a plot showing the state assigned to each window.

Usage

```
python3 epimatrix.py dir entry window --classmap CLASSMAP [--statemap
STATEMAP] [--output OUTPUT] [--plot PLOT]

# Example

python3 epimatrix.py hg38_datasets chr7:140000-150000,chr10:100000-150000
5000 --classmap sort_by_tissue.tsv --statemap statemap.tsv --output
cancer_matrix.tsv --plot cancer_plot.png
```

Arguments

`dir`	Path to the directory containing the raw data.
`entry`	Chromosome region(s).
`window`	Window size.
`--classmap`	Path to sample classification .tsv file.
`--statemap`	[OPTIONAL] Path to state classification .tsv file.
`--output`	[OPTIONAL] Output file for matrix (.tsv).
`--plot`	[OPTIONAL] Output file for state visualization plot (.png).

---

hypertest   *Chromatin state matrix hypergeometric test*

Description

Performs a hypergeometric test on a chromatin state matrix previously generated by epimatrix and saved as a tab-separated values (.tsv) file, in order to identify enriched chromatin states across the parameter specified in epimatrix. Optionally, a heatmap can be generated to display the $-\log_{10}$(p-values) for each state as a function of the parameter.

Usage

```
python3 hypertest.py path [--output OUTPUT] [--plot] [--plot-output] [--rangecap MAXVALUE] [--fdr]

# Example

python3 hypertest.py cancer_matrix.tsv --output htest.tsv --plot --plot-output cancer_heatmap.png --rangecap 10 --fdr
```

Arguments

path	Path to the .tsv file containing the matrix.
--output	[OPTIONAL] Output file for hypergeometric test results (.tsv).
--plot	[OPTIONAL] Generate heatmap with $-\log_{10}$(p-values).
--plot-output	[OPTIONAL] Save heatmap to the specified file (.png).
--fdr	[OPTIONAL] Apply FDR correction (Benjamini-Hochberg).
--rangecap	[OPTIONAL] Upper limit for -log(p-value) in the heatmap color scale; cells containing higher values will appear with the same color. Default: 15.